UNIETP: UNIFYING ENVIRONMENTS FOR GENERALIZABLE EMBODIED TASK PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper focuses on the problem of Embodied Task Planning, where an agent is required to execute a sequence of atomic actions within an interactive environment to complete a user-specified task. Though a variety of simulators and datasets have previously been built for this task, these efforts are largely isolated, with each using its own observation format, action type, and task domain. This fragmentation complicates comprehensive model evaluation and hinders the scalability of training data. As an effort towards generalizable embodied planning, we propose UniETP, a unified interface integrating four commonly used simulators (AI2-THOR, VirtualHome, Habitat, BEHAVIOR). UniETP is characterized by both standardization and diversity. On one hand, it formalizes all the simulators into a consistent observation and action space, and builds an evaluation system to support complicated task goals. On the other hand, it enhances task diversity and complexity across dimensions such as task logic, instance grounding, and instruction understanding, constructing a new dataset with varied levels of difficulty automatically. Extensive experiments on the proposed benchmark are conducted to evaluate the embodied planning capabilities of recent models and validate the utility of our framework.

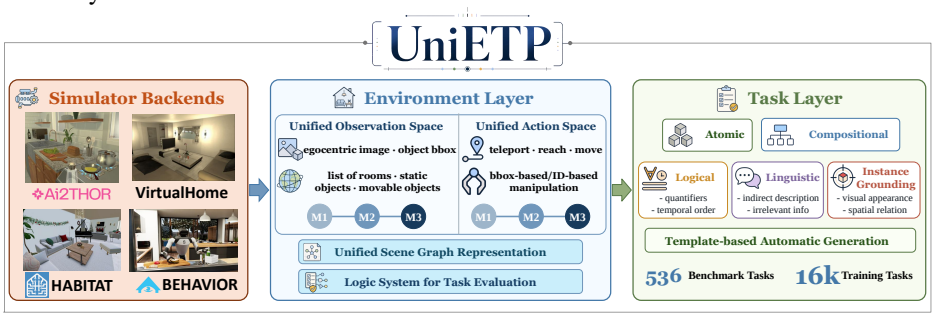


Figure 1: We propose UniETP, an embodied task planning benchmark consisting of an environment layer that unifies four simulators with standardized evaluation protocol, and a task layer that supports automatic generation of various task instances.

1 INTRODUCTION

Recent advances in embodied intelligence have increasingly adopted a hierarchical architecture, in which high-level modules are responsible for task understanding and planning, while low-level controllers handle fine-grained motor execution (Abdolmaleki et al., 2025; Shi et al., 2025b; Gao et al., 2025a). The problem of **Embodied Task Planning** (ETP) focuses on the former. It abstracts away detailed control and formulates decision-making at the level of atomic actions, requiring a model to interpret natural language task instructions and generate an appropriate sequence of actions to be executed in an interactive environment. By bridging language understanding and action generation, ETP plays an important role in enabling embodied agents to accomplish complex, long-horizon tasks. As Vision–Language Models (VLMs) continue to advance, the study of ETP has attracted growing attention as a key component in developing more capable and flexible robotic systems.

In recent years, a series of simulator environments (Kolve et al., 2017; Puig et al., 2024; 2018; Srivastava et al., 2022) has been proposed or re-purposed for the problem of ETP, and many benchmarks and datasets have been built upon them (Shridhar et al., 2021; Liu et al., 2025c; Li et al., 2024b; Yang et al., 2025b; Chang et al., 2025). Despite the substantial progress enabled by these

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

efforts, **isolation** across environments remains a critical challenge for embodied planning. Different simulators typically expose distinct environment information and implement actions in different ways. As a result, datasets built upon them often adopt different observation and action spaces, require planners to use different forms of inputs and outputs, and have different evaluation protocols or implementation details. Consequently, agents designed or tuned for one dataset are often difficult to transfer to another. Moreover, simulators are complementary in terms of scene scale, visual appearance, physical rules, object states, and supported interactions. Such isolation therefore also hinders the expansion of data scale and diversity, which are crucial for improving the generalization ability of embodied agents. Bridging the gap between simulators would allow models to be trained and evaluated on a more diverse set of tasks.

To address these challenges, we propose **UniETP**, a unified benchmark for developing and evaluating generalizable embodied agents. UniETP is organized into two levels: the environment layer and the task layer. At the **environment level**, UniETP unifies four widely used simulation environments through a Python-based environment interface that decouples agents from simulator-specific implementations. It further defines standardized, multi-granularity observation and action spaces, and adopts a unified scene graph representation as an environment abstraction to support unified evaluation logic. In this way, agents can operate under a consistent interface to complete tasks across different simulators.

Building on this, the **task level** supports the construction of diverse tasks with varying levels of difficulty. We define 100+ types of task templates, fully leveraging the rich object states and tool functionalities supported by the four simulators. Based on these templates, we develop an automatic task generation pipeline with the help of a commonsense knowledge base and a large language model (LLM) annotator. The resulting benchmark contains over 500 task instances, covering both simple atomic tasks and long-horizon compositional tasks. In designing these tasks, we pay particular attention to several capability dimensions, including instruction understanding, where instructions may involve complex references, syntactic structures, and distracting information; complex logic, where task goals may include temporal constraints, universal quantification, and other logical structures that go beyond the goal-state-based evaluation; instance-level grounding, where target objects are constrained to specific instances through states, spatial relations, or other conditions, in contrast to the category-level requirements adopted in many existing benchmarks.

In summary, the contributions of this paper are as follows: (1) We unify four mainstream embodied simulators through a standardized environment interface, enabling the development of simulator-agnostic agents. (2) We implement automatic generation of diverse tasks, allowing agents to be evaluated in a fine-grained manner across rich tasks, instructions, and scenes. (3) We conduct extensive experiments on the proposed benchmark to examine the performance of recent VLMs in embodied planning.

2 Related Works

2.1 Embodied Agent

Recent progress in foundation models has led to rapid advances in embodied intelligence, spanning manipulation (Liang et al., 2023; Zitkovich et al., 2023; Intelligence et al., 2025), navigation (Zheng et al., 2024; Zhou et al., 2024; Zhang et al., 2025a), planning (Huang et al., 2022; Li et al., 2022b; Zhao et al., 2023), and reasoning (Majumdar et al., 2024; Chen et al., 2024; Zhao et al., 2025). In particular, a growing body of work employs large models in an agentic manner, using them as high-level decision-making modules that serve as the "brain" of embodied agents and invoke navigation or control skills to accomplish complex tasks. Existing studies have explored designing environment and action representations (Shah et al., 2025; Lou et al., 2025; Liu et al., 2025a; Seo et al., 2025; Huang et al., 2025), incorporating memory modules and reflection mechanisms (Huang et al., 2023; Song et al., 2023; Shinn et al., 2023), improving training strategies (Zhang et al., 2025c; Shi et al., 2025a; Wang et al., 2025a; Xu et al., 2025a; Tian et al., 2025; Xu et al., 2026), and building embodied foundation models (Abdolmaleki et al., 2025; Team et al., 2025; Azzolini et al., 2025).

2.2 Simulators and Benchmarks for Embodied Task Planning

Environments are essential for the evaluation of embodied planners as it requires closed-loop interactions. The basis of such environments is interactive 3D simulators with scene and object assets. AI2-THOR (Kolve et al., 2017) provides near-photorealistic indoor scenes with rich object properties and affordances, and supports large-scale procedural scene generation and embodied manipulation. VirtualHome (Puig et al., 2018) focuses on simulating household activities through executable

Table 1: A comparison with previous benchmarks. UniETP unifies four popular embodied simulators (**A**I2-THOR, **V**irtualHome, **H**abitat, **B**EHAVIOR; **O** is for others). Combining their advantages, it supports visual modality (vis.) with rich object states (states), and a wide range of tasks (#task types). It defines an action space of multiple levels of granularity (mul. act.) to support evaluation at different difficulty levels. We also propose an automatic task generation pipeline (auto. gen.), covering tasks with sophisticated instructions (instr.), complex goal logic (logic), instance-level grounding constraints (inst.). A more detailed description can be found in the Appendix. *For some benchmarks that contain multiple subsets focusing on different domains, we only consider the subsets related to ETP.

Benchmark	sim.	vis.	#task types	states	instr.	logic	inst.	mul. act.	auto. gen.
ALFRED	A	✓	7	✓	✓	✗	✗	✗	✗
ALFWorld	A	✓	6	✓	✗	✗	✗	✗	✗
VisualAgentBench*	B	✓	45	✓	✗	✓	✗	✗	✗
LoTa-Bench	A,V	✗	7+5	✓	✗	✗	✗	✗	✗
ET-Plan-Bench	V	✗	6	✗	✗	✓	✓	✗	✓
LEGENT	O	✓	6	✗	✗	✗	✗	✗	✓
LangR	H	✓	10	✗	✓	✓	✓	✗	✓
EAI	V,B	✗	26+100	✓	✗	✓	✗	✗	✗
EmbodiedBench*	A,H	✓	7+10	✓	✓	✓	✓	✗	✗
PARTNR	H	✓	-	✗	✓	✓	✓	✗	✓
UniETP	A,V,H,B	✓	138	✓	✓	✓	✓	✓	✓

programs, offering a structured abstraction of human daily routines and object interactions. Habitat (Puig et al., 2024) emphasizes scalable embodied AI research with efficient photorealistic simulation, configurable sensors, and support for diverse scene datasets. BEHAVIOR (Li et al., 2024a) targets human-centered household activities with physical realism, including articulated objects, deformable bodies, and liquids. Other simulators (Gan et al., 2021; Wang et al., 2024; Cai et al., 2025; Rana et al., 2023) further broaden the landscape by focusing on larger-scale or domain-specific scenes.

On top of these simulators, embodied task planning benchmarks define evaluation protocols and task specifications that determine how agents interact with environments and what capabilities are tested. For example, ALFRED (Shridhar et al., 2020) introduces instruction following based on AI2-THOR, while ALFWorld (Shridhar et al., 2021) further abstracts it with the TextWorld engine and defines a text-based agent interface. VisualAgentBench (Liu et al., 2025c) considers various visual task domains for agents, including an embodied domain adapted from BEHAVIOR. LoTa-Bench (Choi et al., 2024) combines AI2-THOR and VirtualHome for evaluating language-oriented planners. ET-Plan-Bench (Zhang et al., 2025b) utilizes VirtualHome for text-based tasks with spatial and temporal constraints. EAI (Li et al., 2024b) formalizes task planning into four stages and separately analyzes the performance of each. LEGENT (Cheng et al., 2024) and LangR (Szot et al., 2024) build task generation pipelines for communicable robots and mobile manipulation. EmbodiedBench (Yang et al., 2025b) refines the tasks in LoTa-Bench and LangR, and combines them with low-level navigation and manipulation subsets. PARTNR (Chang et al., 2025) studies planning and reasoning in human-robot collaborative household tasks, with a large collection of tasks involving spatial, temporal, and heterogeneous constraints. Other efforts explore embodied agents in games (Fan et al., 2022; Yang et al., 2024a; Zhong et al., 2025), and safety-centric evaluations (Yin et al., 2024; Sadhu et al., 2025; Lu et al., 2025; Gao et al., 2025b). A comparison among some typical previous benchmarks is presented in Table 1.

3 The UniETP Framework

3.1 Problem Formulation

The problem of Embodied Task Planning (ETP) can be formulated as a finite-horizon partially observable Markov decision process (POMDP) implemented by an underlying simulator, $\mathcal{E} = \langle \mathcal{S}, \mathcal{A}, \mathcal{O}, T, O, R, H \rangle$, where $\mathcal{S}$, $\mathcal{A}$, $\mathcal{O}$ are the spaces of states, actions, and observations, respectively. $T : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ denotes the environment transition dynamics. $O : \mathcal{S} \to \mathcal{O}$ is a mapping from states to observations. R is a reward function. H is the maximum number of action steps. Each task instance is defined by $\tau = \langle I, s_0, L, \mathcal{E} \rangle$, where $s_0 \in \mathcal{S}$ is the initial state, L denotes the formalized task goal, and I is an instruction describing the goal in natural language. The aim of ETP is to build a policy $\pi(a_t \mid h_t, I)$, typically implemented with a VLM that takes the instruction and interaction history $h_t = (o_0, a_1, o_1, ..., o_{t-1})$ as input, and generates actions in the form of text. The

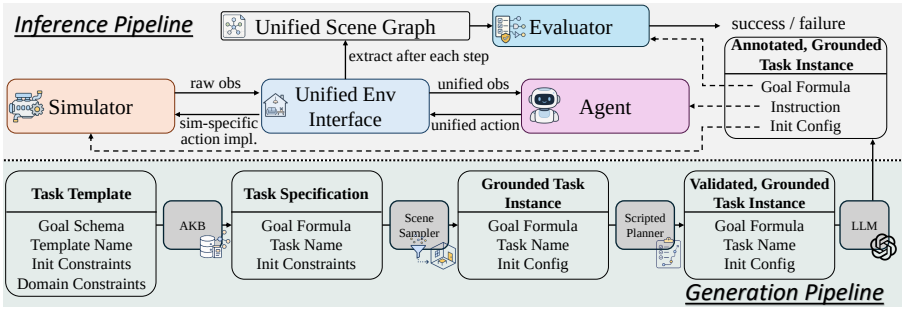


Figure 2: Upper: the ETP inference pipeline in UniETP's environment layer. Lower: the task generation process in UniETP's task layer.

agent receives a binary outcome reward at the end of the episode, indicating whether L has been achieved.

As discussed in Section 1, for a set of simulators $\{\mathcal{E}^i\}_{i=1}^{n_E}$, each of them may have its own $\langle \mathcal{S}^i, \mathcal{A}^i, \mathcal{O}^i, T^i, O^i \rangle$, making the agent π entangled with simulator-specific designs. UniETP addresses this problem by introducing a unified environment layer with a shared observation and action space $\langle \mathcal{O}^{\text{uni}}, \mathcal{A}^{\text{uni}} \rangle$. The agent can thus receive simulator-agnostic observations $o_t^{\text{uni}} \in \mathcal{O}^{\text{uni}}$ and produce simulator-agnostic actions $a_t^{\text{uni}} \in \mathcal{A}^{\text{uni}}$, which are pre-processed from/post-processed to the simulator-compatible states/actions by the projectors in UniETP's environment layer, $\phi^i(s_t^i) = o_t^{\text{uni}}, \psi^i(a_t^{\text{uni}}) = a_t^i$. The following subsections will explain the design of the unified environment layer and the task layer on top of it, and Figure 2 gives a visualization of the whole pipeline.

3.2 The Environment Layer

3.2.1 Unified Interface

Observation. UniETP defines a unified observation space that consists of three components: local view, global information, and feedback on the previous action, $o_t^{\text{uni}} = (l_t, g, f_t)$. The **local view** l_t primarily consists of an egocentric RGB image captured by the agent's head-mounted camera. In addition, UniETP optionally provides a list of bounding boxes (bboxes) for objects visible in the current view, which allows us to disentangle the agent's basic visual perception and grounding skills from its task-oriented reasoning and planning capabilities.

The definition of **global information** varies across prior benchmarks. Some (Shridhar et al., 2021) provide the agent with a list of objects in the scene at the beginning of an episode, while some (Yang et al., 2025b) implicitly expose such information through an action space that includes "`go to <sth>`" for possible objects. We note that the content of global information has a significant impact on task difficulty in complex, cluttered environments. Access to more scene-level prior knowledge can enable the agent to complete tasks in a direct manner, while less information requires more deliberate exploration and environment understanding. To provide a clear and standardized protocol, UniETP organizes global scene information into three levels: rooms, static objects (*e.g.*, furniture), and movable objects. By allowing g to include some or all of these levels, UniETP controls the amount of scene priors available to the agent, and facilitates analysis of its exploration behavior.

The **feedback** f_t represents the result of the previous action a_{t-1}. It is a binary signal indicating whether a_{t-1} is successfully performed. For some commonly identifiable failure types (*e.g.*, invalid action argument, object out of reach), we also include a short explanation in the feedback message to help the agent identify why it fails.

Action. We represent each action instance with an action type plus zero to more objects/rooms as arguments. This parameterized formulation avoids enumerating all possible action instances (Szot et al., 2024; Yang et al., 2025b), mitigating the combinatorial growth of the action space for scenes with a large number of objects. To construct $\mathcal{A}^{\text{uni}}$, we examine the object states and action interfaces supported by different simulators, align actions with equivalent semantics, and merge them into a shared set of 30+ action categories. They can be broadly divided into navigation actions and manipulation actions. For **navigation**, UniETP supports three levels of action granularity. *Movement* consists of basic locomotion actions, such as moving and rotating. *Reaching* consists of turning and getting close to an object currently visible in the view. *Teleportation* consists of turning and getting close to any specified object or room in the scene. These different levels of navigation abstraction affect the difficulty of object search and scene exploration.

Table 2: A comparison of the observation and action format among different evaluation modes.

Eval.	Local Obs		Global Obs			Nav Action			Manip Action	
Mode	Img.	Bbox	Rooms	Stat. Obj.	Mova. Obj.	Teleport	Reach	Move	Atom.	Comp.
M1	✓	✓	✓	✓	✓	by ID	by ID	✓	by ID	by ID
M2	✓	✓	✓	✓	✗	by ID	by ID	✓	by ID	✗
M3	✓	✗	✓	✗	✗	by ID (room only)	by bbox	✓	by bbox	✗

For **manipulation**, UniETP distinguishes between *atomic* and *composite* actions. The former are non-decomposable primitive operations, whereas the latter can be realized by multiple atomic steps. For example, the action `heat with(potato, microwave)` can be decomposed into opening the microwave, placing the potato inside, closing and turning on the microwave, and so on. Introducing composite actions allows agents to bypass planning over certain common sub-procedures, which is another mechanism for controlling task difficulty. Finally, we add `done` to the action space, asking the agent to determine whether the task is completed instead of waiting for an oracle completion signal. The episode will also be terminated when the number of steps reaches H or when the agent has tried H' failed actions.

Scene graph and evaluation logic. Apart from the unified observations and actions, UniETP defines a unified scene graph (USG) as a simulator-agnostic environment representation. In the graph, each node represents an object or a room, and stores its ID, category, properties, current states, and spatial location. The agent is represented as a special node in the graph. Each edge encodes a spatial relation between two nodes. The USG is constructed from the privileged information provided by the simulator and is updated after each executed action. It does not interact with the agent, but it serves as the basis for task evaluation and data generation.

For evaluation, we define a logic system over the USG traces. The logic combines first-order predicates over individual USGs (unary predicates checking whether a node has a certain state, and binary ones checking whether a certain edge exists between two nodes), Boolean connectives (and, or, not), first-order quantifiers (for all, exists), temporal operators over USG sequences (at end, until, before), and counting aggregates with numeric comparisons. Each task instance is associated with a goal formula L built by these components.

3.2.2 Evaluation Protocol

Based on the standardized multi-granularity observation and action spaces, UniETP defines three evaluation modes with varying difficulty, denoted as M1, M2, and M3, as illustrated in Table 2. From the observation perspective, the global information g^{M1} contains a list of all rooms and object instances in the scene. g^{M2} contains rooms and static objects, while g^{M3} has only rooms. The local view l_t^{M1} and l_t^{M2} include a list of bboxes for visible objects, where each box is associated with an object ID and the pixel coordinates in the observation image. In contrast, l_t^{M3} has only the image.

From the action perspective, M1 allows the use of all action categories, including teleportation/reaching/movement-based navigation, and atomic/composite manipulation. The action arguments are specified by object/room IDs. M2 removes composite manipulation actions, and restricts the target of teleportation to rooms and static objects, in accordance with the information available in g^{M2}. Actions in M2 are still parameterized by object/room IDs. M3 further restricts teleportation to rooms only, consistent with g^{M3}. Moreover, except for teleportation-style navigation, all object-targeted actions in M3 must specify their argument using bboxes. We compare the model-generated bbox with the visible objects by computing the intersection-over-union (IoU), and execute the action on the best-matching object. Therefore, M3 advances from choosing the target objects from the list of bboxes to actively locating them.

From M1 to M3, the amount of scene prior gradually decreases, the set of accessible actions becomes more restricted, and the burden of perception, localization, and exploration increases. Consequently, both the overall difficulty and the expected number of action steps increase. Intuitively, M3 simulates planning in a largely unfamiliar environment: the agent only has access to the basic room layout, while object placements and spatial relations are initially unknown. It must actively explore the environment, perceive objects, and accumulate local observations to build a sufficient understanding of the scene for task completion. M2 represents an intermediate setting in which the agent has partial prior knowledge of the environment, for example through a pre-built scene graph or semantic map that records the locations of major, static entities. This setting is well aligned with household applications, where furniture is typically stationary and can be memorized, while smaller movable objects may be relocated by humans and thus require search. M1 further assumes an ideal setting

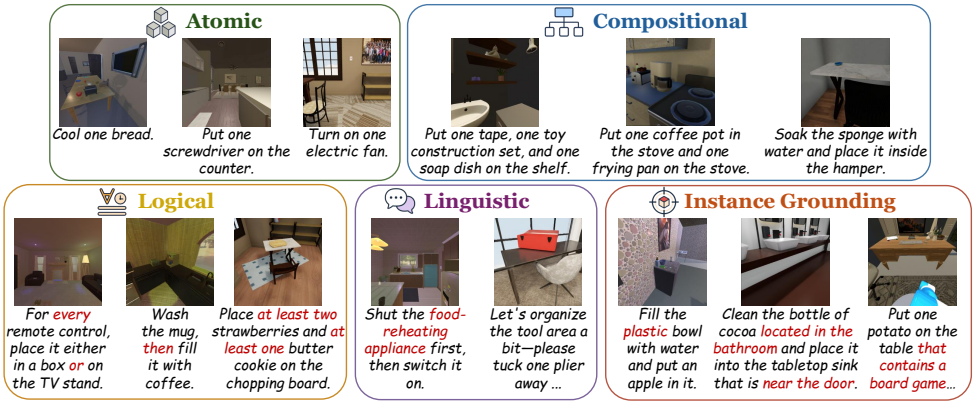


Figure 3: Example tasks on different splits.

that the locations of all objects are known and additionally provides composite actions as high-level skills, allowing the agent to focus more on reasoning over user intent and task logic.

3.3 The Task Layer

3.3.1 Task Generation Pipeline

The task layer of UniETP provides a pipeline for generating task datasets. It starts from a manually designed set of task templates. Each template is a parameterized task specification whose target object categories have not yet been instantiated, *e.g.*, `place two <A> and one <B> inside <C>`, `wash <A> then heat it`. Each template is also paired with a parameterized goal formula, some domain constraints specifying which categories are valid substitutions for each placeholder, and some initial state constraints specifying the conditions that the initial scene must satisfy.

To instantiate the placeholders with appropriate object categories, we construct a commonsense affordance knowledge base (AKB) for each simulator, which is built from their documentation and metadata, supplemented by LLM queries when the information is incomplete. Each AKB lists all the object categories supported by the simulator, the states and functionalities associated with each category, and the feasible spatial relations between any two categories. By combining AKB with the domain constraints defined in each task template, we can fill in the placeholders and convert the parameterized task template into a concrete task specification, including the goal formula L.

To ground each task specification in an initial scene, we first select a simulator $\mathcal{E}^i$ and randomly initialize a scene. We then enforce the initial state constraints by adding missing task-relevant objects and setting their states and locations accordingly. Next, we employ a scripted expert policy π^* to attempt task planning and retain only the tasks that can be successfully completed, thereby ensuring task feasibility. Finally, we use an LLM to generate language instructions I based on the task template name, the goal formula, and the object information in the instantiated scene. This process yields the final annotated, grounded task instance, $\tau = \langle I, s_0^i, L, \mathcal{E}^i \rangle$.

3.3.2 Evaluation Dimensions

When constructing the benchmark tasks, we design several capability-oriented evaluation splits to examine different aspects of ETP. (1) **Atomic.** It contains the most basic task types with a single predicate as goal, evaluating the understanding of basic object states and spatial relations. (2) **Compositional.** It consists of the conjunctions of multiple atomic tasks, evaluating the ability to handle compositional requirements and perform long-horizon planning. (3) **Logical.** It contains tasks with explicit logical structures. Their goal formulas may involve universal quantification, temporal ordering, or counting aggregates and numerical comparisons. (4) **Linguistic.** It contains complex language instructions which may refer to targets through their functions, aliases, or contextual descriptions, requiring the agent to infer the intended object category. They may also introduce background information, distractors, or syntactic structures, requiring the agent to extract the actual objective from linguistically rich descriptions. (5) **Instance Grounding.** It imposes instance-level constraints on task-relevant objects, which require the agent to select specific object instances based on visual attributes or spatial relations (*e.g.*, place the *dirty* cup on the *wooden* table, place the cup *in the coffee machine* onto the table *next to the chair*). Therefore, the agent must perform fine-grained visual recognition and localization, rather than relying only on object categories.

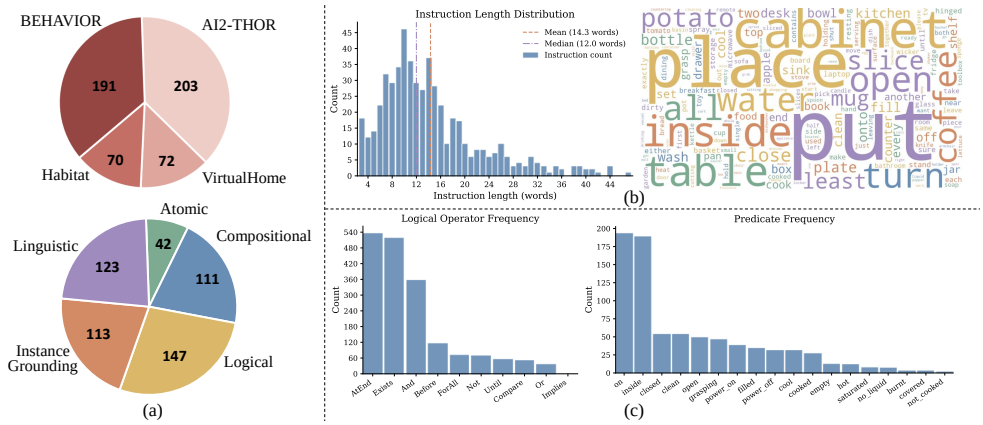


Figure 4: Key statistics of the UniETP benchmark. (a) Numbers of tasks across different simulators and splits. (b) Length distribution and word cloud of the instructions. The complex instructions in the Linguistic split account for the long tail of the distribution. (c) Frequencies of different logical operators and predicates, demonstrating the diversity of goal specifications and object states.

3.3.3 Benchmark Statistics

The current version of UniETP combines AI2-THOR (Kolve et al., 2017), VirtualHome (Puig et al., 2018), Habitat (Puig et al., 2024), and BEHAVIOR (Li et al., 2024a) into a unified interface. Fully leveraging the assets provided by the simulators, the benchmark covers 170 scene layouts. Each scene contains 7 rooms and 210 objects on average, and the objects come from the 125/181/389/1829 categories provided by each simulator. In terms of the action space, UniETP contains a total of 38 action types, taking either object IDs or bboxes as arguments.

We define 138 task templates and generate 536 task instances for the benchmark, covering a wide range of object states, spatial relations, and interaction patterns. The feasibility is validated by the scripted expert policy and the instructions are manually checked to ensure alignment with the task goal. The distributions of tasks, instructions, and goal formulas are presented in Figure 4. We additionally build a larger training set of 16k tasks to support model fine-tuning.

4 Experiments

4.1 Setup

We benchmark a wide range of frontier models on UniETP, including leading proprietary models (GPT, Gemini, Claude), popular open models (Bai et al., 2025; Wang et al., 2025b; Google DeepMind, 2026b), and embodied foundation models (Google DeepMind, 2026a; Tan et al., 2026; Zou et al., 2025; Azzolini et al., 2025; Yang et al., 2025a; Dang et al., 2026; Gong et al., 2026; Hao et al., 2025). We employ a ReAct-style (Yao et al., 2023) pipeline to evaluate each model in closed-loop interactions. In each call, we include the description of the action space, the instruction, the global information, current local observation (image and bbox list for M1/M2, image only for M3), and the action and feedback history in the input. The VLM is prompted to generate optional thinking and one or more actions in JSON format. The maximum steps H and maximum failed steps H' are set to 50/10 for M1 and M2, and 150/30 for M3. The task success rate (SR), evaluated with the goal formula, is used as the main metric.

4.2 Benchmark Results

The evaluation results under M1 and M2 are shown in Table 3. Overall, closed-source models achieve the strongest performance, with Gemini-ER-1.6 obtaining the best average score in M1 and Gemini-3.1-pro in M2. Among open-source VLMs, increased model size leads to consistently better performance, and the gap between medium sized models (*e.g.*, Qwen3.6-27B) and closed-source models is relatively modest. This saturation suggests that scaling model size may not be the only way to enhance planning capability. On the other hand, existing open-source embodied foundation models perform worse than general-purpose VLMs of comparable or even smaller sizes. The result implies that their embodied-specific fine-tuning may not transfer well to our closed-loop evaluation setting, which requires multi-turn interaction, complex instruction following, and long-horizon planning.

Across task splits, Atomic tasks are the easiest as expected, and performance drops clearly on the Compositional split, showing that compound goals and extended horizon substantially increase the

Table 3: Performance comparison on the UniETP benchmark under evaluation modes M1 and M2, including results across the atomic (At.), compositional (Co.), logical (Lo.), instance grounding (In.), linguistic (Li.) splits and the average (Avg.) on the whole benchmark.

	M1						M2					
	At.	Co.	Lo.	In.	Li.	Avg.	At.	Co.	Lo.	In.	Li.	Avg.
GPT-5.5	90	82	76	48	74	72	45	21	33	20	27	27
GPT-5.6-sol	**98**	81	78	44	76	73	48	22	34	25	32	30
Gemini-ER-1.6	93	**86**	**87**	**50**	73	**76**	45	23	33	30	24	29
Gemini-3.1-pro	**98**	79	80	46	74	73	**55**	**31**	33	**35**	**34**	**35**
Claude-sonnet-4.6	90	81	78	45	76	72	45	26	27	25	26	27
Claude-opus-4.7	**98**	80	80	**50**	**80**	75	52	**31**	**36**	29	30	33
Qwen3.5-4B	67	46	41	24	43	41	29	5	10	11	13	11
Qwen3.5-9B	76	68	62	37	60	59	33	10	14	12	15	14
Qwen3.6-27B	86	72	68	42	72	66	45	17	19	19	22	21
InternVL3.5-8B	43	21	19	13	21	21	19	3	9	4	11	8
InternVL3.5-38B	62	36	38	15	37	34	21	7	8	9	11	10
Gemma4-E4B-it	55	32	23	19	24	27	21	5	10	9	10	10
Gemma4-31B-it	86	67	72	40	68	64	40	13	22	19	23	21
RoboBrain2.5-8B	36	19	22	12	24	21	24	5	11	4	8	9
EmbodiedBrain-7B	36	10	17	11	12	15	19	3	7	4	6	6
Vlaser-8B	45	9	7	6	11	11	19	3	3	4	4	5
Cosmos-Reason2-8B	36	17	15	9	20	17	21	5	7	5	6	7
RynnBrain-8B	43	17	16	12	18	18	17	3	5	5	5	5
ACE-Brain-0-8B	36	13	12	11	14	14	31	2	7	5	6	7
MiMo-Embodied-7B	57	15	14	10	14	17	24	3	9	7	8	8

Table 4: Performance comparison under evaluation mode M3.

	At.
GPT-5.5	26
Gemini-3.1-pro	**40**
claude-opus-4.7	26

Table 5: Performance comparison on different simulators

	M1					M2				
	A	V	H	B	Avg.	A	V	H	B	Avg.
Gemini-3.1-pro	64	82	70	79	73	46	64	27	15	35
Qwen3.6-27B	56	69	69	73	66	28	51	11	7	21
RoboBrain2.5-8B	15	40	19	20	21	7	28	1	5	9

difficulty of planning. The Logical and Linguistic splits do not exhibit a consistent degradation relative to the Compositional split, meaning that understanding logical relations or complex linguistic descriptions may not be the dominant bottleneck for zero-shot VLM-based agents. Instance Grounding is the most challenging split for most models, with even the strongest models reaching only 50% in M1. We notice that current agents struggle with fine-grained visual discrimination, and they primarily rely on categories when choosing target objects.

Comparing mode M1 and M2, we observe a substantial and consistent performance drop for all models. This confirms that M2 is considerably more challenging: the agent has to go through an explicit exploration process (teleportation to movable objects is banned), and plan with more primitive actions (composite actions are banned). For M3, we evaluate three closed-source models on the Atomic split only, as shown in Table 4. We can observe an additional increase in difficulty and that even these basic tasks are challenging for the agents, since they must make more detailed navigation decisions and ground target objects by themselves. These results suggest that there remains substantial room for improving VLM-based embodied agents, through, for example, better scene memory, spatial modeling, object grounding, and reflection mechanisms.

4.3 Fine-Grained Analysis on Performance

We further analyze model performance across different simulator backends, as shown in Table 5. VirtualHome is the easiest domain in our benchmark, mainly because it contains fewer interactable

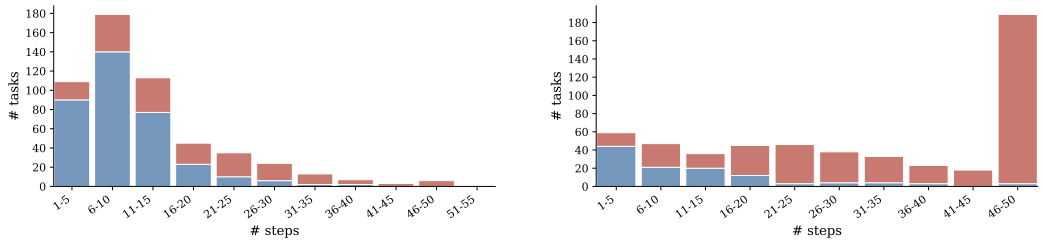


Figure 5: The distribution of the number of steps under mode M1 (left) and M2 (right). The successful/failed episodes are represented in blue/orange.

Table 6: Performance of the SFT models. The regions colored in gray represent splits that require domain generalization.

	A		V		H		B		Avg.
	Seen	Unseen	Seen	Unseen	Seen	Unseen	Seen	Unseen	
Qwen3.5-4B	23	31	58	53	23	41	55	58	41
SFT (w.o. AI2-THOR)	44	38	68	84	81	82	73	69	62
SFT (w.o. unseen tasks)	55	49	68	63	74	82	69	64	63
SFT (full dataset)	59	54	68	84	79	94	74	72	**68**

Table 7: Generalization performance on EmbodiedBench (Yang et al., 2025b)

	EB-ALFRED							EB-Habitat						
Qwen3.5-4B	Base	CS	CI	VA	SR	LH	Avg.	Base	CS	CI	VA	SR	LH	Avg.
raw	28	30	32	34	28	18	28	**66**	16	28	30	32	12	31
SFT on UniETP (same sim.)	46	34	48	44	32	24	38	58	**44**	46	**32**	36	22	40
SFT on UniETP (full)	**48**	**44**	**52**	**48**	**36**	**48**	**46**	**66**	**44**	**52**	28	**38**	**24**	**42**

objects and exposes a smaller set of executable action categories. Under M1, AI2-THOR presents the highest difficulty, as its tasks involve richer object states and more diverse tool affordances. However, under M2, performance on Habitat and BEHAVIOR drops substantially. This is largely due to their larger scene scales: agents must search for the small, movable target objects by traversing multiple rooms, making object discovery and exploration considerably more challenging. Figure 5 shows the distribution of episode lengths and the success ratio within each length range (using Qwen3.6-27B). The performance decreases steadily as the number of steps increases, indicating that models become less reliable when tasks require longer horizons and larger interaction context. In addition, due to the differences in the granularity of action space, M2 incurs a larger step cost than M1. Also, a lot of M2 episodes reach the maximum step limit, typically when the agent fails to find all the target objects.

4.4 Analysis on Training and Generalization

In this subsection, we present a preliminary attempt to develop generalizable embodied agents using UniETP. Based on the training set constructed as in Section 3.3, we perform supervised fine-tuning (SFT) on Qwen3.5-4B. As shown in Table 6, the fine-tuned model achieves a substantial performance improvement and even outperforms models with more parameters (Qwen3.6-27B, Gemma4-31B-it), demonstrating the reliability of our data generation and supervision construction pipeline.

Furthermore, we exclude all episodes from a specific simulator (AI2-THOR), or from a subset of task templates (about 1/3 of the full template set), in the training set. At test time, the resulting models still show clear improvements over the original pretrained model in the unseen simulator environments and the unseen task types. These results suggest that bridging simulator boundaries and unifying the observation and action interface can contribute meaningfully to the training of agents with stronger generalization capability.

We also investigate whether UniETP can help agents acquire capabilities that transfer to other benchmarks. As shown in Table 7, we perform SFT with Qwen3.5-4B on UniETP's training tasks and evaluate the resulting model on EmbodiedBench (Yang et al., 2025b). The results show that training on UniETP leads to clear performance improvements, demonstrating the generality and transferability of the task design in UniETP. Furthermore, compared with training only on data from the corresponding simulator (using the AI2-THOR subset for evaluation on EB-ALFRED and the Habitat subset for evaluation on EB-Habitat), training on the full UniETP dataset yields further improvements on the majority of evaluation splits. This finding provides additional evidence that UniETP's unified environment interface facilitates cross-simulator generalization.

5 Conclusion

In this work, we propose UniETP, a unified benchmark for evaluating embodied planning agents across heterogeneous simulation environments. UniETP transforms four commonly-used simulators into a standardized observation and action space, and defines an evaluation protocol with adjustable difficulty. Furthermore, it introduces an automated task generation pipeline to construct a benchmark covering diverse capability dimensions, together with a large-scale training set. Comprehensive experiments across a wide range of models reveal insights into the performance and limitations of VLM-based agents in embodied scenarios.

AI USE STATEMENT

In this work, we used generative AI tools to generate task instructions, build affordance knowledge bases for some simulators, and assist with translation. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, design or provide feedback on research methodology or experiments, clean or reformat dataset, support qualitative and thematic data analysis, interpret results. The tasks of formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses are not applicable to this work. Additionally, we used generative AI tools to assist in coding, polish the research paper, search for information, summarize or analyse existing literature, identify relevant literature, and edit the draft to improve readability. We have reviewed all AI-assisted work. The LLM-generated task instructions on the benchmark were manually verified for alignment with the task goals. The LLM-generated codes and paper paragraphs were also manually reviewed for correctness. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

To the best of our knowledge, this work does not involve human subjects, potentially harmful insights, fairness, legal issues, privacy, or other ethical concerns. We introduce a new benchmark built on simulators, scenes, and object assets from existing open-source projects. On top of these resources, we develop a unified environment interface and define new interactive tasks. These tasks focus on everyday household scenarios and do not involve safety-critical risks.

REPRODUCIBILITY STATEMENT

We provide the implementation code for the UniETP benchmark in the supplementary material, together with instructions for running it. We also include a small subset of the benchmark tasks to illustrate the data format, and the full benchmark and training set will be released upon acceptance. Section A in the appendix describes how we wrap and unify the different simulators. Section C details the dataset generation pipeline. Sections D and E present the key configurations used for evaluation and training.

REFERENCES

Abbas Abdolmaleki, Saminda Abeyruwan, Joshua Ainslie, Jean-Baptiste Alayrac, Montserrat Gonzalez Arenas, Ashwin Balakrishna, Nathan Batchelor, Alex Bewley, Jeff Bingham, Michael Bloesch, et al. Gemini robotics 1.5: Pushing the frontier of generalist robots with advanced embodied reasoning, thinking, and motion transfer. *arXiv preprint arXiv:2510.03342*, 2025.

Alisson Azzolini, Junjie Bai, Hannah Brandon, Jiaxin Cao, Prithvijit Chattopadhyay, Huayu Chen, Jinju Chu, Yin Cui, Jenna Diamond, Yifan Ding, et al. Cosmos-reason1: From physical common sense to embodied reasoning. *arXiv preprint arXiv:2503.15558*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Muzhen Cai, Xiubo Chen, Yining An, Jiaxin Zhang, Xuesong Wang, Wang Xu, Weinan Zhang, and Ting Liu. Cookbench: A long-horizon embodied planning benchmark for complex cooking scenarios. *arXiv preprint arXiv:2508.03232*, 2025.

Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, et al. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks. In *International Conference on Learning Representations*, volume 2025, pp. 65205–65268, 2025.

Lirong Che, Shuo Wen, Shan Huang, Chuang Wang, Yuzhe Yang, Gregory Dudek, Xueqian Wang, and Jian Su. Mansion: Multi-floor language-to-3d scene generation for long-horizon tasks. *arXiv preprint arXiv:2603.11554*, 2026.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024.

Junting Chen, Checheng Yu, Xunzhe Zhou, Tianqi Xu, Yao Mu, Mengkang Hu, Wenqi Shao, Yikai Wang, Guohao Li, and Lin Shao. Emos: Embodiment-aware heterogeneous multi-robot operating system with llm agents. In *International Conference on Learning Representations*, volume 2025, pp. 56670–56690, 2025.

Yang Chen, Ye-Xin Xie, Lirong Che, Danyang Peng, Yuzhe Yang, Peiwen Lin, Xu Cao, Chuang Wang, Lei Yuan, Jian Su, et al. Cocobench: A cooperative coordination benchmark for embodied multi-agent task planning. *arXiv preprint arXiv:2608.28266*, 2026a.

Ying Chen, Lihuang Fang, Rui Jiang, Mingxu Wang, Zhifeng Gu, Lei Yi, and Jie Chen. Done, but not sure: Disentangling world completion from self-termination in embodied agents. *arXiv preprint arXiv:2605.08747*, 2026b.

Qinchuan Cheng, Zhantao Gong, Pengzhan Sun, Angela Yao, Xulei Yang, and Shijie Li. Ego2world: Compiling egocentric cooking videos into executable worlds for belief-state planning. *arXiv preprint arXiv:2605.13335*, 2026a.

Zhili Cheng, Zhitong Wang, Jinyi Hu, Shengding Hu, An Liu, Yuge Tu, Pengkai Li, Lei Shi, Zhiyuan Liu, and Maosong Sun. Legent: Open platform for embodied agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 335–345, 2024.

Zhili Cheng, Ran Li, Jinyi Hu, Yuge Tu, Shiqi Dai, Shengding Hu, Yang Shi, Lei Shi, and Maosong Sun. Embodiedeval: Evaluate multimodal llms as embodied agents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11420–11432, 2026b.

Jae-Woo Choi, Youngwoo Yoon, Hyobin Ong, Jaehong Kim, and Minsu Jang. Lota-bench: Benchmarking language-oriented task planners for embodied agents. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=ADSxCpCu9s`.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. Textworld: A learning environment for text-based games. In *Workshop on Computer Games*, pp. 41–75. Springer, 2018.

Liu Dai, Haina Wang, Weikang Wan, and Hao Su. Manitaskgen: A comprehensive task generator for benchmarking and improving vision-language agents on embodied decision-making. *arXiv preprint arXiv:2505.20726*, 2025.

Ronghao Dang, Jiayan Guo, Bohan Hou, Sicong Leng, Kehan Li, Xin Li, Jiangpin Liu, Yunxuan Mao, Zhikai Wang, Yuqian Yuan, et al. Rynnbrain: Open embodied foundation models. *arXiv preprint arXiv:2602.14979*, 2026.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. Procthor: Large-scale embodied ai using procedural generation. *Advances in Neural Information Processing Systems*, 35:5982–5994, 2022.

Linxi Fan, Guanzhi Wang, Yunfan Jiang, Ajay Mandlekar, Yuncong Yang, Haoyi Zhu, Andrew Tang, De-An Huang, Yuke Zhu, and Anima Anandkumar. Minedojo: Building open-ended embodied agents with internet-scale knowledge. *Advances in Neural Information Processing Systems*, 35: 18343–18362, 2022.

C Gan, J Schwartz, S Alter, M Schrimpf, J Traer, J De Freitas, J Kubilius, A Bhandwaldar, N Haber, M Sano, et al. Threedworld: A platform for interactive multi-modal physical simulation. *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Chuang Gan, Siyuan Zhou, Jeremy Schwartz, Seth Alter, Abhishek Bhandwaldar, Dan Gutfreund, Daniel LK Yamins, James J DiCarlo, Josh McDermott, Antonio Torralba, 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.

Chongkai Gao, Zixuan Liu, Zhenghao Chi, Junshan Huang, Xin Fei, Yiwen Hou, Yuxuan Zhang, Yudi Lin, Zhirui Fang, and Lin Shao. Vla-os: Structuring and dissecting planning representations and paradigms in vision-language-action models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 136705–136736. Curran Associates, Inc., 2025a. doi: 10.52202/085713-4561. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/c7af751b5e0a407c62ac023e3cb381f9-Paper-Conference.pdf.

Hongcheng Gao, Hailong Qu, Jingyi Tang, Jiahao Wang, Zihao Huang, Hengkang Qiao, Shihong Huang, Junming Yang, Yi Li, Hongyixuan Yuan, et al. Spatialworld: Benchmarking interactive spatial reasoning of multimodal agents in real-world tasks. *arXiv preprint arXiv:2606.09669*, 2026.

Siyuan Gao, Jiashu Yao, Haoyu Wen, Yuhang Guo, Zeming Liu, and Heyan Huang. Homesafebench: A benchmark for embodied vision-language models in free-exploration home safety inspection. *arXiv preprint arXiv:2509.23690*, 2025b.

Xiaofeng Gao, Ran Gong, Tianmin Shu, Xu Xie, Shu Wang, and Song-Chun Zhu. Vrkitchen: an interactive 3d virtual environment for task-oriented learning. *arXiv preprint arXiv:1903.05757*, 2019.

Xiaofeng Gao, Qiaozi Gao, Ran Gong, Kaixiang Lin, Govind Thattai, and Gaurav S Sukhatme. Dialfred: Dialogue-enabled agents for embodied instruction following. *IEEE Robotics and Automation Letters*, 7(4):10049–10056, 2022.

Ziyang Gong, Zehang Luo, Anke Tang, Zhe Liu, Shi Fu, Zhi Hou, Ganlin Yang, Weiyun Wang, Xiaofeng Wang, Jianbo Liu, et al. Ace-brain-0: Spatial intelligence as a shared scaffold for universal embodiments. *arXiv preprint arXiv:2603.03198*, 2026.

Google DeepMind. Gemini Robotics-ER 1.6 Model Card. https://deepmind.google/models/model-cards/gemini-robotics-er-1-6/, April 2026a.

Google DeepMind. Gemma 4 Model Card. https://ai.google.dev/gemma/docs/core/model_card_4, April 2026b.

Xiaoshuai Hao, Lei Zhou, Zhijian Huang, Zhiwen Hou, Yingbo Tang, Lingfeng Zhang, Guang Li, Zheng Lu, Shuhuai Ren, Xianhui Meng, et al. Mimo-embodied: X-embodied foundation model technical report. *arXiv preprint arXiv:2511.16518*, 2025.

Yining Hong, Huang Huang, Manling Li, Li Fei-Fei, Leonidas Guibas, Jiajun Wu, and Yejin Choi. Learning from trials and errors: Reflective test-time planning for embodied llms. *arXiv preprint arXiv:2602.21198*, 2026.

Jiani Huang, Amish Sethi, Matthew Kuo, Mayank Keoliya, Neelay Velingker, JungHo Jung, Ser-Nam Lim, Ziyang Li, and Mayur Naik. ESCA: Contextualizing embodied agents via scene-graph generation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pp. 9118–9147. PMLR, 2022.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning*, pp. 1769–1782. PMLR, 2023.

Physical Intelligence, Kevin Black, Noah Brown, James Darpinian, Karan Dhabalia, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Manuel Y. Galliker, Dibya Ghosh, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Devin LeBlanc, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Allen Z. Ren, Lucy Xiaoyang Shi, Laura Smith, Jost Tobias Springenberg, Kyle Stachowicz, James Tanner, Quan Vuong, Homer Walke, Anna Walling, Haohuan Wang, Lili Yu, and Ury Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization, 2025.

Chenxi Jiang, Chuhao Zhou, and Jianfei Yang. Rei-bench: Can embodied agents understand vague human instructions in task planning? *arXiv preprint arXiv:2505.10872*, 2025.

Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X Chang, and Manolis Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16384–16393, 2024.

Byeonghwi Kim, Minhyuk Seo, and Jonghyun Choi. Online continual learning for interactive instruction following agents. In *International Conference on Learning Representations*, volume 2024, pp. 19632–19651, 2024a.

Taewoong Kim, Cheolhong Min, Byeonghwi Kim, Jinyeon Kim, Wonje Jeung, and Jonghyun Choi. Realfred: An embodied instruction following benchmark in photo-realistic environments. In *European Conference on Computer Vision*, pp. 346–364. Springer, 2024b.

Yejin Kim, Wilbert Pumacay, Omar Rayyan, Max Argus, Winson Han, Eli VanderBilt, Jordi Salvador, Abhay Deshpande, Rose Hendrix, Snehal Jauhri, et al. Molmospaces: A large-scale open ecosystem for robot navigation and manipulation. *arXiv preprint arXiv:2602.11337*, 2026.

Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017.

Zixing Lei, Sheng Yin, Yichen Xiong, Yuanzhuo Ding, Wenhao Huang, Yuxi Wei, Qingyao Xu, Yiming Li, Weixin Li, Yunhong Wang, et al. Embomatrix: A scalable training-ground for embodied decision-making. *arXiv preprint arXiv:2510.12072*, 2025.

Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Elliott Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, et al. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. In *Conference on Robot Learning*, pp. 455–465. PMLR, 2022a.

Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Wensi Ai, Benjamin Martinez, et al. Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation. *arXiv preprint arXiv:2403.09227*, 2024a.

Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Erran Li Li, Ruohan Zhang, et al. Embodied agent interface: Benchmarking llms for embodied decision making. *Advances in Neural Information Processing Systems*, 37:100428–100534, 2024b.

Rui Li, Zixuan Hu, Wenxi Qu, Jinouwen Zhang, Zhenfei Yin, Sha Zhang, Xuantuo Huang, Hanqing Wang, Tai Wang, Jiangmiao Pang, et al. Labutopia: High-fidelity simulation and hierarchical benchmark for scientific embodied agents. *arXiv preprint arXiv:2505.22634*, 2025a.

Shuang Li, Xavier Puig, Chris Paxton, Yilun Du, Clinton Wang, Linxi Fan, Tao Chen, De-An Huang, Ekin Akyürek, Anima Anandkumar, et al. Pre-trained language models for interactive decision-making. volume 35, pp. 31199–31212, 2022b.

Xiaoxi Li, Wenxiang Jiao, Jiarui Jin, Guanting Dong, Jiajie Jin, Yinuo Wang, Hao Wang, Yutao Zhu, Ji-Rong Wen, Yuan Lu, et al. Deepagent: A general reasoning agent with scalable toolsets. *arXiv preprint arXiv:2510.21618*, 2025b.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500. IEEE, 2023.

Yuan-Hong Liao, Xavier Puig, Marko Boben, Antonio Torralba, and Sanja Fidler. Synthesizing environment-aware activities via activity sketches. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6291–6299, 2019.

Jie Liu, Pan Zhou, Yingjun Du, Ah-Hwee Tan, Cees G. M. Snoek, Jan-Jakob Sonke, and Efstratios Gavves. Capo: Cooperative plan optimization for efficient embodied multi-agent cooperation. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Kehui Liu, Zixin Tang, Dong Wang, Zhigang Wang, Xuelong Li, and Bin Zhao. Coherent: Collaboration of heterogeneous multi-robot system with large language models. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 10208–10214. IEEE, 2025b.

Xiao Liu, Tianjie Zhang, Yu Gu, Iat Long Iong, Song XiXuan, Yifan Xu, Shudan Zhang, Hanyu Lai, Jiadai Sun, Xinyue Yang, Yu Yang, Zehan Qi, Shuntian Yao, Xueqiao Sun, Siyi Cheng, Qinkai Zheng, Hao Yu, Hanchen Zhang, Wenyi Hong, Ming Ding, Lihang Pan, Xiaotao Gu, Aohan Zeng, Zhengxiao Du, Chan Hee Song, Yu Su, Yuxiao Dong, and Jie Tang. Visualagentbench: Towards large multimodal models as visual foundation agents. In *The Thirteenth International Conference on Learning Representations*, 2025c.

Zhihong Liu, Yang Li, Rengming Huang, Cewu Lu, and Panpan Cai. Any house any task: Scalable long-horizon planning for abstract human tasks. *arXiv preprint arXiv:2602.12244*, 2026.

Zhichen Lou, Kechun Xu, Zhongxiang Zhou, and Rong Xiong. Explorevlm: Closed-loop robot exploration task planning with vision-language models. *arXiv preprint arXiv:2508.11918*, 2025.

Xiaoya Lu, Zeren Chen, Xuhao Hu, Yijin Zhou, Weichen Zhang, Dongrui Liu, Lu Sheng, and Jing Shao. Is-bench: Evaluating interactive safety of vlm-driven embodied agents in daily household tasks. *arXiv preprint arXiv:2506.16402*, 2025.

Arjun Majumdar, Anurag Ajay, Xiaohan Zhang, Pranav Putta, Sriram Yenamandra, Mikael Henaff, Sneha Silwal, Paul Mcvay, Oleksandr Maksymets, Sergio Arnaud, et al. Openeqa: Embodied question answering in the era of foundation models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 16488–16498, 2024.

Mohammad Mohammadi, Daniel Honerkamp, Martin Büchner, Matteo Cassinelli, Tim Welschehold, Fabien Despinoy, Igor Gilitschenski, and Abhinav Valada. More: Mobile manipulation rearrangement through grounded language reasoning. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1805–1811. IEEE, 2025.

Anatoly Onishchenko, Alexey Kovalev, and Aleksandr Panov. Lookplangraph: Embodied instruction following method with vlm graph augmentation. In *Workshop on Reasoning and Planning for Large Language Models*, 2025.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. Teach: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 2017–2025, 2022.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8494–8502, 2018.

Xavier Puig, Tianmin Shu, Shuang Li, Zilin Wang, Yuan-Hong Liao, Joshua B Tenenbaum, Sanja Fidler, and Antonio Torralba. Watch-and-help: A challenge for social perception and human-ai collaboration. In *International Conference on Learning Representations*, 2021.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars, and robots. In *International Conference on Learning Representations*, volume 2024, pp. 15306–15336, 2024.

Ram Ramrakhya, Matthew Chang, Xavier Puig, Ruta Desai, Zsolt Kira, and Roozbeh Mottaghi. Grounding multimodal llms to embodied agents that ask for help with reinforcement learning. *arXiv preprint arXiv:2504.00907*, 2025.

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. In *Conference on Robot Learning*, pp. 23–72. PMLR, 2023.

Tanmana Sadhu, Yanan Chen, and Ali Pesaranghader. Vestabench: An embodied benchmark for safe long-horizon planning under multi-constraint and adversarial settings. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 2122–2145, 2025.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9339–9347, 2019.

SeungWon Seo, SeongRae Noh, Junhyeok Lee, SooBin Lim, Won Hee Lee, and HyeongYeop Kang. Reveca: Adaptive planning and trajectory-based validation in cooperative language agents using information relevance and relative proximity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 23295–23303, 2025.

Rutav Shah, Albert Yu, Yifeng Zhu, Yuke Zhu, and Roberto Martín-Martín. Bumble: Unifying reasoning and acting with vision-language models for building-wide mobile manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13337–13345. IEEE, 2025.

Bokui Shen, Fei Xia, Chengshu Li, Roberto Martín-Martín, Linxi Fan, Guanzhi Wang, Claudia Pérez-D'Arpino, Shyamal Buch, Sanjana Srivastava, Lyne Tchapmi, 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)*, pp. 7520–7527. IEEE, 2021.

Junhao Shi, Zhaoye Fei, Siyin Wang, Qipeng Guo, Jingjing Gong, and Xipeng Qiu. World-aware planning narratives enhance large vision-language model planner. *arXiv preprint arXiv:2506.21230*, 2025a.

Lucy Xiaoyang Shi, Brian Ichter, Michael Equi, Liyiming Ke, Karl Pertsch, Quan Vuong, James Tanner, Anna Walling, Haohuan Wang, Niccolo Fusai, et al. Hi robot: Open-ended instruction following with hierarchical vision-language-action models. *arXiv preprint arXiv:2502.19417*, 2025b.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10740–10749, 2020.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. {ALFW}orld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11523–11530. IEEE, 2023.

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2998–3009, 2023.

Yifan Song, Weimin Xiong, Xiutian Zhao, Dawei Zhu, Wenhao Wu, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. Agentbank: Towards generalized llm agents via fine-tuning on 50000+ interaction trajectories. In *EMNLP (Findings)*, 2024.

Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Elliott Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, Karen Liu, et al. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments. In *Conference on robot learning*, pp. 477–490. PMLR, 2022.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in neural information processing systems*, 34:251–266, 2021.

Andrew Szot, Max Schwarzer, Harsh Agrawal, Bogdan Mazoure, Rin Metcalf, Walter Talbott, Natalie Mackraz, R Devon Hjelm, and Alexander T Toshev. Large language models as generalizable policies for embodied tasks. In *The Twelfth International Conference on Learning Representations*, 2024.

Huajie Tan, Enshen Zhou, Zhiyu Li, Yijie Xu, Yuheng Ji, Xiansheng Chen, Cheng Chi, Pengwei Wang, Huizhu Jia, Yulong Ao, et al. Robobrain 2.5: Depth in sight, time in mind. *arXiv preprint arXiv:2601.14352*, 2026.

BAAI RoboBrain Team, Mingyu Cao, Huajie Tan, Yuheng Ji, Xiansheng Chen, Minglan Lin, Zhiyu Li, Zhou Cao, Pengwei Wang, Enshen Zhou, et al. Robobrain 2.0 technical report. *arXiv preprint arXiv:2507.02029*, 2025.

Wanxin Tian, Shijie Zhang, Kevin Zhang, Xiaowei Chi, Chun-Kai Fan, Junyu Lu, Yulin Luo, Qiang Zhou, Yiming Zhao, Ning Liu, Siyu Lin, Zhiyuan Qin, Xiaozhu Ju, Shanghang Zhang, and Jian Tang. Seea-r1: Tree-structured reinforcement fine-tuning for self-evolving embodied agents. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 78458–78499. Curran Associates, Inc., 2025. doi: 10.52202/085713-2628. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/7103cd82de95a7b30983fcf74ba499ac-Paper-Conference.pdf.

Andrea Tupini, Lars Liden, Reuben Tan, Yu Wang, and Jianfeng Gao. Asgardbench–evaluating visually grounded interactive planning under minimal feedback. *arXiv preprint arXiv:2603.15888*, 2026.

Hanqing Wang, Jiahe Chen, Wensi Huang, Qingwei Ben, Tai Wang, Boyu Mi, Tao Huang, Siheng Zhao, Yilun Chen, Sizhe Yang, et al. Grutopia: Dream general robots in a city at scale. *arXiv preprint arXiv:2407.10943*, 2024.

Ruiyang Wang, Hao-Lun Hsu, Swarajh Mehta, Jiwoo Kim, Zhihao Dou, and Miroslav Pajic. Gavel: Graph world models for verified and efficient long-horizon llm task planning. *arXiv preprint arXiv:2609.19315*, 2026.

Siyin Wang, Zhaoye Fei, Qinyuan Cheng, Shiduo Zhang, Panpan Cai, Jinlan Fu, and Xipeng Qiu. World modeling makes a better planner: Dual preference optimization for embodied task planning. *arXiv preprint arXiv:2503.10480*, 2025a.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025b.

Zhiheng Xi, Yiwen Ding, Wenxiang Chen, Boyang Hong, Honglin Guo, Junzhe Wang, Xin Guo, Dingwen Yang, Chenyang Liao, Wei He, et al. Agentgym: Evaluating and training large language model-based agents across diverse environments. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27914–27961, 2025.

Jiannan Xiang, Tianhua Tao, Yi Gu, Tianmin Shu, Zirui Wang, Zichao Yang, and Zhiting Hu. Language models meet world models: Embodied experiences enhance language models. *Advances in neural information processing systems*, 36:75392–75412, 2023.

Hang Xu, Zang Yu, Yehui Tang, Pengbo Hu, Yuhao Tang, and Hao Dong. Mcts-ep: Empowering embodied planning with online preference optimization. *arXiv preprint arXiv:2509.17116*, 2025a.

Manjie Xu, Xinyi Yang, Wei Liang, Chi Zhang, and Yixin Zhu. Learning to plan with personalized preferences. *arXiv preprint arXiv:2502.00858*, 2025b.

Peiran Xu, Jiaqi Zheng, and Yadong Mu. Roboagent: Chaining basic capabilities for embodied task planning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15276–15290, June 2026.

Fan Yang, Soumya Teotia, Shaunak A Mehta, Prajit KrisshnaKumar, Quanting Xie, Jun Liu, Yueqi Song, Wenkai Li, Atsunori Moteki, Kanji Uchino, et al. Indoorr2x: Indoor robot-to-everything coordination with llm-driven planning. *arXiv preprint arXiv:2603.20182*, 2026a.

Ganlin Yang, Tianyi Zhang, Haoran Hao, Weiyun Wang, Yibin Liu, Dehui Wang, Guanzhou Chen, Zijian Cai, Junting Chen, Weijie Su, et al. Vlaser: Vision-language-action model with synergistic embodied reasoning. *arXiv preprint arXiv:2510.11027*, 2025a.

Huaigang Yang, Ya Li, Min Ren, Bo Dai, Zhenliang Zhang, and Zhaofeng He. Saferelbench: A spatial-relation-aware benchmark for process-level safety in vlm-driven embodied agents. *arXiv preprint arXiv:2607.14543*, 2026b.

Jingkang Yang, Yuhao Dong, Shuai Liu, Bo Li, Ziyue Wang, Haoran Tan, Chencheng Jiang, Jiamu Kang, Yuanhan Zhang, Kaiyang Zhou, et al. Octopus: Embodied vision-language programmer from environmental feedback. In *European conference on computer vision*, pp. 20–38. Springer, 2024a.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, Heng Ji, Huan Zhang, and Tong Zhang. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. In *Forty-second International Conference on Machine Learning*, 2025b.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16277–16287. IEEE, 2024b.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Sriram Yenamandra, Arun Ramachandran, Karmesh Yadav, Austin S Wang, Mukul Khanna, Theophile Gervet, Tsung-Yen Yang, Vidhi Jain, Alexander Clegg, John M Turner, Zsolt Kira, Manolis Savva, Angel X Chang, Devendra Singh Chaplot, Dhruv Batra, Roozbeh Mottaghi, Yonatan Bisk, and Chris Paxton. Homerobot: Open-vocabulary mobile manipulation. In *7th Annual Conference on Robot Learning*, 2023. URL https://openreview.net/forum?id=b-cto-fetlz.

Sheng Yin, Xianghe Pang, Yuanzhuo Ding, Menglan Chen, Yutong Bi, Yichen Xiong, Wenhao Huang, Zhen Xiang, Jing Shao, and Siheng Chen. Safeagentbench: A benchmark for safe task planning of embodied llm agents. *arXiv preprint arXiv:2412.13178*, 2024.

Hao Yuan, Yuxin Wang, Lei Ji, and Zhiwei Yu. From failures to supervision: Dynamicenvplan for robust long-horizon embodied planning. *arXiv preprint arXiv:2608.00613*, 2026.

Yundaichuan Zhan, Minghe Gao, Zhongqi Yue, Wendong Bu, Wenqiao Zhang, Guoming Wang, Jisheng Dang, Juncheng Li, Siliang Tang, and Yueting Zhuang. SCOPE: Evolving symbolic world for planning in open-ended environments. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=PLJ53zWDTD.

Hongxin Zhang, Weihua Du, Jiaming Shan, Qinhong Zhou, Yilun Du, Joshua B Tenenbaum, Tianmin Shu, and Chuang Gan. Building cooperative embodied agents modularly with large language models. *arXiv preprint arXiv:2307.02485*, 2023.

Jiazhao Zhang, Anqi Li, Yunpeng Qi, Minghan Li, Jiahang Liu, Shaoan Wang, Haoran Liu, Gengze Zhou, Yuze Wu, Xingxing Li, et al. Embodied navigation foundation model. *arXiv preprint arXiv:2509.12129*, 2025a.

Lingfeng Zhang, Yuening Wang, Hongjian Gu, Atia Hamidizadeh, Zhanguang Zhang, Yuecheng Liu, Yutong Wang, David Gamaliel Arcos Bravo, Junyi Dong, Shunbo Zhou, et al. Et-plan-bench: Embodied task-level planning benchmark towards spatial-temporal cognition with foundation models. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 21566–21573. IEEE, 2025b.

Wenqi Zhang, Mengna Wang, Gangao Liu, Xu Huixin, Yiwei Jiang, Yongliang Shen, Guiyang Hou, Zhe Zheng, Hang Zhang, Xin Li, et al. Embodied-reasoner: Synergizing visual search, reasoning, and action for embodied interactive tasks. *arXiv preprint arXiv:2503.21696*, 2025c.

Baining Zhao, Ziyou Wang, Jianjie Fang, Chen Gao, Fanhang Man, Jinqiang Cui, Xin Wang, Xinlei Chen, Yong Li, and Wenwu Zhu. Embodied-r: Collaborative framework for activating embodied spatial reasoning in foundation models via reinforcement learning. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 11071–11080, 2025.

Zirui Zhao, Wee Sun Lee, and David Hsu. Large language models as commonsense knowledge for large-scale task planning. *Advances in neural information processing systems*, 36:31967–31987, 2023.

Duo Zheng, Shijia Huang, Lin Zhao, Yiwu Zhong, and Liwei Wang. Towards learning a generalist model for embodied navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13624–13634, 2024.

Fangwei Zhong, Kui Wu, Churan Wang, Hao Chen, Hai Ci, Zhoujun Li, and Yizhou Wang. Unrealzoo: Enriching photo-realistic virtual worlds for embodied ai. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5769–5779, 2025.

Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7641–7649, 2024.

Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pp. 3357–3364. IEEE, 2017.

Zilin Zhu, Longteng Guo, Yanghong Mei, Bowen Pang, Zongxun Zhang, Xingjian He, Ruyi Ji, and Jing Liu. When robots do the chores: A benchmark and agent for long-horizon household task execution. *arXiv preprint arXiv:2605.14504*, 2026.

Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart, Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. In *Conference on Robot Learning*, pp. 2165–2183. PMLR, 2023.

Ding Zou, Feifan Wang, Mengyu Ge, Siyuan Fan, Zongbing Zhang, Wei Chen, Lingfeng Wang, Zhongyou Hu, Wenrui Yan, Zhengwei Gao, et al. Embodiedbrain: Expanding performance boundaries of task planning for embodied intelligence. *arXiv preprint arXiv:2510.20578*, 2025.

A Details of the Simulators

A.1 AI2-THOR

The House Of inteRactions (THOR) is a Unity 3D-based simulator originally proposed for the task of target-driven visual navigation (Zhu et al., 2017). It is subsequently extended to iTHOR (Kolve et al., 2017), which provides 120 indoor scenes, comprising 30 kitchens, 30 bathrooms, 30 living rooms, and 30 bedrooms. The environment contains $\sim$100 categories of interactable objects, with multiple object models per category, together with a rich set of object properties (*e.g.*, hot and cold, clean and dirty), interaction affordances (*e.g.*, mugs can be placed in coffee machines), and physics-based dynamics (*e.g.*, dirty object will become clean under running water). The embodied agent is represented as a virtual, capsule-shaped entity, and its physical body is not rendered in the egocentric observation. The agent can interact with objects within a predefined distance range. At most one object can be directly held at a time. Once picked up, the object is attached to the agent's virtual hand and remains visible in front of the camera, typically occupying the lower portion of the egocentric view. ProcTHOR (Deitke et al., 2022) further introduces a procedural generation pipeline for constructing diverse environments, which can produce an unbounded number of semantically plausible scene layouts and populate them with objects in reasonable configurations. The resulting environments preserve the same object interaction semantics and affordances as those used in iTHOR.

In UniETP, we use AI2-THOR v5.0.0 and employ the original 120 single-room layouts provided by iTHOR, and the ProcTHOR-10k multi-room scene dataset pre-generated by ProcTHOR, following their original train/test splits. When constructing the initial scene for each task, we leverage AI2-THOR's built-in randomization of movable-object placements, lighting conditions, and object materials to further increase scene and visual diversity. Regarding the action space, UniETP largely inherits the interaction APIs provided by the "Controller" class in AI2-THOR for both movement and atomic manipulation actions. Goal-directed navigation determines the teleportation target by sampling interactable poses around the intended object. Composite manipulation actions first check the object category and relevant affordances (*e.g.*, a stove burner can be used to heat a pot, while a fridge can be used to cool an apple), and then execute a sequence of atomic actions or directly modify the object states. UniETP's action execution makes full use of the object states supported by AI2-THOR, including open/closed, powered on/off, filled with liquid, cooked, clean/dirty, and temperature. By default, RGB observations are rendered at a resolution of 500×500 with a 90° field of view. The movement step size is 0.25 m, while camera-view adjustments use a step size of 30°.

A.2 VirtualHome

VirtualHome (Puig et al., 2018) was originally developed to model and analyze complex human activities in household environments (Liao et al., 2019; Puig et al., 2021). The latest release, VirtualHome v2.3, provides two complementary simulation backends: a Unity3D-based simulator and an evolving graph simulator. The former provides a visually grounded embodied environment, in which cameras can be attached to agents or placed at fixed locations in the scene to render visual observations while actions are executed through animated interactions. In contrast, the Evolving Graph simulator runs entirely in Python and represents the environment as a scene graph (SG). Actions are simulated by updating object states, represented as node attributes, and inter-object relations, represented as edges, without rendering visual observations. VirtualHome v2.3 provides 50 custom-designed household scenes populated with a diverse set of objects, together with a rich collection of object properties, object states, and inter-object relations. It defines $\sim$40 categories of atomic actions, but many of them do not induce changes to object states. All action categories are supported by the evolving graph simulator, whereas only a subset can be physically instantiated and rendered in the Unity simulator. VirtualHome employs humanoid avatars as embodied agents. At the simulator level, an agent can simultaneously hold up to two objects, one in each hand. For consistency with the other environments, we restrict the maximum number of simultaneously held objects to one in our environment layer.

UniETP is implemented on top of VirtualHome v2.3 and uses the 50 scene layouts provided by the platform. We largely follow the action interfaces of the VirtualHome Unity simulator, including basic navigation and manipulation actions such as grasping, placing, opening, closing, switching.

Turning is implemented by directly invoking the camera extrinsics interface. For goal-directed navigation, after calling VirtualHome's official interface for moving the agent to the target object, we further adjust the camera orientation to center the target object in view. By default, RGB observations are rendered at a resolution of 640×480, with the field-of-view parameter set to 60°.

A.3 HABITAT

Habitat (Savva et al., 2019) provides a flexible and high-performance embodied simulation platform built upon a variety of 3D scene datasets, supporting multimodal sensory observations through its underlying Habitat-Sim simulator. Habitat 2.0 (Szot et al., 2021) further introduces interactive environments based on ReplicaCAD and proposes the Home Assistant Benchmark for mobile manipulation centered on object rearrangement. Habitat 3.0 (Puig et al., 2024) extends the platform to human-robot collaboration and introduces two new tasks: Social Navigation and Social Rearrangement. HSSD-200 (Khanna et al., 2024) is a synthetic scene dataset designed for use with the Habitat simulator. It contains more than 200 high-quality synthetic 3D indoor scenes populated with $\sim$20,000 high-quality 3D models of real-world objects. Habitat supports multiple embodiments via imported URDF models. In our environment, we employ a Fetch mobile manipulator equipped with a single robotic arm as the agent. The robot can hold at most one object at a time.

UniETP is built on Habitat-Lab/Habitat-Sim v0.3.3. The scenes and objects come from HSSD-200, and we follow PARTNR (Chang et al., 2025) to import additional objects from OVMM (Yenamandra et al., 2023). Habitat provides direct control over the agent and camera pose, object transforms, and articulated joint states, and we utilize a translation step of 0.25 m, a yaw increment of 30°, and a head-pitch increment of 10°. For navigation, we use Habitat's "Pathfinder" for safe stepping and "navmesh" snapping. For object placement, we sample candidate poses from the target receptacle and validate them using support, collision, visibility, and distance checks. In particular, Habitat supports part-level interaction. For example, a cabinet with multiple doors can be opened at the level of an individual part, and an object can be placed on/inside a particular part. In evaluation modes M1 and M2, visible articulated parts are associated with individual bounding boxes, allowing the agent to select a specific interaction region. In M3, the bounding boxes output for actions such as placing, opening, and closing are also matched against and grounded to concrete articulated parts. The default RGB observation resolution is set to 512×512.

A.4 BEHAVIOR

iGibson (Shen et al., 2021) is built on the Bullet physics engine. It provides 15 interactive, home-scale scenes modeled after real-world houses and populated with rigid and articulated objects. iGibson 2.0 (Li et al., 2022a) further introduces a rich set of object states, such as temperature, wetness, cleanliness, and sliced or toggled states, together with a generative mechanism that uses logical predicates and semantic rules to populate scenes with objects in semantically plausible configurations. BEHAVIOR (Benchmark for Everyday Household Activities in Virtual, Interactive, and ecOlogical enviRonments) (Srivastava et al., 2022) is a benchmark built on iGibson 2.0 that consists of 100 realistic and complex household activities. It introduces the BEHAVIOR Domain Definition Language (BDDL) to formally specify the objects, initial conditions, and goal conditions of each task, while supporting the sampling of valid environment initializations that satisfy these specifications. BEHAVIOR-1K (Li et al., 2024a) further expands the benchmark to 1,000 diverse everyday activities collected based on human needs and upgrades the simulation platform to OmniGibson, which is built on NVIDIA Omniverse and PhysX. OmniGibson supports realistic simulation and rendering of rigid bodies, deformable bodies, and liquids, substantially broadening the range of physical interactions that can be modeled. BEHAVIOR-1K also extends the state and predicate system to support a wider variety of object states and state transitions. The benchmark provides 50 high-quality, fully interactive scenes and more than 9,000 richly annotated object models spanning a large number of object categories. BEHAVIOR-1K supports multiple robot embodiments and we use the Fetch mobile manipulator, which can hold at most one object at a time.

UniETP is implemented with OmniGibson v3.7.2. It leverages the object assets and 50 fully interactive scenes provided by BEHAVIOR, and partitions them into non-overlapping training and benchmark scene sets. For action execution, we mainly follow OmniGibson's `SymbolicSemanticActionPrimitives` framework, which disables the CuRobo-based

motion-generation path and realizes actions by their semantic effects. Specifically, opening, closing, and toggling directly update the corresponding object states, with additional reachability and post-condition checks. Object placing reuses OmniGibson's predicate-based pose sampler, and handles object release, pose assignment, settling, and relation verification. Slicing invokes OmniGibson's `SlicingRule`, while it additionally synchronizes the newly generated objects with UniETP's SG representation. Grasping is handled kinematically by updating the object pose and maintaining an explicit grasped-object record, while navigation samples suitable poses from the room and traversability information and directly updates the robot pose, without executing low-level arm or locomotion trajectories. We further extend the primitive set to cover the rich state and particle interactions required by BEHAVIOR, including soaking, cleaning, filling, pouring, spreading, cooking, burning, washing, and cooling. These actions directly invoke OmniGibson's object-state, particle-system, and transition-rule APIs as appropriate. For example, filling and pouring explicitly manipulate simulated particles and verify the resulting liquid states, whereas cooking, burning, and cooling realize the corresponding temperature-dependent semantic states after checking the relevant object and tool affordances. The RGB observations are rendered at a default resolution of 512×512.

Examples of the tasks and trajectories in each simulator can be found in the visualizations in Section F.1.

B Comparison with Existing Datasets

B.1 ALFRED and Its Extensions

ALFRED (Shridhar et al., 2020) is a representative benchmark for embodied instruction following (EIF). Built on the 120 scenes in AI2-THOR, it involves 58 object classes and 26 receptacle classes, and defines 7 task types and 2,685 task instances. Each task instance is grounded in 3 randomly initialized scenes and the grounded tasks are annotated with expert trajectories and step-by-step instructions. The action space of ALFRED contains basic navigation commands and atomic manipulation actions. For each manipulation action, the agent must specify the target object using a pixel-level segmentation mask. Subsequent benchmarks (Padmakumar et al., 2022; Gao et al., 2022; Kim et al., 2024b;a; Jiang et al., 2025; Yin et al., 2024; Tupini et al., 2026; Chen et al., 2026a; Zhu et al., 2026; Zhan et al., 2026; Yang et al., 2026a; Chen et al., 2026b) further extend this setting by introducing other forms of interactions, realistic scenes, continual-learning settings, longer horizon, complexity of instructions, safety constraints, and multi-robot cooperations. **ALFWorld** (Shridhar et al., 2021) constructs TextWorld (Côté et al., 2018)-based abstract environments aligned with the embodied tasks in ALFRED, replacing ALFRED's low-level navigation and mask-based interactions with higher-level textual actions. Its tasks and environment configurations are derived from ALFRED and adapted into corresponding text-based games. ALFWorld provides aligned textual and visual embodied environments: in the visual setting, the agent operates from egocentric visual observations, whereas in the textual setting, it interacts with the environment entirely through text commands and receives textual feedback describing the visible objects and outcomes of its previous actions. With its compact and standardized text-based interaction interface, ALFWorld has become a widely used benchmark for evaluating planning and decision-making capabilities of LLM-based agents. It is also frequently incorporated as a representative embodied environment into broader generalist-agent benchmarks (Song et al., 2024; Xi et al., 2025; Li et al., 2025b). There are also subsequent works that focus on generating training trajectories and thinking traces (Zhang et al., 2025c).

Following ALFRED and ALFWorld, UniETP leverages the rich object states and affordances provided by AI2-THOR and adopts similar abstractions for navigation and manipulation. In addition, UniETP standardizes the action interface into multiple levels of granularity for evaluation at different difficulty levels. To better align with the output format of VLMs, manipulation actions are grounded using bounding boxes rather than pixel-level masks. UniETP also incorporates multi-room ProcTHOR environments to increase scene scale and spatial complexity, while broadening the task space through more diverse task templates.

LEGENT (Cheng et al., 2024) is aimed for scalable generation of embodied scenes, tasks, and trajectories. It supports both ProcTHOR-style procedural scene generation and LLM-guided scene generation adapted from Holodeck (Yang et al., 2024b). It includes tasks like navigation, object rearrangement, and embodied question answering. Built on the LEGENT simulator, **EmbodiedE-**

val (Cheng et al., 2026b) evaluates VLMs with a discrete action-option space: navigation is constrained to scene-specific navigation graphs, manipulation is represented as atomic actions, and question answering uses annotated candidate answers. It contains 328 tasks across 125 scenes from Objaverse-based synthetic scenes, AI2-THOR, HSSD, and Sketchfab. Task completion is determined by conjunctions of predicates over the final state or the interaction history.

LEGENT and EmbodiedEval both provide diverse scenes and embodied tasks, with their task coverage extending to embodied question answering and social interaction. UniETP instead places greater emphasis on diverse, long-horizon manipulation tasks and further extends the evaluation across multiple heterogeneous simulation environments. Moreover, rather than exposing the agent to a task-specific list of candidate actions, UniETP adopts a parameterized, multi-granularity action space and defines evaluation modes with varying levels of scene prior and action abstraction, enabling controlled evaluation at different difficulty levels.

B.2 BEHAVIOR AND ITS EXTENSIONS

Building upon BEHAVIOR and OmniGibson, **VisualAgentBench** (Liu et al., 2025c) (VAB) and **OctoVerse** (Yang et al., 2024a) both expose semantic action interfaces to facilitate task-level planning. VAB formulates 20 actions and 89 types of tasks (45 types are used for testing), and uses egocentric observation image with overlaid object bounding boxes. VAB-OmniGibson is combined with multimodal agent tasks from other domains to provide a comprehensive evaluation of VLM capabilities. OctoGibson defines 476 tasks based on a set of 16 atomic actions, uses images from multiple egocentric and static viewpoints as observations, and constructs action-and-reasoning trajectories through GPT-based trial-and-error exploration. OctoVerse further combines OctoGibson with tasks in Minecraft and GTA into a benchmark suite. **PbP** (Xu et al., 2025b) studies planning conditioned on user preferences inferred from demonstration videos. **IS-Bench** (Lu et al., 2025) and **SafeRelBench** (Yang et al., 2026b) introduce safety-aware household scenarios, and evaluate both outcome correctness and process safety. Hong et al. (2026) explicitly focus on potential planning failures caused by delayed consequences, such as premature commitments, spatial fitting conflicts, and sequential dependencies. It uses GPT-5 to deliberately generate tasks containing such failure-inducing structures, and incorporates 3D point-cloud observations to emphasize spatial reasoning. **DynamicEnvPlan** (Yuan et al., 2026) considers planning under environment deviations and execution uncertainty. Besides, several works (Mohammadi et al., 2025; Liu et al., 2025b; Onishchenko et al., 2025; Wang et al., 2026) extract scene graphs from BEHAVIOR scenes to serve as an interface for long-horizon planning.

Following these works, UniETP similarly builds an atomic action interface on top of OmniGibson, abstracting low-level control while making extensive use of the rich object states and affordances supported by the environment. Beyond that, UniETP integrates multiple heterogeneous simulators under unified observation and action interfaces, introduces evaluation modes with different levels of task difficulty and action abstraction, and provides a scalable, automated task-generation pipeline rather than relying on a fixed set of manually curated task instances for evaluation.

B.3 BENCHMARKS ON HABITAT

LangR (Szot et al., 2024) is an object rearrangement benchmark built on Habitat 2.0. It defines an action space of 70 action instances. At each timestep, the agent selects one action from a set of 70 candidate actions to execute. The benchmark defines 10 task types and implements an automatic task-instance generation pipeline. For instructions, it provides a large collection of instruction templates that incorporate object references based on attributes, functions, and spatial relations, as well as constructs such as conditional statements, universal quantifiers, and distracting information, thereby placing greater demands on the model's reasoning capabilities. **Ask-to-Act** (Ramrakhya et al., 2025) further introduces ambiguous instructions and allows the agent to query the user for clarification through a dedicated asking action. **ManiTaskGen** (Dai et al., 2025) further improves the task-generation pipeline, enabling more automated construction of diverse mobile manipulation tasks, including both process-based and outcome-based tasks. Compared with these works, UniETP leverages the larger and more diverse set of scenes in HSSD-200 and it supports a broader range of object interactions and a more diverse set of tasks by integrating the complementary strengths of multiple simulators.

Habitat-MAS (Chen et al., 2025) and **PARTNR** (Chang et al., 2025) both study multi-agent collaboration in HSSD scenes. Habitat-MAS incorporates robots with diverse embodiments and complementary mobility, perception, and manipulation capabilities. Its Robot Resume mechanism derives descriptions of robot capabilities from their physical specifications and kinematic tools to support embodiment-aware task planning and assignment, followed by decentralized parallel execution. PARTNR implements Python-based evaluation functions that represent task goals using propositions, proposition dependencies, and constraints. Its tasks cover constraint-free settings, spatial-relation constraints, temporal constraints, and heterogeneous agent-capability constraints. UniETP draws on PARTNR's design principles for task logic complexity. However, our task goal is deterministically derived from predefined task templates, rather than generated by an LLM as in PARTNR. UniETP currently focuses on long-horizon and complex single-agent embodied task planning, leaving multi-agent collaboration for future work.

B.4 Benchmarks in the Text Domain

The following benchmarks evaluate agents in a textual environment, where the observations and actions all take the form of natural language. UniETP instead focuses on task planning with visual observations, and places specific emphasis on the design of object grounding in its environment layer.

ET-Plan-Bench (Zhang et al., 2025b) constructs navigation and manipulation tasks in VirtualHome and systematically increases task difficulty by introducing spatial and temporal constraints. The benchmark provides an automated task-generation and evaluation pipeline, deriving executable tasks and ground-truth plans from simulator scene graphs. **Embodied Agent Interface** (EAI) (Li et al., 2024b) provides a unified framework for diagnosing the capabilities of LLMs in embodied decision making across VirtualHome and BEHAVIOR. It formalizes planning into four ability modules: goal interpretation, subgoal decomposition, action sequencing, and transition modeling, with standardized input-output specifications for each module. EAI evaluates these modules separately and introduces fine-grained metrics to identify errors related to each stage. **VestaBench** (Sadhu et al., 2025) further includes safety requirements. **AHAT** (Liu et al., 2026) extracts scene graphs from HSSD and Gibson scenes as a textualized environment representation. Besides, many studies on LLM-based planning also construct their own training or evaluation datasets on top of VirtualHome (Li et al., 2022b; Singh et al., 2023; Zhao et al., 2023; Huang et al., 2022; Xiang et al., 2023).

B.5 Integrated Benchmarks

LoTa-Bench (Choi et al., 2024) integrates an ALFWorld-style AI2-THOR environment with the VirtualHome environment. It systematically investigates factors such as model selection, prompt construction, in-context examples, domain-specific fine-tuning, and feedback-based replanning. However, LoTa-Bench primarily focuses on language-based planning and remains a benchmark in the text domain. **EmbodiedBench** (Yang et al., 2025b) (EB) contains four domains that evaluate embodied intelligence at different levels. Two domains are designed for low-level navigation and manipulation, while the other two focus on task planning. The EB-ALFRED domain extends LoTa-Bench to visual modality, and the EB-Habitat domain largely inherits LangR. It pays particular attention to generalization across task instructions, and introduces multiple evaluation splits to assess the agent's ability to interpret task goals based on visual appearance, commonsense knowledge, spatial relations, and other cues. **SpatialWorld** (Gao et al., 2026) is a very recent effort towards benchmark unification, while it places greater emphasis on spatial reasoning and covers a wide range of outdoor and digital game environments.

Compared with these existing efforts to integrate multiple simulators, UniETP incorporates a broader range of household simulation environments, provides more unified action and observation spaces, defines controllable evaluation modes with varying levels of difficulty, and develops an automated task-generation framework that is compatible across heterogeneous environments.

B.6 Benchmarks with Other Simulators

Beyond the simulators considered in UniETP, there are also various benchmarks built upon other backends, scenes, and assets, such as ThreeDWorld (Gan et al., 2021; 2022; Zhang et al., 2023),

Table 8: Some examples of the task templates.

Atomic	**Compositional**	**Temporal**	**Logical**	**Numerical**
Open one A.	Put one A and one B in C.	Open one A, then put it in B.	Put all A in one B.	Put exactly two A on one B.
Turn on one A.	Put one open A and one B on C.	Slice A, then cook one slice.	Remove all A from B.	Put at least two A and at least one B in one C.
Cook one A.	Cook one A and put it on B on C.	Put A in B, then power on B.	For every A, place it in B or on C.	

Isaac Sim (Wang et al., 2024; Li et al., 2025a), and Unreal Engine (Gao et al., 2019; Zhong et al., 2025). Besides, Ego2World (Cheng et al., 2026a) extracts symbolic environments from egocentric human activity videos. Another line of works (Lei et al., 2025; Kim et al., 2026; Che et al., 2026) focuses on creating diverse household scenes for task planning with generative AI.

UniETP marks a first step in unifying the environments for embodied task planning, and we plan to integrate support for more simulators in future work.

C Details of the Task Generation Process

There are basically two paradigms for constructing planning tasks: scene-to-task and task-to-scene. In the former, a realistic scene is first generated, and feasible tasks are subsequently constructed based on the configurations of the objects present in the scene. In contrast, the task-to-scene paradigm begins by specifying a task goal and then constructs a scene that satisfies the initial conditions required for the goal, thereby grounding the task definition into a concrete environment. We adopt the latter paradigm because it provides greater control over the distribution of task categories, making it easier to systematically regulate the difficulty and diversity of the benchmark.

The pipeline for automated task generation is briefly introduced in Section 3.3.1 and more details will be provided below. Here we first clarify a few frequently used terms: a *task template* contains symbolic object-category placeholders; a *task specification* instantiates those placeholders with simulator-supported categories; a *grounded task instance* binds the specification to an executable initial scene; and a final, *annotated task instance* additionally associates the grounded task instance with a natural-language instruction. The complete instance is $\tau = \langle I, s_0^i, L, \mathcal{E}^i \rangle$, where I is the instruction, s_0^i is the initial state in simulator $\mathcal{E}^i$, and L is the formal goal formula evaluated on the resulting trajectory.

C.1 Task Template and Formal Goal Definition

A task template is a simulator-agnostic, parameterized task specification, formally defined as $\mathcal{T} = \langle n, \Theta, \mathcal{C}, G, d, \mathcal{I}_0 \rangle$. Here, n is a unique template name and some examples are shown in Table 8. $\Theta = \{\theta_k\}$ is a set of object-category placeholders such as `A`, `B`, and `C`. $\mathcal{C}$ contains constraints for the object categories to be determined. G is a factory that constructs the goal formula after the placeholders are instantiated. d is a natural-language task description (with placeholders). $\mathcal{I}_0$ is a set of declarative initial-state requirements, which may specify object cardinalities, initial object states, required or forbidden spatial relations, and the availability of suitable tools. Some examples of template definitions are presented in Table 9.

The goal factory compiles a template into a compositional formula over a trajectory of Unified Scene Graphs (USGs). Atomic predicates describe unary object states, such as `open`, `dirty`, or `filled`, and binary spatial relations, such as `inside` or `on`. The formula library supports existential and universal quantification, conjunction, disjunction, negation, and implication. It also supports temporal operators (*e.g.*, at end, before, until) as well as counting and numerical comparison.

Table 9: Examples of the components in task templates.

Template name	Category constraints	Goal schema	Initial-state requirements
`open_A`	A can be opened	$\exists x \in A:\ \mathsf{End}[\mathsf{Open}(x)]$	At least one A; all A are closed.
`A_B_in_C`	A, B can be grasped; A, B can be placed in C	$\exists x \in A, y \in B, z \in C:$ $\mathsf{End}[\mathsf{In}(x,z) \wedge \mathsf{In}(y,z)]$	At least one A, one B, and one C; no A or B is initially in any C.
`fill_water_then_heat_A`	A can be filled with liquid; A can be heated	$\exists x \in A:$ $[\mathsf{FilledWater}(x) \prec \mathsf{Hot}(x)] \wedge$ $\mathsf{End}[\mathsf{FilledWater}(x) \wedge \mathsf{Hot}(x)]$	At least one A; at least one compatible water-filling tool; at least one compatible heating tool.
`all_A_in_B_or_on_C`	A can be grasped; A can be placed in B; A can be placed on C	$\exists y \in B, z \in C:\ \forall x \in A,$ $\mathsf{End}[\mathsf{In}(x,y) \vee \mathsf{On}(x,z)]$	At least three A; at least one B and one C.
`exact_2_A_on_B_and_exact_1_A_in_C`	A can be grasped; A can be placed on B; A can be placed in C	$\exists y \in B, z \in C:$ $\mathsf{End}[\#\{x \in A : \mathsf{On}(x,y)\} = 2$ $\wedge\ \#\{x \in A : \mathsf{In}(x,z)\} = 1]$	At least three A; at least one B and one C.

C.2 Affordance Knowledge Base and Task Instantiation

We construct one affordance knowledge base (AKB) for each simulator to represent their object taxonomies and interaction semantics. An AKB contains (i) the object categories exposed by the simulator, (ii) the states and functional properties supported by each category, and (iii) feasible binary relations between categories, including placement and tool use compatibility. These entries are compiled from simulator documentation and metadata, and missing entries are supplemented through LLM-assisted queries.

For a selected simulator $\mathcal{E}^i$, the instantiator resolves the placeholders in Θ by querying its AKB. Let $\mathcal{O}^i$ be the object category set of $\mathcal{E}^i$. Once a complete assignment $\alpha : \Theta \to \mathcal{O}^i$ is found, it is applied to the template fields. In particular, string substitution produces the instantiated description d_α; invoking the goal factory produces the formal goal $L_\alpha := G(\alpha)$; and placeholder substitution produces the grounded initial requirements $\mathcal{I}_{0,\alpha}$. The resulting task specification is therefore $\sigma^i_\alpha = \langle n, \alpha, d_\alpha, L_\alpha, \mathcal{I}_{0,\alpha}, \mathcal{E}^i \rangle$. This stage is still category-level: it may specify, for example, *open one microwave*, but it does not yet identify a concrete microwave instance or an initial scene.

C.3 Scene Grounding and Instance-Reference Construction

Given a specification σ^i_α, the scene sampler selects a base scene provided by simulator $\mathcal{E}^i$ and attempts to enforce every requirement in $\mathcal{I}_{0,\alpha}$. The grounding operations include inserting required objects where supported, modifying object states and placements, removing forbidden relations, establishing required ones, ensuring that compatible tools are present, and optionally adding distractors. The resulting scene is retained only if it satisfies all initial-state requirements and simulator-specific validity checks; otherwise, another scene or specification is sampled.

For the instance grounding split, the scene sampler additionally searches for attributes or relations that distinguish coexisting instances. Supported reference cues include object state, material attribute, relative size, parent receptacle, child objects, room, and neighboring object. Valid cues, together with the corresponding object-instance identifiers, are recorded for subsequent instruction generation. During evaluation, these identifiers restrict the goal logic's quantification domain from ranging over all scene objects of the target category to covering only the instances distinguished by the cues.

For reproducibility, each accepted scene stores the base scene identifier and the information needed to reconstruct the initialization, such as randomization arguments or the modified environment graph, object additions and placements, state-changing actions, and the robot pose.

Table 10: Details of the unified navigation action space. "obj" is short for the target object. ✓ and ✗ indicate whether an action is implemented or not under the corresponding evaluation mode or simulator.

Action	Arguments			Implementation			
	M1	M2	M3	THOR	VH	Habitat	BEHAVIOR
Move forward	-	-	-	✓	✓	✓	✓
Move backward	-	-	-	✓	✗	✓	✓
Move left	-	-	-	✓	✗	✓	✓
Move right	-	-	-	✓	✗	✓	✓
Turn left	-	-	-	✓	✓	✓	✓
Turn right	-	-	-	✓	✓	✓	✓
Look up	-	-	-	✓	✗	✓	✓
Look down	-	-	-	✓	✗	✓	✓
Turn base to viewing direction	-	-	-	✗	✗	✓	✗
Reset head direction	-	-	-	✗	✗	✓	✗
Turn to point	✗	✗	pixel	✓	✗	✓	✓
Turn to object	obj ID	obj ID	obj bbox	✓	✓	✓	✓
Go to room	room ID	room ID	room ID	✓	✓	✓	✓
Go to object	obj ID	obj ID	obj bbox	✓	✓	✓	✓
Done	-	-	-	✓	✓	✓	✓

C.4 Feasibility Validation and Expert Trajectory Generation

Scene construction establishes the requested initial conditions but does not by itself guarantee that a task can be completed under the simulator's dynamics and action interface. We therefore validate the grounded task instances with a scripted expert policy π^*. The expert associates each template with a symbolic plan sketch composed of operators such as finding an object or tool, grasping or placing it, and changing its state. The category assignment and the instance-reference (when present) then ground the plan sketch to concrete simulator identifiers. The expert executes in evaluation mode M1. It has access to the underlying USG of the scene and maintains the current progress of the plan schema. It works in a closed loop rather than replaying a fixed action sequence, and can therefore retry searches or manipulation and use backend-specific fallbacks when an interaction fails (*e.g.*, when an object is occluded or not suitable for interaction).

After the plan sketch is accomplished, the trace of recorded USGs is evaluated against the goal logic. A candidate task instance is retained only if the evaluator reports success and the episode contains more than one environment step, which filters out initially satisfied and degenerate tasks. On the training set, the same expert policy is employed and the execution log is converted to step-level SFT samples with the prompt template.

C.5 Instruction Generation

We use off-the-shelf LLMs to generate instructions for the validated, grounded task instances. The prompt includes the instantiated task description, a readable rendering of the formal logic, the instance-reference information (when present), and the set of object categories in the scene. The model is requested to produce three semantically equivalent instructions: (1) **L1-Direct**. It names the target categories explicitly and states all required counts, relations, and temporal ordering as directly as possible. (2) **L2-Referential**. It preserves the category-level goal but may refer to a category through its function, common use, or a natural paraphrase. The prompt requires the description to be uniquely resolvable from the listed scene categories and forbids hallucinated attributes or extra goals. (3) **L3-Syntactic**. It introduces more varied syntax and task-relevant context, and may include non-goal background or a limited distractor clause, while preserving exactly the same required goal and ordering constraints.

For the linguistic split, one of L2 or L3 is sampled as the instruction, while the other splits of the benchmark use L1. The generated instructions are manually reviewed for clarity and agreement with the task logic.

Table 11: Details of the unified manipulation action space. "obj" is short for the target object. "recep" is short for the receptacle when the agent is holding an object. "tool" is the target tool when the agent is holding an object.

Action	Arguments			Implementation			
	M1	M2	M3	THOR	VH	Habitat	BEHAVIOR
Pick up	obj ID	obj ID	obj bbox	✓	✓	✓	✓
Place on	recep ID	recep ID	recep bbox	✓	✓	✓	✓
Place in	recep ID	recep ID	recep bbox	✓	✓	✓	✓
Place to point	✗	✗	pixel	✓	✗	✓	✓
Drop	-	-	-	✓	✗	✓	✓
Open	obj ID	obj ID	obj bbox	✓	✓	✓	✓
Close	obj ID	obj ID	obj bbox	✓	✓	✓	✓
Turn on	obj ID	obj ID	obj bbox	✓	✓	✗	✓
Turn off	obj ID	obj ID	obj bbox	✓	✓	✗	✓
Pour to	obj ID	obj ID	obj bbox	✓	✗	✗	✓
Slice	obj ID	obj ID	obj bbox	✓	✗	✗	✓
Wipe	obj ID	obj ID	obj bbox	✗	✗	✗	✓
Spread to	obj ID	obj ID	obj bbox	✗	✗	✗	✓
Move hand left	-	-	-	✓	✗	✗	✗
Move hand right	-	-	-	✓	✗	✗	✗
Rotate hand	-	-	-	✓	✗	✗	✗
Heat with tool	tool ID	✗	✗	✓	✗	✗	✗
Cool with tool	tool ID	✗	✗	✓	✗	✗	✓
Wash with tool	tool ID	✗	✗	✓	✗	✗	✓
Cook with tool	tool ID	✗	✗	✓	✗	✗	✓
Fill with tool	tool ID	✗	✗	✓	✗	✗	✓
Burn with tool	tool ID	✗	✗	✗	✗	✗	✓
Soak with tool	tool ID	✗	✗	✗	✗	✗	✓

D Details of the Evaluation Process

D.1 Environment Details

The environment layer of UniETP manages the lifecycle of each episode, which includes scene initialization, returning RGB images and auxiliary observations, receiving an action type together with its associated parameters, and returning the action success status, feedback, termination flag, and the observation for the next step. An episode terminates when the maximum number of steps is reached or when the agent issues the terminal action 'done'. After each action, the Unified Scene Graph (USG) is extracted and recorded. At the end of the episode, the complete USG trace is passed to a Python-based logical evaluation system, which determines whether the task goal has been satisfied.

Instance segmentation rendering is implemented in each simulator. The rendered segmentation masks are used to compute 2D bounding boxes for the objects in view. In evaluation mode 3, the bounding box predicted by the agent is compared against the boxes of all objects within the view using Intersection over Union (IoU). If the maximum IoU exceeds 0.1, the object with the highest IoU is selected as the target of the action; otherwise, the action is considered failed.

The maximum steps and maximum failed steps are set to 50/10 for M1 and M2, and 150/30 for M3.

D.2 Action Space

The full action space for UniETP's environment layer is shown in Table 10 and Table 11. Each simulator implements a subset of the shared action space according to their backend state support and rendering capabilities. The same action may require different parameterizations under different evaluation modes.

D.3 PROMPT DESIGN

At each environment step, the agent receives the task instruction, the current RGB observation, the structured scene information permitted by the evaluation mode, and the action history. Below we show the full prompt template.

The following system prompt is shared across model backends. The action space is detailed in the previous subsection.

```
You are an embodied agent operating in a simulated environment.
Your goal is to complete a task described by a natural language
    instruction by executing a sequence of actions.
At each step, you must decide the next action(s) based on the current
    observation, action history, and environment information.

---

## Available Actions

{action space definition}

---

## Constraints and Hints

1. Avoid repeating ineffective actions from history. Prefer actions that
    make progress toward completing the task.
2. You can hold at most one object in your hand at a time.
3. In order to perform actions such as "place", "pour", or "spread", you
    need to have the corresponding object in hand. In order to perform
    actions such as "slice", you need to have a proper tool in hand.
4. Put your reasoning in the "reasoning" field. Keep it concise and avoid
     repeated thinking. The reasoning field can be an empty string when
    it is unnecessary.
5. If the task is already completed, output the "done" action to finish
    the episode.
```

We also provide some descriptions on the output format for each specific evaluation mode:

```
Mode 1:

1. For navigation actions, you can go to any object or room in the scene.
2. For manipulation actions, the target object must be visible in the
    current observation, and its ID must appear in the "Objects currently
     in view" list. In addition, you must be sufficiently close to the
    target object.
```

```
Mode 2:

1. For navigation actions, you can go to any object or room with a known
    ID.
2. For manipulation actions, the target object must be visible in the
    current observation, and its ID must appear in the "Objects currently
     in view" list. In addition, you must be sufficiently close to the
    target object.
3. The provided object and room ID lists contain only the large objects
    in the scene. If a required small object is not listed, explore the
    scene using navigation or fine-grained movement until the object
    appears in view.
```

```
Mode 3:
```

```
1. When an action uses a bounding box, use the format {bbox_format}. When
     an action uses a pixel, use the format {pixel_format}.
2. For navigation actions, use the bounding box of an object in the
    current view to get close to that object.
3. For manipulation actions, first get sufficiently close to the target
    object or tool, and then use its bounding box as the action argument.
     The bounding box must correspond to the latest observation image
    after the previous action. Therefore, navigation and manipulation
    actions should normally not be generated consecutively in the same
    output.
```

The following user prompt provides the task-specific information for the current episode. The scene information may contain the list of rooms/objects in the scene, and the object bounding boxes in the view. The action history contains the previously generated actions and their corresponding feedback.

```
Task:
{task instruction}

Observation:
<attached image>

{scene information}

Action history:
{action_history}

---

Please output the next action(s) you should take.
```

D.4 Model Details

For proprietary models, we call GPT, Gemini, and Claude models through their respective APIs. We set the sampling temperature to 0.2 and the maximum number of generated tokens to $2{,}048$. We use the default thinking effort. If the generation terminates because the token limit is reached, we issue one additional request with the thinking level explicitly set to low. We employ the structured outputs mode and provide a Pydantic schema for the output in the request. The schema contains a textual `reasoning` field and an `actions` array, where each action specifies an `action_name` and a typed argument object.

The open-source models (Qwen, Intern, Gemma) are served locally through OpenAI-compatible endpoints backed by vLLM. We use the same temperature, maximum tokens, and structured output mode as the proprietary models. If a vLLM response terminates with `finish_reason=length`, we issue one retry with thinking explicitly disabled while retaining the same prompt, token limit, and JSON schema. Outputs are validated against the same Pydantic action schema. When it fails, the parser subsequently attempts code-fence removal, extraction of an embedded JSON value, JSON repair, and normalization of single-action or top-level-list variants.

For other open-source models specifically designed for embodied tasks, we largely follow the inference procedures provided in their original codebases, deploying them with vLLM or loading them locally through the HuggingFace interface. We observe that these models generally perform worse on UniETP than general-purpose VLMs. One possible reason is that the observation and action representations used during their training differ from UniETP's unified interface, while their domain-specific training may limit their ability to follow complex instructions and previously unseen action protocols. To reduce the impact of formatting errors and interface mismatches on evaluation results, we introduce several lightweight prompting and post-processing strategies based on common failure patterns observed for these models. These include mapping bare object category names in model outputs to valid object IDs, normalizing action names, removing redundant action arguments, explicitly instructing the model to avoid unnecessary actions, adopting output formats that better match the model's capabilities, and applying JSON repair when necessary.

Table 12: Ablation study on the effect of action success signal (Su.) and failure feedback (Fe.).

		M1						M2					
Su.	Fe.	At.	Co.	Lo.	In.	Li.	Avg.	At.	Co.	Lo.	In.	Li.	Avg.
✓	✓	**93**	**62**	55	37	**59**	**56**	**67**	**34**	16	**22**	**26**	**28**
✓	✗	87	**62**	**57**	**41**	54	**56**	60	32	**20**	**22**	22	27
✗	✗	80	32	28	26	46	36	20	17	12	11	15	14

Table 13: Generalization performance on the ALFWorld benchmark's `eval_in_distribution` (ID) and `eval_out_of_distribution` (OOD) splits.

Model	ID	OOD	Overall
Qwen3.5-4B	25.7	20.9	23.4
+SFT on UniETP	**38.6**	**29.9**	**34.3**

E Details of the Training Process

To obtain the results in Table 6, we build a training dataset with expert trajectories following the pipeline in Section C. The training set contains 16k annotated task instances, of which 4.6k have instance-level references. The expert trajectories are transformed into 145k step-level training samples. We also hold out 3,142 samples from 351 episodes for validation. We apply SFT to a Qwen3.5-4B base model using ms-swift, with the vision module frozen. Training uses 8 NVIDIA H20 GPUs with an effective batch size of 32. We train for 5 epochs using AdamW with a learning rate of 10^{-6}, weight decay of 0.01, cosine learning-rate decay, and a 3% warm-up ratio. During training and inference, reasoning is disabled. We also use a more concise prompt template than that shown in Section D.3, which ignores some fixed hints and constraints.

F More Experimental Results

F.1 Visualization

In Figures 6-13, we present the visualization of some tasks in each simulator and the successful trajectories produced by the tested models.

F.2 Analysis on the Effect of Action Feedback

Table 12 presents an ablation study on action feedback, using Qwen3.6-27B. The experiments are conducted on the AI2-THOR split of UniETP benchmark. Under both M1 and M2, removing textual failure feedback has little impact on the overall performance. We observe in experiments that strong models are often able to infer plausible causes of failure, whereas weaker models have limited ability to recover once a failure occurs. As a result, textual explanation is not always effectively utilized by the agent. In contrast, removing the binary action success indicator leads to a clear performance drop. This suggests that current agents still have difficulty inferring whether an action has been successfully executed solely from visual observations, highlighting the importance of explicit post-verification mechanisms in embodied agentic systems.

F.3 Generalization

In Table 7 we have shown the results on EmbodiedBench (EB) to validate the generalization ability of the model trained on UniETP. Here we supplement an experiment on ALFWorld (Shridhar et al., 2021) (AW) in Table 13, which also proves the benefit of UniETP. We note that EB and AW take different forms of observation and action spaces. Specifically, EB-ALFRED allows the agent to teleport to any object and provides a list of all objects in the scene through the candidate action list, while it disallows high-level composite manipulation actions. EB-Habitat only permits teleportation to furniture and provides a list of all furniture in the scene. Its manipulation space is limited to pick, place, open, and close actions. Similarly, ALFWorld only allows teleportation to furniture



Figure 6: A task example on AI2-THOR, evaluation mode M2.



Figure 7: A task example on AI2-THOR, evaluation mode M1.

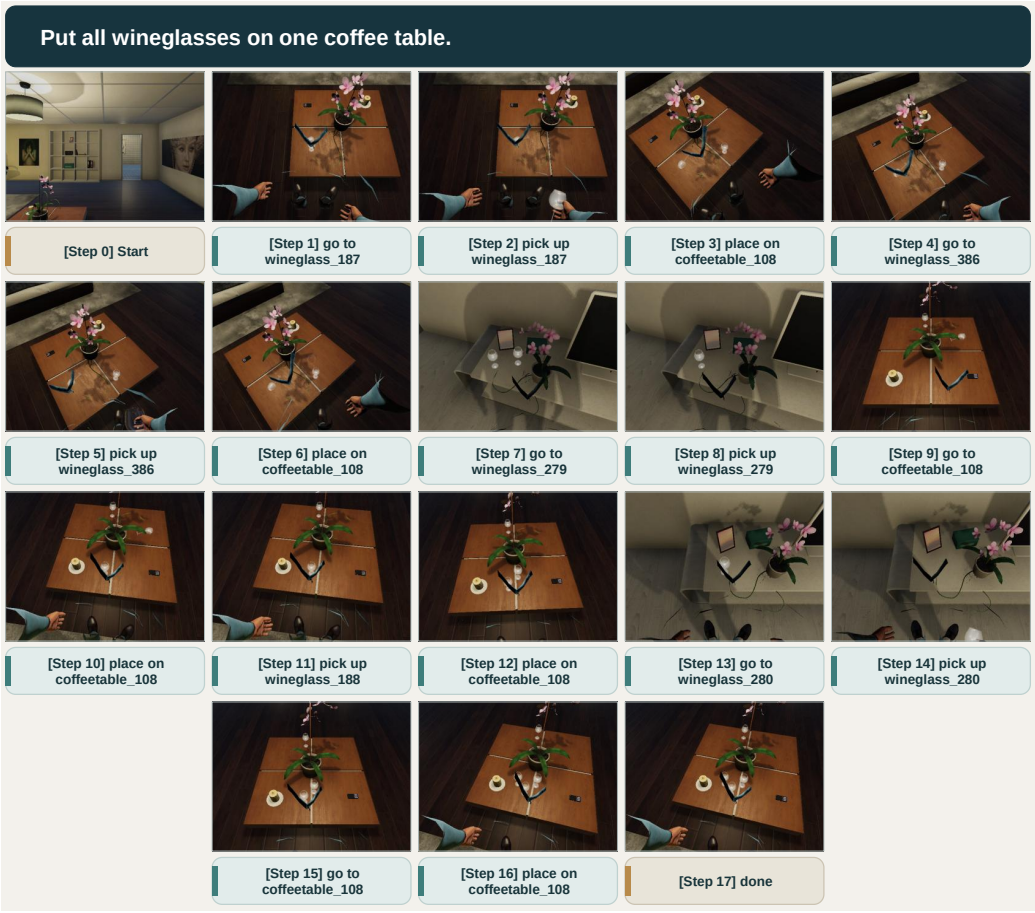


Figure 8: A task example on VirtualHome, evaluation mode M1.

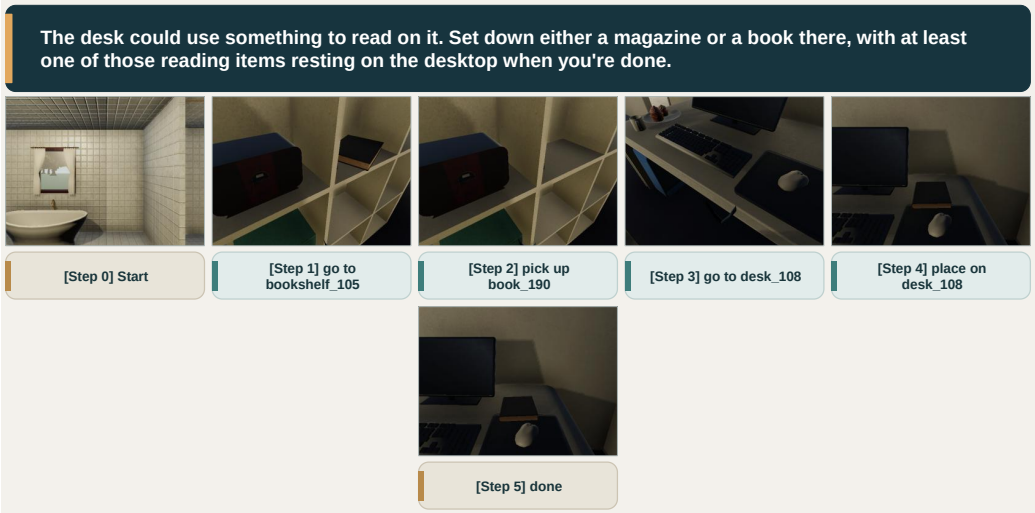


Figure 9: A task example on VirtualHome, evaluation mode M2.

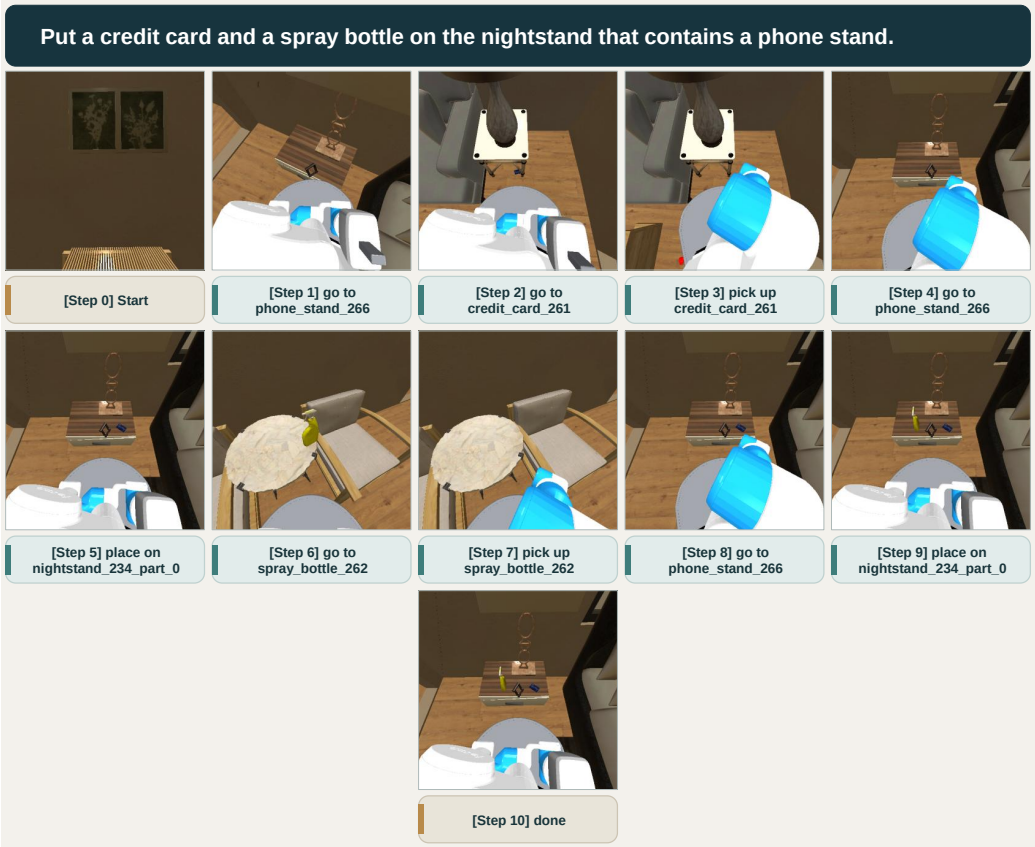


Figure 10: A task example on Habitat, evaluation mode M1.

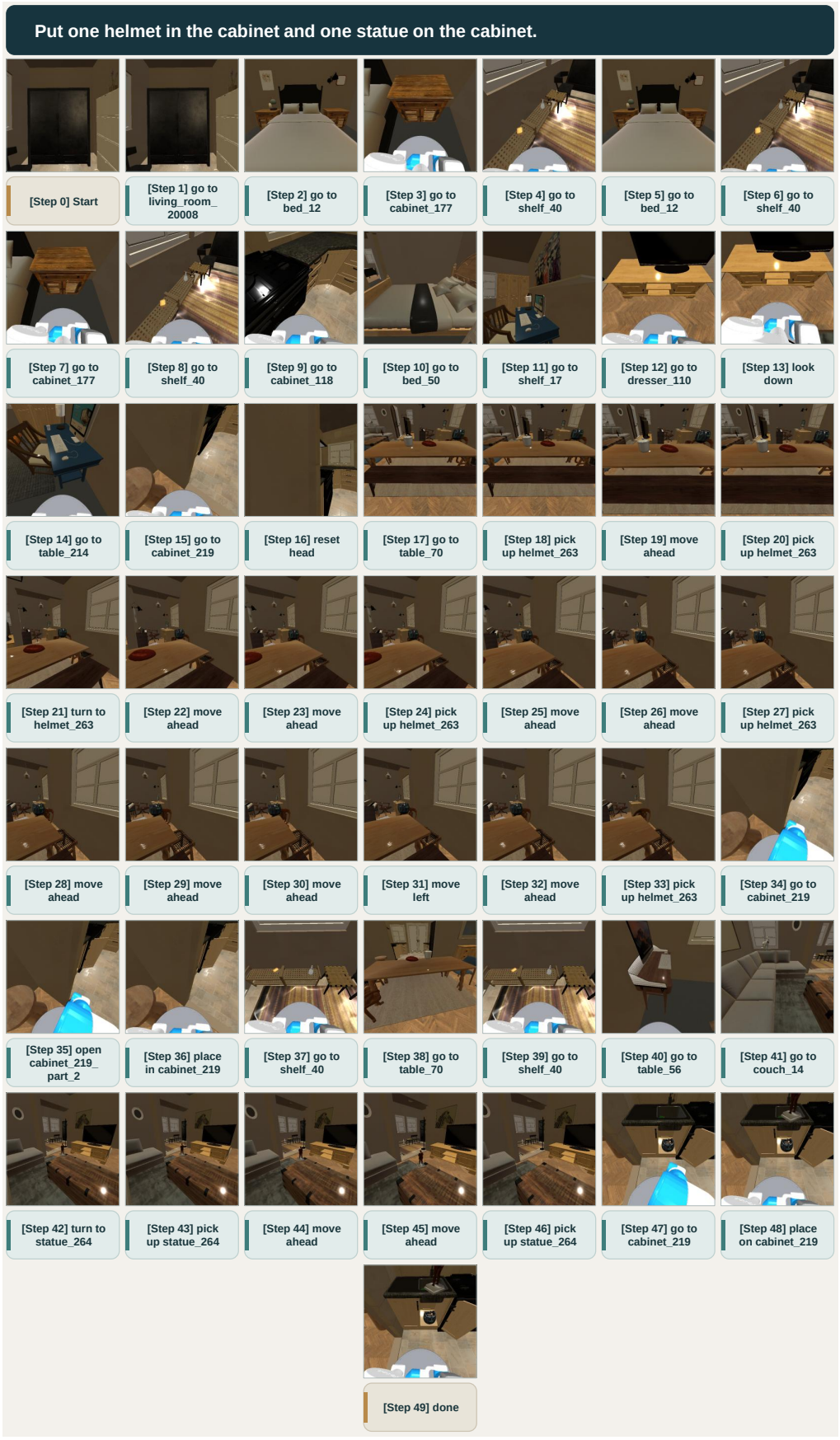


Figure 11: A task example on Habitat, evaluation mode M2.

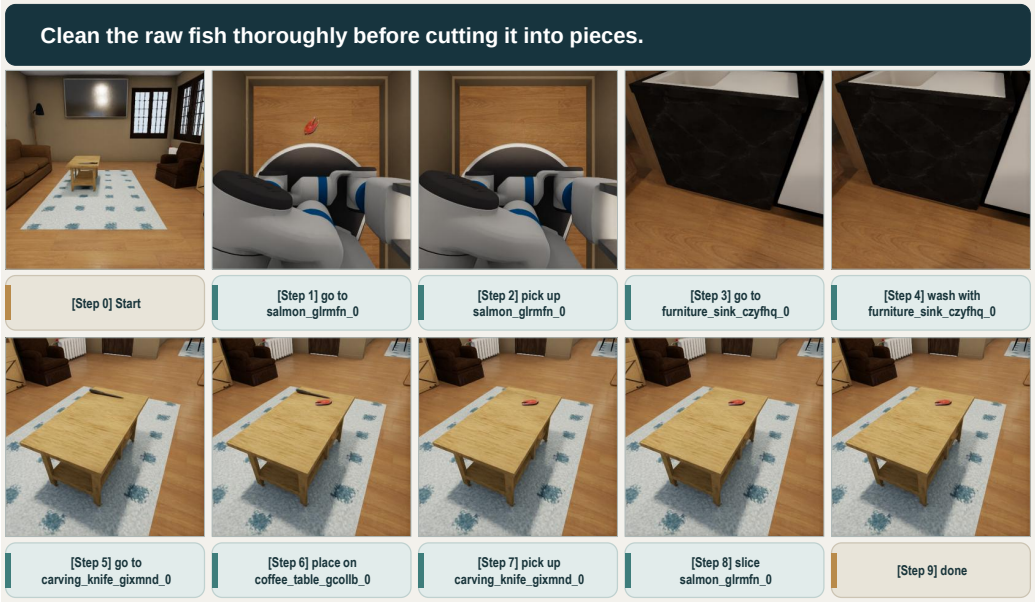


Figure 12: A task example on BEHAVIOR, evaluation mode M1.



Figure 13: A task example on BEHAVIOR, evaluation mode M2.

and exposes a list of all furniture, but additionally supports high-level, tool-assisted manipulation actions. Since the interfaces of these benchmarks do not strictly align with either the M1 or M2 interface of UniETP, we make benchmark-specific adaptations to both expert trajectory generation and the training-data format. For ALFWorld, we use the expert policy under the M2 mode and additionally introduce a random exploration procedure. For EB, we largely follow the expert policy for the M1 mode described in Section C, but replace each composite action with a sequence of atomic actions. We also remove the input bounding-box list from the training samples and randomly drop a subset of action types that are not used in the current episode from the action-space specification.

F.4 Error Analysis

We manually examine the failure cases of the frontier models and identify the following recurrent failure modes. (1) **Reference grounding**. The instance grounding split is particularly challenging for all models, where the agent often identifies the correct object category but selects an instance that does not satisfy the relational description in the instruction. (2) **Goal understanding**. The agent may successfully complete several subgoals while failing to maintain the full set of logical constraints, especially when the goal logic involves universally quantified conditions or compound requirements on object locations and states. (3) **Invalid placement**. When placing objects, the agent often overlooks constraints and details, such as attempting to place objects to a crowded container or one without sufficient interior space, or targeting the wrong part when interacting with an articulated object. It may also confuse "on" and 'inside" relations, resulting in placements that fail to satisfy the specified goal conditions. (4) **Invalid grasping**. Similarly, the agent may repeatedly attempt to grasp an object when it is too far away, occluded, or while already holding another object. (5) **Affordance understanding**. The agent may misinterpret object functionalities, for example by attempting to open non-openable objects, or by ignoring action preconditions or selecting inappropriate tools when executing composite manipulation actions. (6) **Object searching**. This issue frequently arises in evaluation mode M2, where teleportation to small objects is disabled and the agent must actively explore the scene to locate target objects. Besides, across the above categories, a common weakness is the failure to make effective use of action feedback. After receiving a failure signal, the agent may repeatedly retry the same invalid action instead of revising its overall plan. In some cases, the agent overconfidently outputs a sequence of actions ending with "done", and the episode may terminate in failure if an earlier action in the sequence fails.

G Limitations and Future Works

Despite its broad coverage, UniETP currently focuses on tasks with formally specified and unambiguous goals. Real-world requests are often more open-ended, subjective, or underspecified, and extending the benchmark to such settings may require carefully calibrated LLM-as-a-Judge protocols or interactive goal clarification between robots and users. Also, the current environments are static and the benchmark focuses on single-agent planning, leaving dynamic changes caused by external actors and multi-agent coordination underexplored. In addition, although the linguistic and instance grounding splits already cover diverse descriptions of object categories and instances, task references could be further refined using spatial relations, user preferences, and other scene or context-dependent constraints. In future works, we plan to integrate more simulators and domains beyond household environments to further extend the data scale and diversity. We will also explore combining high-level planning with low-level continuous control and replacing oracle actions with realistic skills, which will enable end-to-end, closed-loop evaluation of complete embodied systems.