Mobile Manipulation with Active Inference for Long-Horizon Rearrangement Tasks

Corrado Pezzato* **Ozan Çatal*** **Toon Van de Maele** **Riddhi J. Pitliya**

Tim Verbelen
VERSES AI Research Lab
Los Angeles, CA 90016, USA
{corrado.pezzato, ozan.catal}@verses.ai

Abstract

Despite growing interest in active inference for robotic control, its application to complex, long-horizon tasks remains untested. We address this gap by introducing a fully hierarchical active inference architecture for goal-directed behavior in realistic robotic settings. Our model combines a high-level active inference model that selects among discrete skills realized via a whole-body active inference controller. This unified approach enables flexible skill composition, online adaptability, and recovery from task failures without requiring offline training. Evaluated on the Habitat Benchmark for mobile manipulation, our method outperforms state-of-the-art baselines across the three long-horizon tasks, demonstrating for the first time that active inference can scale to the complexity of modern robotics benchmarks.

1 Introduction

Active inference offers a principled framework for modeling the action-perception loop, unifying inference and control. Both continuous and discrete formulations have been developed to capture sensorimotor and cognitive processes [18, 26], and past works have explored hybrid schemes that integrate discrete decision-making with continuous control in the context of handwriting [19] and oculomotor tasks [16, 17].

More recently, hybrid continuous-discrete approaches have shown promise in generating rich, goal-directed behavior in 2D simulated settings for reaching, grasping, and tool use [24, 21, 23]. However, active inference has yet to demonstrate scalability to the complexity and long time horizons required by modern robotics benchmarks. In particular, no prior work has convincingly shown that active inference alone can match or exceed the performance of state-of-the-art methods in realistic robotic tasks.

In this paper, we address this gap by introducing a fully hierarchical hybrid architecture for active inference-based control in long-horizon mobile manipulation tasks. Fig. 1 provides a high-level overview. Our system combines a high-level active inference agent that reasons over task-relevant abstractions with a novel whole-body controller based on continuous hierarchical active inference [22, 20]. This integration allows for flexible and robust skill execution, supports online adaptation, and eliminates the need for offline training. We evaluate our approach on three long-horizon mobile manipulation tasks from the Habitat Benchmark [28], namely `TidyHouse`, `PrepareGroceries`, and `SetTable`. These tasks require complex, multi-step interactions with

*Equal contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Embodied World Models for Decision Making.

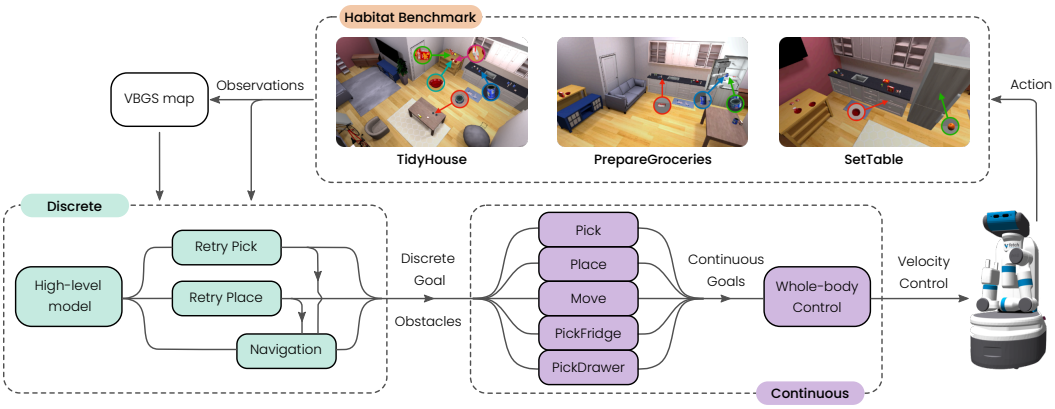

Figure 1: **Solution overview**. Overview of the proposed solution and the Habitat Tasks described in detail in section 2.

articulated objects and constrained environments, such as retrieving items from drawers or refrigerators, transporting them across rooms, and placing them on surfaces. Success demands both long-term planning and precise, reactive motor control. Our method outperforms state-of-the-art baselines across all three tasks, providing a compelling demonstration of active inference.

1.1 Related work

Long-Horizon mobile manipulation challenges that demand both navigation and manipulation abilities are well-suited to evaluating the effectiveness of embodied AI algorithms. Works like the Habitat Benchmark [28], ThreeDWorld [8], and ManipulaTHOR [5] require robots to navigate in indoor apartments and rearrange household objects. We chose to focus on the Habitat Benchmark because of its challenging nature: it requires continuous motor control (base and arm), interaction with articulated objects (opening drawers and fridges), and involves complicated scene layouts with clutter. In the literature, long-horizon problems have been tackled with task-and-motion-planning (TAMP) approaches [13, 12, 27, 9].

Although effective, these methods often rely on accurate knowledge of the scene and objects, and are computationally expensive. Learning-based approaches have emerged in recent years as an alternative; however, monolithic end-to-end RL solutions to long-horizon tasks are shown to be prone to failure [28, 10]. The main reasons for this are the high sample complexity, inefficient exploration, as well as complicated reward design.

A common strategy for addressing long-horizon tasks in RL is to decompose them into shorter, more manageable subtasks. For instance, the authors of Habitat [28] propose a hierarchical framework for mobile manipulation. This integrates classical task planning to generate high-level symbolic goals, while individual low-level skills are trained using RL to achieve these goals. This method demonstrates superior performance compared to monolithic end-to-end RL policies and traditional sense-plan-act pipelines. However, naively chaining multiple skills can result in hand-off issues [28], where the terminal state of one skill falls outside the distribution of states encountered during training by the subsequent skill, or leads to states that are infeasible for it to handle. This is especially an issue for stationary manipulation skills. Prior work often decouples the mobile base from the manipulator to simplify the inverse kinematics of redundant systems [25].

In contrast, [10] highlights that mobile manipulation skills are inherently more robust to error accumulation during skill chaining. By leveraging the robot's full embodiment, these skills enable more effective subtask execution by allowing the robot to reposition itself. Improved subtask formulation, skill composability, and reusability allowed [10] to reach state-of-the-art performance to date on Habitat.

Active inference offers a distinct perspective on decision-making and control, framing both as aspects of a unified inference process. The theory proposes that complex movements can emerge naturally from generative models that encode goals as prior beliefs and observation preferences [18].

Within this framework, the nervous system is seen as maintaining a hierarchical generative model that continuously produces and refines perceptual hypotheses.

Early work on hybrid active inference combining discrete and continuous processes focused on understanding systems such as oculomotion [7, 16, 17] and handwriting [19]. For instance, [7] proposed linking discrete and continuous states by using Bayesian model averaging over discrete priors, and converting the resulting continuous posterior into a discrete representation through Bayesian model comparison. These models typically use a discrete state-space to generate empirical priors, which then guide a continuous controller through sequences of attractive setpoints to achieve articulated behavior.

Hybrid active inference has been extended to dynamic tasks that demand flexible planning [24, 21, 23]. These scenarios require agents to infer object dynamics, decompose tasks into subgoals, and coordinate multiple degrees of freedom to execute composite actions. While such studies demonstrate the promise of hybrid active inference in complex motor control, no implementation has yet scaled to meet the complexity of established robotics benchmarks. In this work, we introduce a fully hierarchical hybrid active inference system that, for the first time, outperforms state-of-the-art baselines on the Habitat Benchmark, demonstrating its viability for long-horizon robotic manipulation.

1.2 Habitat Benchmark

The Habitat Benchmark [28] comprises three long-horizon mobile manipulation tasks visualized in Fig. 1: `TidyHouse`, `PrepareGroceries`, and `SetTable`.

In `TidyHouse`, the robot is tasked with relocating five objects from an initial to a designated goal position. Both the start and goal locations are typically on open surfaces such as tables or kitchen counters. The `PrepareGroceries` task involves moving two objects from an already open refrigerator to a countertop and returning one object from the counter back into the fridge. Finally, in `SetTable`, the agent must move a bowl from a closed drawer to a table, and then place an apple retrieved from a closed fridge on the same table. This scenario involves interacting with articulated objects and picking and placing items within confined containers. All tasks are specified as a sequence of subgoals. Each subgoal is a tuple (s_1, s^*), where s_1 is the initial 3D center-of-mass position of an object and s^* denotes its goal position. For instance, `TidyHouse` is defined by a set of five such tuples: $\{(s_1^i, s^{*i})\}_{i=1}^5$. The generated scenes for the tasks are built upon the ReplicaCAD dataset, which provides a diverse set of 105 photorealistic indoor environments. Each episode instantiates a rearrangement task by randomly sampling rigid objects from the YCB dataset and placing them on annotated support surfaces, resulting in cluttered initial configurations.

2 Methods

2.1 Planning with Universal Generative Models

To tackle the Habitat Benchmark, we propose an active inference agent, endowed with a hierarchy of generative models, each minimizing the Variational Free Energy. In this universal generative model [6] actions at one level become the preferences of the level below. As we traverse down the hierarchy, each model's time horizon becomes shorter, and the planning becomes more fine-grained. Each component in the hierarchy has its own set of responsibilities. The high-level model orchestrates the sequence of logical actions to take, i.e., where to move and what to pick or place. The *Navigation model* manages the path planning through the environment, while the *Pick & Place model* coordinates how targets should be approached or released. Finally, at the bottom of the hierarchy lies the active inference whole-body controller, which calculates joint controls for the robot and performs obstacle avoidance. These models keep track of their surroundings using a Variational Bayes approach to Gaussian splatting [14, 15], which integrates RGBD observations into a probabilistic world map.

2.1.1 High Level Model

At the most abstract level, the agent consists of a partially observed Markov Decision Process (POMDP) capturing dynamics over possible states of the object that need to be collected. This allows for the formation of beliefs over these states and robust planning of the agent's various skills.

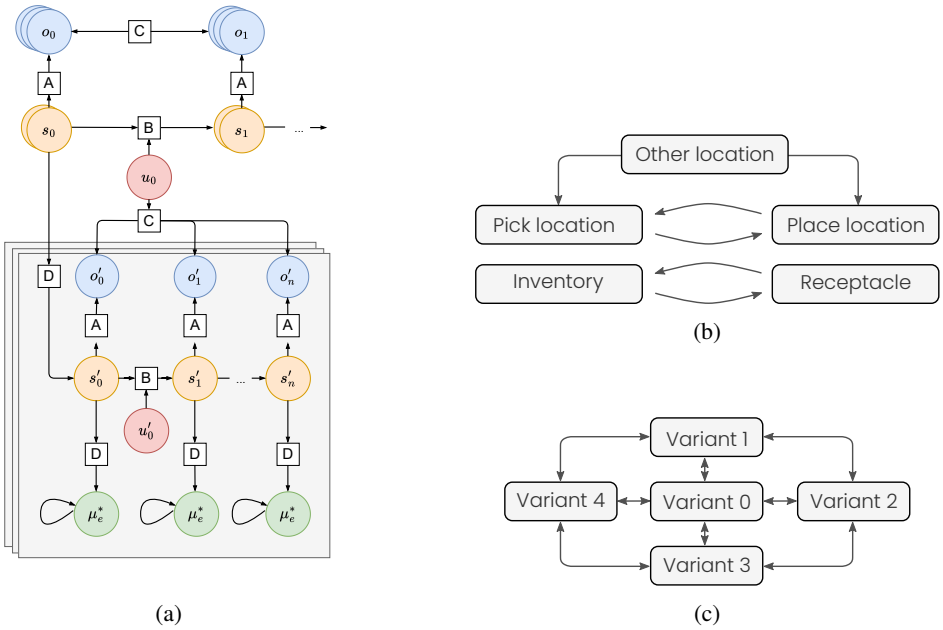

Figure 2: **The Generative Model.** (a) The high-level model sequences skills, each implemented by a generative model interacting with the continuous controller. (b) Dynamics at the highest level. The highest level models the robot location (top) and the object location (bottom). The robot can be either at an other location – irrelevant to the task – a pick location, or a place location. The object can be in the robot's inventory or receptacle (location). (c) Dynamics at the pick & place level. The model switches between various approach parameters based on feedback from the low-level controller to increase robustness against failures.

State and action inference in this POMDP is achieved by minimizing variational free energy according to the FEP [18]. Crucially, this model tracks the robot's position with respect to the pick/place location as well as the object's relation to the robot, the pick, and the place location. This is then used to schedule either movement or manipulation skills. We present this model using generic active inference terms in Fig. 2a. In contrast to classical active inference models, the actions in our model are abstractions of the skills they represent. The joint controls are generated by the continuous lowest-level model.

2.1.2 Pick & Place Model

Every interaction with the environment can fail due to unforeseen circumstances or invalid prior beliefs. To accommodate this, the hierarchy maintains a retry model for the failure-prone skills such as picking and placing. As shown in Fig. 2c, this model maintains a set of possible approach directions and switches between them based on detected successes or failures of a pick or place. Each approach parameter represents a goal for the lower level controller that can be enacted from that location.

2.1.3 Navigation Model

Crucially, still missing from our description of the hierarchical model is a way to move from one point to another. Discrete active inference models are well suited for this as shown in [4]. However, due to the static nature of the environment as well as the prior knowledge about the object location we opted to use A* pathfinding [11] to generate waypoints that act as extrinsic goals for the low level controller.

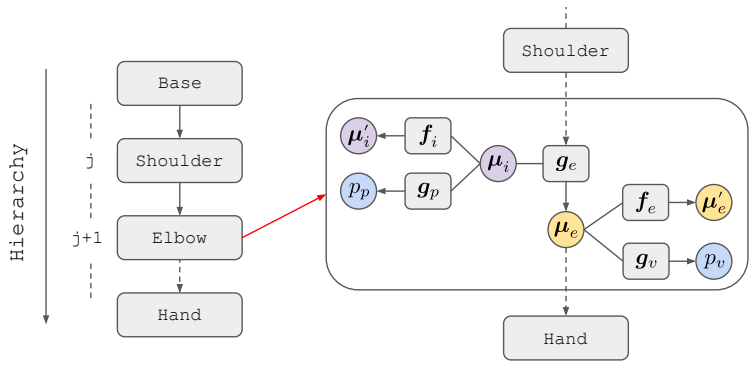

Figure 3: Overview of the Hierarchical Active Inference approach for mobile manipulator control. Intrinsic and extrinsic beliefs $\boldsymbol{\mu}_i$, $\boldsymbol{\mu}_e$ are internal representations of joint angles and Cartesian poses, respectively. They generate proprioceptive and visual predictions p_p and p_v at their level according to the generative models $\boldsymbol{g}_v$, $\boldsymbol{g}_p$. They are also linked through a kinematic generative model $\boldsymbol{g}_e$. The functions $\boldsymbol{f}_i$ and $\boldsymbol{f}_e$ describe the dynamics and are used to guide goal-directed behavior.

2.2 Perception with Variational Bayes Gaussian Splatting

The models need to infer the structure of the environment to keep track of obstacles and goals in the world. Following the Variational Bayesian approach used throughout the other models, we use a Variational Bayes Gaussian Splat (VBGS) [15] to build a 3D representation of the world from RGBD observations. In this model, the world is represented as a 6D Gaussian mixture over 3D points in space with corresponding 3D color information; the generative model is updated online from observations using Coordinate Ascent Variational Inference (CAVI) [1, 2, 3] without needing a replay buffer or observation queue. As with a normal 3D Gaussian Splat, the model effectively forms a radiance field that captures the room's free and occupied space. This allows for easy obstacle avoidance further down the hierarchy. The parameters of the distributions of component k (i.e. μ_k, Σ_k) that generate $\vec{s}$ and $\vec{c}$ are random variables, z is the associated mixture component for a given data point, dependent on the categorical parameters π.

2.3 Continuous control with Active Inference

Recent works proposed Hierarchical Active Inference (HAIF) schemes for continuous control of kinematic chains [22, 20]. In this section, we extend previous work [20] to whole-body control of differential drive mobile manipulators. Such robots are composed of a wheeled mobile base and one or more robot arms. We leverage the modularity of HAIF and define one generative model for base control and one for arm control, and then link them through *top-down prediction errors* and *bottom-up predictions*. This results in an overall control scheme that coordinates the whole body of the mobile manipulator at once. An overview of the HAIF approach is depicted in Fig. 3. Importantly, each block in the hierarchy has the same structure and follows the same update rules for state estimation and control. The difference for base and arm control lies in the definition of the generative model $\boldsymbol{g}_e$ and the physical quantities the internal and external beliefs represent, as explained next. In general, the kinematic generative model $\boldsymbol{g}_e$ computes the extrinsic beliefs at the current level j given the current intrinsic beliefs $\boldsymbol{\mu}_i^j$ and the extrinsic beliefs from the level below $\boldsymbol{\mu}_e^{j-1}$, i.e. $\boldsymbol{\mu}_e^j = \boldsymbol{g}_e(\boldsymbol{\mu}_i^j, \boldsymbol{\mu}_e^{j-1})$. The goal is to obtain a set of equations to describe how the internal beliefs of active inference agents are generated and updated over time. Following [22], the biologically plausible belief update equations are:

$$\dot{\boldsymbol{\mu}}_i^j = \begin{bmatrix} \boldsymbol{\mu}_i^{j'} + \pi_p^j \boldsymbol{\varepsilon}_p^j + \partial_{\boldsymbol{\mu}_i} \boldsymbol{g}_e^\top \pi_e^{j+1} \boldsymbol{\varepsilon}_e^{j+1} + \partial \boldsymbol{f}_i^{j\top} \pi_{\boldsymbol{\mu}_i}^j \boldsymbol{\varepsilon}_{\boldsymbol{\mu}_i}^j \\ -\pi_{\boldsymbol{\mu}_i}^j \boldsymbol{\varepsilon}_{\boldsymbol{\mu}_i}^j \end{bmatrix} \tag{1}$$

$$\dot{\boldsymbol{\mu}}_e^j = \begin{bmatrix} \boldsymbol{\mu}_e^{j'} - \pi_e^j \boldsymbol{\varepsilon}_e^j + \partial_{\boldsymbol{\mu}_e} \boldsymbol{g}_e^\top \pi_e^{j+1} \boldsymbol{\varepsilon}_e^{j+1} + \pi_v^j \boldsymbol{\varepsilon}_v^j + \partial \boldsymbol{f}_e^{j\top} \pi_{\boldsymbol{\mu}_e}^j \boldsymbol{\varepsilon}_{\boldsymbol{\mu}_e}^j \\ -\pi_{\boldsymbol{\mu}_e}^j \boldsymbol{\varepsilon}_{\boldsymbol{\mu}_e}^j \end{bmatrix}, \tag{2}$$

where π_p, π_e, π_v are precision parameters for proprioceptive, extrinsic, and visual models, and $\boldsymbol{\varepsilon}_p$, $\boldsymbol{\varepsilon}_e$ and $\boldsymbol{\varepsilon}_v$ are the proprioceptive, extrinsic, and visual prediction errors, respectively. Finally,

$\varepsilon^j_{\boldsymbol{\mu}_i} = \boldsymbol{\mu}_i^{j'} - \boldsymbol{f}_i^j(\boldsymbol{\mu}_i^j)$ and $\varepsilon^j_{\boldsymbol{\mu}_e} = \boldsymbol{\mu}_e^{j'} - \boldsymbol{f}_e^j(\boldsymbol{\mu}_i^j)$ are dynamics prediction errors, with precision $\pi_{\boldsymbol{\mu}_i}$ and $\pi_{\boldsymbol{\mu}_e}$. These are used to achieve goal-directed behavior and collision avoidance, as explained later. In the equations above, we assumed to be able to observe joint positions, velocities, and link positions such that $\boldsymbol{g}_p$ and $\boldsymbol{g}_v$ are identity mappings. Link positions can be computed from joint positions via forward kinematics or estimated via visual input. We now have to find a suitable form for $\boldsymbol{g}_e$ to easily compute the gradients with respect to intrinsic and extrinsic beliefs for both the arm and the base.

2.3.1 Arm generative model

Similarly to [20], the generative model for the HAIF agent comprises an intrinsic belief $\boldsymbol{\mu}_i$ about joint angles and links' lengths, as well as an extrinsic belief $\boldsymbol{\mu}_e$ about a link's absolute Cartesian position and orientation, for each joint j:

$$\boldsymbol{\mu}_i^j = \begin{bmatrix}\theta^j, & l^j\end{bmatrix}^\top \qquad \boldsymbol{\mu}_e^j = \begin{bmatrix}x^j, & y^j, & z^j, & q_w^j, & q_x^j, & q_y^j, & q_z^j\end{bmatrix}^\top = \begin{bmatrix}\boldsymbol{t}^j & \boldsymbol{q}^j\end{bmatrix}^\top. \tag{3}$$

The function $\boldsymbol{g}_e$ describes the 3D position and orientation of the subsequent link of a kinematic chain given the pose of the previous one. We can define the generative model $\boldsymbol{g}_e$ as in [20] (see A.1 for more details):

$$\boldsymbol{g}_e(\boldsymbol{\mu}_i^j, \boldsymbol{\mu}_e^{j-1}) = \begin{bmatrix}\boldsymbol{t}^{j-1} + \boldsymbol{h}(\boldsymbol{q}^{j-1} \cdot [0\ \boldsymbol{t}^j] \cdot \boldsymbol{q}^{j-1*}) \\ \boldsymbol{q}^{j-1} \cdot \boldsymbol{q}^j\end{bmatrix}, \tag{4}$$

where $\boldsymbol{q}^{j-1*}$ is the conjugate quaternion, $''\cdot''$ represents the Hamilton product, and $\boldsymbol{h}(\cdot)$ is a function that returns the imaginary coefficients of a quaternion. In our HAIF agent, the translation $\boldsymbol{t}^{j-1}$ and quaternion $\boldsymbol{q}^{j-1}$ are given by the extrinsic beliefs $\boldsymbol{\mu}_e^{j-1}$. The translation vector $\boldsymbol{t}^j$ and rotation $\boldsymbol{q}^j$ are instead dependent on the kinematic properties of the current link j and the joint angle and length $\theta^j,\ l^j$. The generative model can then be fully specified as a function of the intrinsic and extrinsic beliefs, and the gradients can be computed in closed:

$$\frac{\partial \boldsymbol{g}_e}{\partial \boldsymbol{\mu}_i} = \begin{bmatrix}\partial_\theta \boldsymbol{g}_e \\ \partial_l \boldsymbol{g}_e\end{bmatrix} \in \mathbb{R}^{2\times 7}, \qquad \frac{\partial \boldsymbol{g}_e}{\partial \boldsymbol{\mu}_e} = \begin{bmatrix}\partial_{x,y,z} \boldsymbol{g}_e \\ \partial_{\boldsymbol{q}} \boldsymbol{g}_e\end{bmatrix} \in \mathbb{R}^{7\times 7} \tag{5}$$

Thanks to the choice of using quaternions as singularity-free orientation representation, these gradients are easy to compute since the terms in the generative model are either linear or quadratic in the parameters, or they appear as arguments of sine and cosine functions.

2.3.2 Differential drive generative model

We now extend the HAIF for robot arm control to a differential drive robot. To do so, we write a simple kinematic model based on Euler's updates where the robot base position and orientation with respect to a world frame are expressed as:

$$\begin{cases} x_{t+1} = x_t + V_t \cos(\theta_t)\delta t \\ y_{t+1} = y_t + V_t \sin(\theta_t)\delta t \\ \theta_{t+1} = \theta_t + \omega_t \delta t, \end{cases} \tag{6}$$

where $V,\ \omega$ are respectively forward and rotational velocities. By considering small wheel increments $\Delta\phi_{R,L} = \phi_{\{R,L\}_t} - \phi_{\{R,L\}_{t-1}}$ in between timesteps where $\phi_{\{R,L\}}$ are the right and left wheel rotations, the expressions for forward and angular velocities result:

$$V_t = \frac{r}{2\delta t}\left(\Delta\phi_R + \Delta\phi_L\right) \tag{7}$$

$$\omega_t = \frac{r}{l\delta t}\left(\Delta\phi_R - \Delta\phi_L\right) \tag{8}$$

The terms $r,\ l$ are the wheel radius and distance respectively. The generative model for the differential drive HAIF is defined as a one-level hierarchical model where there are two controllable states, the wheel rotations:

$$\boldsymbol{g}_e(\boldsymbol{\mu}_i^j, \boldsymbol{\mu}_e^{j-1}) = \begin{bmatrix} x_{t-1} + \frac{r}{2}\left(\Delta\phi_R + \Delta\phi_L\right)\cos(\theta_{t-1}) \\ y_{t-1} + \frac{r}{2}\left(\Delta\phi_R + \Delta\phi_L\right)\sin(\theta_{t-1}) \\ \theta_{t-1} + \frac{r}{l}\left(\Delta\phi_R - \Delta\phi_L\right) \end{bmatrix}, \tag{9}$$

We set the intrinsic beliefs to be wheel rotations and extrinsic beliefs to be position $x - y$ and orientation θ with respect to a world frame:

$$\boldsymbol{\mu}_i = [\phi_R,\ \phi_L]^\top \qquad \boldsymbol{\mu}_e = [x,\quad y,\quad \theta]^\top . \tag{10}$$

The internal and external beliefs are then updated through the gradient of the generative model:

$$\frac{\partial \boldsymbol{g}_e}{\partial \boldsymbol{\mu}_i} = \begin{bmatrix} \partial_{\phi_R} \boldsymbol{g}_e \\ \partial_{\phi_L} \boldsymbol{g}_e \end{bmatrix} \in \mathbb{R}^{2\times 3}, \qquad \frac{\partial \boldsymbol{g}_e}{\partial \boldsymbol{\mu}_e} = \begin{bmatrix} \partial_x \boldsymbol{g}_e \\ \partial_y \boldsymbol{g}_e \\ \partial_\theta \boldsymbol{g}_e \end{bmatrix} \in \mathbb{R}^{3\times 3} \tag{11}$$

2.3.3 Whole-body generative model

The arm and base models can be combined into a single whole-body model by defining an overall hierarchical structure that combines the two. From eqs. (1) and (2), one can notice how the prediction errors at the level above $\boldsymbol{\varepsilon}_e^{j+1}$ influence the beliefs at the current level $\dot{\boldsymbol{\mu}}_i^j$ and $\dot{\boldsymbol{\mu}}_e^j$. By this logic, the extrinsic prediction errors of the first level of the hierarchy will not have any influence since there is no level left below in the chain. However, we can propagate these errors back to the top level of the hierarchy of the mobile base kinematic model. By doing so, the base can further minimize free energy by moving its wheels. In turn, we can propagate up from the base kinematic model the predictions about the first link's position and orientation of the robot arm, closing the loop. Mathematically, all equations remain the same apart from the one corresponding to the update of internal beliefs $\dot{\boldsymbol{\mu}}_i^j$ for the base, which becomes:

$$\dot{\boldsymbol{\mu}}_i^0 = \begin{bmatrix} \boldsymbol{\mu}_i^{0'} + \pi_p^0 \boldsymbol{\varepsilon}_p^0 + \partial_{\boldsymbol{\mu}_i} \boldsymbol{g}_e^\top \pi_e^1 (\kappa_{base} \boldsymbol{\varepsilon}_{e,base}^1 + \kappa_{arm} \boldsymbol{\varepsilon}_{e,arm}^2) + \partial \boldsymbol{f}_i^{0\top} \pi_{\boldsymbol{\mu}_i}^0 \boldsymbol{\varepsilon}_{\boldsymbol{\mu}_i}^0 \\ -\pi_{\boldsymbol{\mu}_i}^0 \boldsymbol{\varepsilon}_{\boldsymbol{\mu}_i}^0 \end{bmatrix} \tag{12}$$

where κ_{base} and κ_{arm} are tuning parameters to weight the effect of arm and base prediction errors. By tuning these parameters, one can shape robot behavior, for instance, to allow more or less base response due to the arm's extrinsic prediction errors. The equation for $\dot{\boldsymbol{\mu}}_e^0$ remains the same since the gradient with respect to extrinsic beliefs is zero. This is because the values $x_{t-1},\ y_{t-1},\ \theta_{t-1}$ are simply constants from the previous time step and not beliefs from the level below.

This simple change allows using the base motion to minimize the arm's prediction errors, extending the arm's reachability beyond its stationary workspace. This is crucial for the successful completion of the Habitat Benchmark. Additionally, we still preserve the ability to send individual goals to the base as explained below.

2.3.4 Goals, obstacles, and control

To realize goal-directed behavior, we can define attractive goals and repulsive forces as in [22, 20]. Goals can be both intrinsic (joint positions) or extrinsic (Cartesian poses) for the arm and base, and they can be combined to define future desired states $\boldsymbol{\mu}^*$. Goals act as attractors, forming dynamic functions $\boldsymbol{f}_a = \kappa_a(\boldsymbol{\mu}^* - \boldsymbol{\mu})$ that linearly minimize the distance between the desired and current states. The desired states can be defined flexibly in terms of the current beliefs as

$$\boldsymbol{\mu}^* = N\boldsymbol{\mu} + \boldsymbol{n}^*, \tag{13}$$

where N achieves dynamic behaviors, such as keeping a limb vertical by imposing the $x,\ y$ coordinates of a link to be the same as the previous one, while $\boldsymbol{n}^*$ imposes an attractor to a static configuration. Some examples of basic goals that can be given to the mobile manipulator in the Habitat Benchmark are 1) **End-effector** goal: the robot will use its whole body to reach a target $(x^*,\ y^*,\ z^*)$ position, 2) **Base goal**: the robot will move its base to reach a goal $(x^*,\ y^*,\ \theta^*)$, where θ^* can be updated over time such that the robot faces the goal $\theta_t^* = \arctan 2(y^* - y_t, x^* - x_t)$, 3) **Arm joint goal**: the robot will reach a specific joint configuration, 4) **Combinations of the above**: the user can mix goals for example making the base move while keeping the arm in a certain joint configuration. The same idea of attractive forces can be used for collision avoidance through repulsive forces, where a repulsive state $\boldsymbol{\mu}^!$ has to be avoided (see appendix A.2).

Given a goal, the control action is computed by minimizing the proprioceptive component of the free energy with respect to the control signals [22]:

$$\dot{\boldsymbol{a}} = -\partial_a \mathcal{F}_p = -\partial_a \tilde{\boldsymbol{s}}_p \pi_p \tilde{\boldsymbol{\varepsilon}}_p \tag{14}$$

where $-\partial_a \tilde{\boldsymbol{s}}_p$ is the partial derivative of proprioceptive observations with respect to the control, and $\tilde{\boldsymbol{\varepsilon}}_p = \tilde{\boldsymbol{s}}_p - \boldsymbol{g}_p(\boldsymbol{\mu})$ are the generalized proprioceptive prediction errors.

2.4 Whole-body high-level skills for objects rearrangement

Similarly to the chosen Habitat Baseline [10], we define a set of abstract high-level skills that the high-level planner can sequence at runtime. These skills are implemented with the whole-body controller, and are divided into `Pick`, `Place`, `Move`, `PickFromFridge`, `PickFromDrawer`. The skills are defined as a fixed sequence of goals given to the whole-body controller to realize an overall behavior. Every skill computes a whole-body control action for the robot. Details about the skills can be found in appendix A.3.

3 Experiments

3.1 Experiments setup

Agent embodiment: The Habitat Benchmark employs a Fetch robot that features a mobile wheeled base, a 7DoF robotic arm, and a parallel-jaw gripper. It is equipped with two RGB-D cameras with a resolution of 128×128 pixels on both the arm and the head. The robot perceives its state through proprioceptive sensing, which includes joint angles of the arm and the Cartesian coordinates of the end-effector. The robot can also sense the goal positions (3DoF), as well as a scalar to indicate whether an object is held. Obstacle positions are sensed by querying the map model.

Action space: The action space is a 10DoF continuous space, for whole-body control. It is composed of forward and angular base velocities, a 7DoF arm velocity action, and a 1DoF gripper action. Grasping is abstract as in previous work [28, 10], such that if the gripper action is positive, the object closest to the end-effector within 15cm will be snapped to the gripper. An object is instead released by a negative action.

Evaluation metrics Each task in the Habitat Benchmark is composed of a sequence of subtasks that must be completed. As in previous work [28, 10], we measure performance by reporting the completion rate at each subtask stage, with the success rate of the final subtask representing the overall task success. Notably, if the previous subtask has failed, the current subtask is also considered a failure independently of its outcome. At the start of each evaluation episode, the robot's base is placed at a random position and orientation, ensuring no collisions, while the arm begins in a resting configuration.

Baselines We compare our model against methods in [10], namely a Monolithic RL approach and a Multi-skill RL Mobile manipulation (MM). The latter had the best performance among several learning-based and classical task and motion planning approaches. The Monolithic RL is an end-to-end RL policy for each complete task (`TidyHouse`, `PrepareGroceries`, and `SetTable`). Different reward functions are selected according to oracle knowledge about the current subtask being executed, such as picking or placing, to train a single policy for such long-horizon tasks. The Multi-skill RL Mobile manipulation approach, instead, trains different mobile manipulation policies that are then chained by an oracle task planner executed in open loop. For details about the baselines, we refer the reader to [10].

3.2 Results

We report the benchmark results in Fig. 4. Our approach outperforms the baselines in all three tasks, averaging 72.5% completion rate in `TidyHouse`, 77% completion rate in `PrepareGroceries`, and 50% completion rate in `SetTable` over five seeds. The best performing baseline, MM, instead, averages 71%, 64%, and 29% respectively. Considering all three tasks combined, we achieved a 66.5% success rate compared to the 54.7% of the MM baseline. Notably, the MM baseline requires extensive offline training. That is 6400 episodes per task across varied layouts and configurations in the Habitat environments, and 100 million steps per skill across a total of 7 skills. In contrast, our method relies on hand-tuning each skill over just a handful of episodes and is evaluated directly on unseen layouts and configurations, demonstrating strong generalization without the need for data-intensive training. However, we still rely on privileged information, such as the floor map for path planning and articulated object states. These assumptions will be removed in future work.

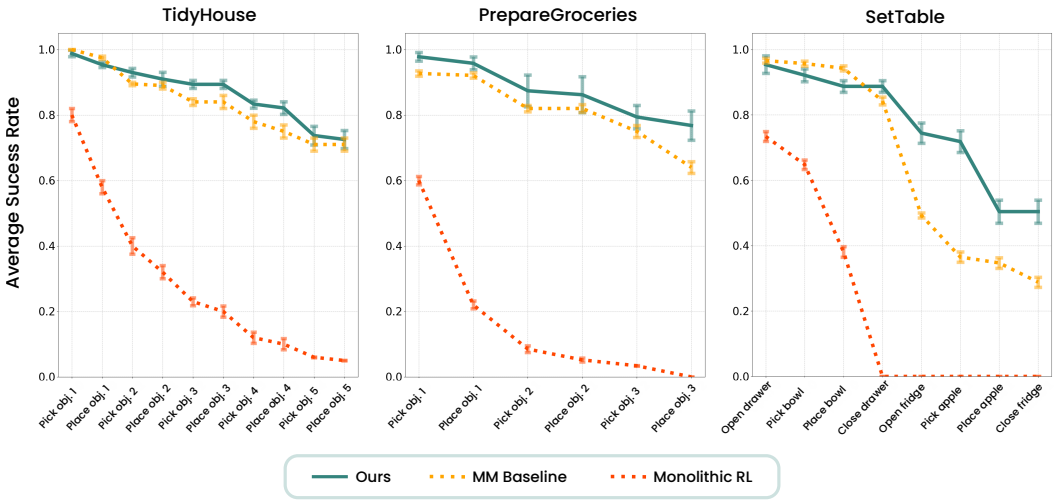

Figure 4: Evaluation results on the Habitat Benchmark, averaged over 100 episodes. Each task is evaluated on different apartment layouts and divided into stages. For a stage to be successful, all previous stages must also be successful. **TidyHouse**: Evaluated over five pick-and-place stages, from `Pick obj. 1` to `Place obj. 5`. **PrepareGroceries**: Measured from `Pick obj. 1` to `Place obj. 3`. **SetTable**: Involves a more complex sequence including `Open drawer` → `Pick bowl` → `Place bowl` → `Close drawer`, and similarly for the fridge and apple.

4 Discussion and Conclusion

In this work, we proposed a hierarchical active inference model to address the Habitat Benchmark, surpassing state-of-the-art performance across all three benchmark tasks. Our system is composed of two key components: a high-level model that selects appropriate low-level skills based on discrete observations, and a set of low-level skills defined through goals for a novel whole-body controller using hierarchical active inference. This architecture enables the system to flexibly adapt to task failures and dynamically adjust behavior in response to environmental changes. Importantly, our method operates online, without requiring offline training, and supports real-time adaptation of the high-level plan. While our current implementation still relies on certain privileged information (such as access to a global map for path planning), our future work will focus on enabling the agent to actively explore and construct maps in real-time. Additionally, knowledge of the state of articulated objects, such as drawers and refrigerators, will be inferred directly from raw RGBD sensory input. Moreover, while low-level skills are currently composed of a fixed sequence of goals for the continuous whole-body controller, one could add an intermediate hierarchical level as in [21] to smoothly transition between subgoals. Another interesting direction to explore would be learning skills directly from demonstration. In summary, our hierarchical hybrid active inference model demonstrates promising results in goal-directed robotic control within complex environments. With further enhancements in perception, exploration, and skill acquisition, we believe this framework could serve as a foundation for more generalized and scalable robotic agents.

References

[1] Beal, M.J.: Variational Algorithms for Approximate Bayesian Inference. Ph.D. thesis, University College London (2003)

[2] Bishop, C.M.: Pattern Recognition and Machine Learning. Springer (2006)

[3] Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: A review for statisticians. Journal of the American Statistical Association **112**(518), 859–877 (2017). https://doi.org/10.1080/01621459.2017.1285773, `https://doi.org/10.1080/01621459.2017.1285773`

[4] Çatal, O., Van de Maele, T., Pitliya, R.J., Albarracin, M., Pattisapu, C., Verbelen, T.: Belief sharing: A blessing or a curse. In: Buckley, C.L., Cialfi, D., Lanillos, P., Pitliya, R.J., Sajid, N.,

Shimazaki, H., Verbelen, T., Wisse, M. (eds.) Active Inference. pp. 121–133. Springer Nature Switzerland, Cham (2025)

[5] Ehsani, K., Han, W., Herrasti, A., VanderBilt, E., Weihs, L., Kolve, E., Kembhavi, A., Mottaghi, R.: Manipulathor: A framework for visual object manipulation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4497–4506 (2021)

[6] Friston, K.J., Da Costa, L., Tschantz, A., Kiefer, A., Salvatori, T., Neacsu, V., Koudahl, M., Heins, C., Sajid, N., Markovic, D., Parr, T., Verbelen, T., Buckley, C.L.: Supervised structure learning. Biological Psychology **193**, 108891 (2024). https://doi.org/https://doi.org/10.1016/j.biopsycho.2024.108891, https://www.sciencedirect.com/science/article/pii/S0301051124001510

[7] Friston, K.J., Parr, T., de Vries, B.: The graphical brain: Belief propagation and active inference. Network neuroscience **1**(4), 381–414 (2017)

[8] Gan, C., Zhou, S., Schwartz, J., Alter, S., Bhandwaldar, A., Gutfreund, D., Yamins, D.L., DiCarlo, J.J., McDermott, J., Torralba, A., et al.: The threedworld transport challenge: A visually guided task-and-motion planning benchmark towards physically realistic embodied ai. In: 2022 International conference on robotics and automation (ICRA). pp. 8847–8854. IEEE (2022)

[9] Garrett, C.R., Lozano-Pérez, T., Kaelbling, L.P.: Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In: Proceedings of the international conference on automated planning and scheduling. vol. 30, pp. 440–448 (2020)

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

[11] Hart, P.E., Nilsson, N.J., Raphael, B.: A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics **4**(2), 100–107 (1968). https://doi.org/10.1109/TSSC.1968.300136

[12] Kaelbling, L.P., Lozano-Pérez, T.: Hierarchical task and motion planning in the now. In: 2011 IEEE International Conference on Robotics and Automation. pp. 1470–1477. IEEE (2011)

[13] Kaelbling, L.P., Lozano-Pérez, T.: Integrated task and motion planning in belief space. The International Journal of Robotics Research **32**(9-10), 1194–1227 (2013)

[14] Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023), https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

[15] Van de Maele, T., Çatal, O., Tschantz, A., Buckley, C.L., Verbelen, T.: Variational bayes gaussian splatting (2024)

[16] Parr, T., Friston, K.J.: Active inference and the anatomy of oculomotion. Neuropsychologia **111**, 334–343 (2018)

[17] Parr, T., Friston, K.J.: The discrete and continuous brain: from decisions to movement—and back again. Neural computation **30**(9), 2319–2347 (2018)

[18] Parr, T., Pezzulo, G., Friston, K.J.: Active inference: the free energy principle in mind, brain, and behavior. MIT Press (2022)

[19] Parr, T., Pezzulo, Friston, K.J., Giovanni: Generative models for sequential dynamics in active inference. Springer Nature Link (2023)

[20] Pezzato, C., Buckley, C., Verbelen, T.: Why learn if you can infer? robot arm control with hierarchical active inference. In: The First Workshop on NeuroAI@ NeurIPS2024 (2024)

[21] Priorelli, M., Stoianov, I.P.: Deep hybrid models: Infer and plan in a dynamic world. Entropy **27**, 570 (2025). https://doi.org/10.3390/e27060570, https://doi.org/10.3390/e27060570

[22] Priorelli, M., Pezzulo, G., Stoianov, I.P.: Deep kinematic inference affords efficient and scalable control of bodily movements. Proceedings of the National Academy of Sciences **120**(51), e2309058120 (Dec 2023). https://doi.org/10.1073/pnas.2309058120, `https://pnas.org/doi/10.1073/pnas.2309058120`

[23] Priorelli, M., Stoianov, I.P.: Slow but flexible or fast but rigid? discrete and continuous processes compared. Heliyon **10**(20) (2024)

[24] Priorelli, M., Stoianov, I.P.: Dynamic planning in hierarchical active inference. Neural Networks p. 107075 (2025)

[25] Sandakalum, T., Ang Jr, M.H.: Motion planning for mobile manipulators—a systematic review. Machines **10**(2), 97 (2022)

[26] Smith, R., Friston, K.J., Whyte, C.J.: A step-by-step tutorial on active inference and its application to empirical data. PubMed (2022)

[27] Srivastava, S., Fang, E., Riano, L., Chitnis, R., Russell, S., Abbeel, P.: Combined task and motion planning through an extensible planner-independent interface layer. In: 2014 IEEE international conference on robotics and automation (ICRA). pp. 639–646. IEEE (2014)

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

A Appendix

A.1 Kinematic generative model

To generative model in eq. (4) is defined from generic transformation matrices. To compute the absolute position and orientation of the current link j given the absolute position and orientation of the previous one, we can write:

$$ {}^{w}T_j = \left[\begin{array}{c|c} R^{j-1}R^j & \boldsymbol{t}^{j-1} + R^{j-1}\boldsymbol{t}^j \\ \hline \mathbf{0} & 1 \end{array}\right], \tag{15} $$

where w indicates the world frame as an absolute reference, R represents a rotation matrix, and $\boldsymbol{t}$ a translation vector. The world frame can be the base link of a robot arm. From eq. (15), we note that the resulting absolute rotation of a link is the multiplication of two rotation matrices. However, we can express this as a quaternion multiplication $\boldsymbol{q}^{j-1} \cdot \boldsymbol{q}^j$. Similarly, we can rotate a vector $\boldsymbol{t}^j$ by a quaternion $\boldsymbol{q}^{j-1}$ corresponding to R^{j-1}, leading to eq. (4). Considering a generic Denavit–Hartenberg (DH) transformation matrix

$$ \left[\begin{array}{ccc|c} \cos\theta^j & -\sin\theta^j\cos\alpha^j & \sin\theta^j\sin\alpha^j & l^j\cos\theta^j \\ \sin\theta^j & \cos\theta^j\cos\alpha^j & -\cos\theta^j\sin\alpha^j & l^j\sin\theta^j \\ 0 & \sin\alpha^j & \cos\alpha^j & d^j \\ \hline 0 & 0 & 0 & 1 \end{array}\right], \tag{16} $$

we note that the translation vector is simply $\boldsymbol{t}^j = [l^j\cos\theta^j, l^j\sin\theta^j, d^j]$. According to the DH convention, the rotational part of the transformation matrix is the composition of a rotation θ^j about the previous z-axis and a rotation of α^j around the x-axis. We can then write:

$$ \boldsymbol{q}^j = \left[\cos\tfrac{\theta^j}{2}\cos\tfrac{\alpha^j}{2},\quad \cos\tfrac{\theta^j}{2}\cos\tfrac{\alpha^j}{2},\quad \cos\tfrac{\theta^j}{2}\cos\tfrac{\alpha^j}{2},\quad \cos\tfrac{\theta^j}{2}\cos\tfrac{\alpha^j}{2}\right]. \tag{17} $$

The generic kinematic model in eq. (4) can be expressed as

$$ \boldsymbol{g}_e(\boldsymbol{\mu}_i^j, \boldsymbol{\mu}_e^{j-1}) = \begin{bmatrix} \boldsymbol{t}^{j-1} + \boldsymbol{h}(\boldsymbol{q}^{j-1}\cdot[0\ \boldsymbol{t}^j]\cdot\boldsymbol{q}^{j-1*}) \\ \boldsymbol{q}^{j-1}\cdot\boldsymbol{q}^j \end{bmatrix}, = \begin{bmatrix} x^{j-1} + x_{tf} \\ y^{j-1} + y_{tf} \\ z^{j-1} + z_{tf} \\ q_{w,\,tf} \\ q_{x,\,tf} \\ q_{y,\,tf} \\ q_{z,\,tf} \end{bmatrix}, \tag{18} $$

where x_{tf}, y_{tf}, z_{tf} and $q_{*,\,tf}$ are the transformed translations and rotation. Computing the Hamilton products yields the following expressions for the transformed positions

$$ \begin{aligned} x_{tf} &= {q_w^{j-1}}^2 l^j\cos\theta^j + {q_x^{j-1}}^2 l^j\cos\theta^j - {q_y^{j-1}}^2 l^j\cos\theta^j - {q_z^{j-1}}^2 l^j\cos\theta^j \\ &\quad + 2q_x^{j-1}q_y^{j-1}l^j\sin\theta^j + 2q_x^{j-1}q_z^{j-1}d^j + 2q_w^{j-1}q_y^{j-1}d^j - 2q_w^{j-1}q_z^{j-1}l^j\sin\theta^j, \\ y_{tf} &= {q_w^{j-1}}^2 l^j\sin\theta^j - {q_x^{j-1}}^2 l^j\sin\theta^j + {q_y^{j-1}}^2 l^j\sin\theta^j - {q_z^{j-1}}^2 l^j\sin\theta^j \\ &\quad + 2q_x^{j-1}q_y^{j-1}l^j\cos\theta^j + 2q_y^{j-1}q_z^{j-1}d^j - 2q_w^{j-1}q_x^{j-1}d^j + 2q_w^{j-1}q_z^{j-1}l^j\cos\theta^j, \\ z_{tf} &= {q_w^{j-1}}^2 d^j - {q_x^{j-1}}^2 d^j - {q_y^{j-1}}^2 d^j + {q_z^{j-1}}^2 d^j \\ &\quad + 2q_x^{j-1}q_z^{j-1}l^j\cos\theta^j + 2q_y^{j-1}q_z^{j-1}l^j\sin\theta^j - 2q_w^{j-1}q_y^{j-1}l^j\cos\theta^j + 2q_w^{j-1}q_x^{j-1}l^j\sin\theta^j, \end{aligned} $$

and orientation:

$$ \begin{aligned} q_{w,\,tf} &= q_w^{j-1}\cos\frac{\theta^j}{2}\cos\frac{\alpha^j}{2} - q_x^{j-1}\cos\frac{\theta^j}{2}\sin\frac{\alpha^j}{2} - q_y^{j-1}\sin\frac{\theta^j}{2}\sin\frac{\alpha^j}{2} - q_z^{j-1}\sin\frac{\theta^j}{2}\cos\frac{\alpha^j}{2}, \\ q_{x,\,tf} &= q_w^{j-1}\cos\frac{\theta^j}{2}\sin\frac{\alpha^j}{2} + q_x^{j-1}\cos\frac{\theta^j}{2}\cos\frac{\alpha^j}{2} + q_y^{j-1}\sin\frac{\theta^j}{2}\cos\frac{\alpha^j}{2} - q_z^{j-1}\sin\frac{\theta^j}{2}\sin\frac{\alpha^j}{2}, \\ q_{y,\,tf} &= q_w^{j-1}\sin\frac{\theta^j}{2}\sin\frac{\alpha^j}{2} - q_x^{j-1}\sin\frac{\theta^j}{2}\cos\frac{\alpha^j}{2} + q_y^{j-1}\cos\frac{\theta^j}{2}\cos\frac{\alpha^j}{2} + q_z^{j-1}\cos\frac{\theta^j}{2}\sin\frac{\alpha^j}{2}, \\ q_{z,\,tf} &= q_w^{j-1}\sin\frac{\theta^j}{2}\cos\frac{\alpha^j}{2} + q_x^{j-1}\sin\frac{\theta^j}{2}\sin\frac{\alpha^j}{2} - q_y^{j-1}\cos\frac{\theta^j}{2}\sin\frac{\alpha^j}{2} + q_z^{j-1}\cos\frac{\theta^j}{2}\cos\frac{\alpha^j}{2}. \end{aligned} $$

A.2 Collision avoidance in HAIF

A repulsive state $\boldsymbol{\mu}^!$ can be imposed on intrinsic beliefs, to realize joint limit avoidance, or extrinsic beliefs for collision avoidance with the environment. We define joint limit avoidance as:

$$\boldsymbol{f}_{r,\theta}(\boldsymbol{\mu}) = \begin{cases} 0, & \text{if } ||\boldsymbol{e}_\theta|| > \gamma_\theta \\ k_{r,\theta}\zeta(1/\gamma_\theta - 1/||\boldsymbol{e}_\theta||), & \text{otherwise} \end{cases}, \tag{19}$$

where $\boldsymbol{e}_\theta = \boldsymbol{\mu}^!_\theta - \boldsymbol{\mu}_\theta$, $\boldsymbol{\mu}_\theta$ is the slice of beliefs about joint angles, $\boldsymbol{\mu}^!_\theta$ are the joint limits, and γ_θ is a chosen threshold. The variable $\zeta \in \{-1,\ 1\}$ is negative for lower limits and positive for upper limits. The collision avoidance strategy is instead the same as [22]:

$$\boldsymbol{f}_{r,obst}(\boldsymbol{\mu}) = \begin{cases} 0, & \text{if } ||\boldsymbol{e}_{obst}|| > \gamma_{obst} \\ k_{r,obst}(1/\gamma_{obst} - 1/||\boldsymbol{e}_{obst}||)\boldsymbol{e}_{obst}/||\boldsymbol{e}_{obst}||^3, & \text{otherwise} \end{cases}, \tag{20}$$

where $\boldsymbol{e}_{obst} = \boldsymbol{\mu}^!_{pos} - \boldsymbol{\mu}_{pos}$, $\boldsymbol{\mu}_{pos}$ is the slice of beliefs about link positions, and $\boldsymbol{\mu}^!_{pos}$ is the position of an obstacle given by the VBGS module. Goal attractors and repulsive forces for joint limits and collision avoidance are then summed together to form the dynamics function of a single level. This allows one to achieve behaviors such as reaching a target while avoiding an obstacle. Parameters are manually chosen to achieve sufficient performance in the test cases, but could be automatically optimized.

A.3 Whole-body Skills

The routines for the different skills for the mobile manipulator are defined as follows:

- `Pick`: The robot unfolds its arm (joint goal), moves to a pre-grasp position above the target object (end-effector + joint goal), and then proceeds to the grasp pose to perform the grasp once close enough (end-effector goal). After grasping, it retreats to the post-grasp pose (end-effector goal) and folds the arm back into a compact configuration (joint goal) (see Fig. 5).

Figure 5: Evolution of the `Pick` skill over time.

- `Place`: It mirrors the `Pick` sequence, but targets a specified place location.
- `PickFromDrawer`: The end-effector is moved in front of the drawer hinge and grasps the handle once close enough (end-effector + joint goal). Then, the robot executes a linear backward trajectory to pull the drawer open (end-effector goal). The object is picked as in `Pick`. Finally, the robot end-effector is placed in front of the handle again (end-effector goal), and the drawer is pushed close following a linear trajectory (end-effector goal) (see Fig. 6).
- `PickFromFridge`: The robot unfolds its arm (joint goal), moves in front of the fridge handle, and grasps it once close enough (end-effector goal). It then follows a circular trajectory to partially open the door (end-effector goal). After that, the arm retreats (joint goal), and finally, the arm starts a linear trajectory from behind the half-opened door to push it to fully open (end-effector goal). The object is picked as in `Pick`, and then the robot first moves to the left of the fridge door (base + joint goal), and after it follows a linear trajectory to push the door closed (end-effector goal) (see Fig. 7).
- `Move`: The `NavModel` computes a global path towards a final goal and orientation, and provides the move skill with the current active subgoal $(x^*,\ y^*)$, along with the final desired position and orientation. At each step, the heading θ^* toward the subgoal is computed. A predefined joint configuration (joint goal) for the arm is set to avoid collisions. The skill

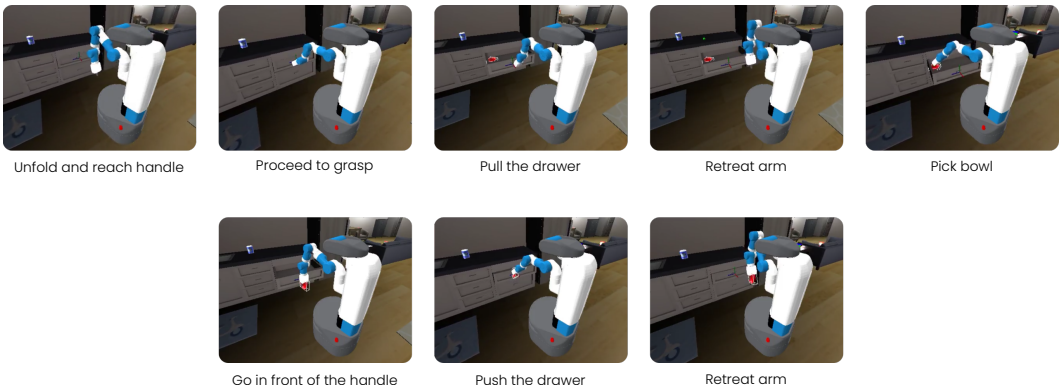

Figure 6: Evolution of the `PickFromDrawer` skill over time.

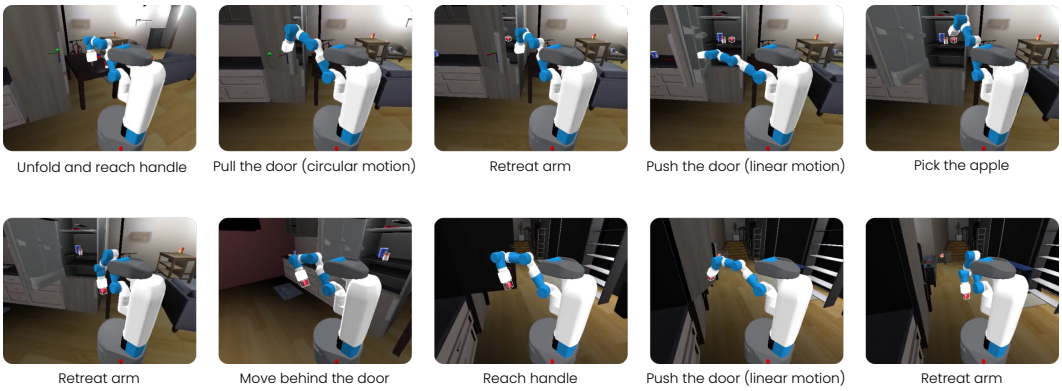

Figure 7: Evolution of the `PickFromFridge` skill over time.

terminates when the robot is within a threshold distance of the target pose. See Fig. 8) for an example. The reach threshold for the position is kept at 0.8m while the one for orientation to 0.3rad. These are particularly loose since we rely on whole-body manipulation skills and are not required to precisely position the base before executing them.

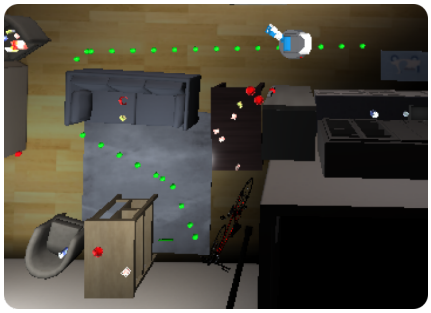

Figure 8: Top view of the `Move` skill where the robot moves through subgoals following the global path.

A.4 Example evolution of a probabilistic map

In Fig. 9 we present an example of how a probabilistic map can evolve through time using VBGS.

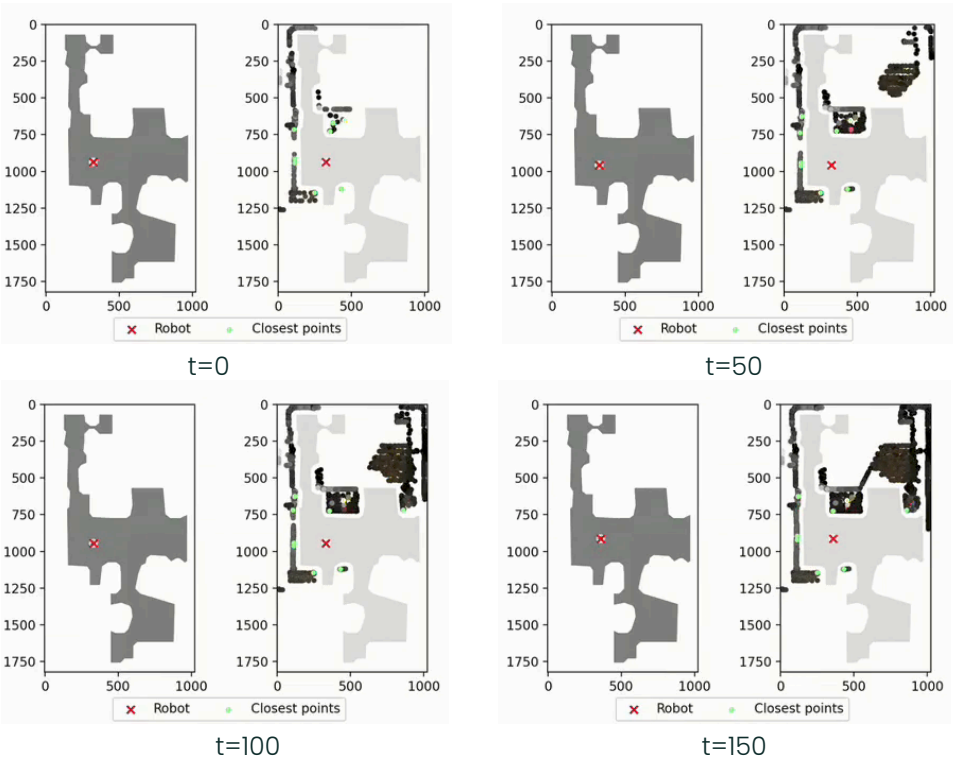

Figure 9: An example of the probabilistic map evolution with VBGS in one Habitat apartment. The left side of each panel shows the location of the robot on the ground truth floor plan. The right side overlays the Gaussian components over the obstacles projected onto the floor.