Exploratory, Communicative, and Deployable: Vision-Driven Embodied Agents for Open-World Mobile Manipulation

Boyu Mi[1,2⋆], Mengchen Ma[2,3⋆], Yifei Yao[1,2⋆], Xing Gao[2✉], Junting Chen[4], Yangzi Li[5], Zihou Zhu[2], Guohao Li[6], Zhenfei Yin[7], Tai Wang[2], Yao Mu[1,2], Jiangmiao Pang[2], and Hanqing Wang[2✉]

[1]Shanghai Jiao Tong University, Shanghai, China
[2]Shanghai Artificial Intelligence Laboratory, Shanghai, China
[3]Southeast University, Nanjing, China
[4]National University of Singapore, Singapore
[5]Zhejiang University, Hangzhou, China
[6]Camel AI, https://CAMEL-AI.org, United Kingdom
[7]University of Oxford, Oxford, United Kingdom

Abstract. Real-world deployment of embodied agents requires active exploration, visual grounding, and interactive intent disambiguation. However, existing frameworks often rely on privileged simulator states or assume complete instructions, bypassing realistic deployment challenges. To bridge this gap, we present **REAL**, an agentic framework for open-world mobile manipulation. **REAL** establishes sim-to-real-consistent environment APIs without oracle perception and integrates a simulated user to enable human-in-the-loop interaction. Within this environment, we design diverse task compositions to drive data collection, supervised fine-tuning, and online reinforcement learning, systematically optimizing agent performance. To comprehensively evaluate this approach, we introduce **REAL-Bench**, a benchmark spanning 241 tasks across active exploration, visual distraction, articulated manipulation, and interactive disambiguation. Experimental results demonstrate that our trained agent outperforms leading commercial closed-source VLMs on interactive tasks with a 56.9% success rate. Further empirical analysis reveals that our hierarchical training pipeline successfully aligns the model's tool-use capabilities while maintaining robust open-vocabulary reasoning under extended exploration horizons. Finally, we deploy and evaluate our framework on a physical dual-arm mobile robot, where it achieves a 78.3% end-to-end success rate over 60 real-world episodes. These physical trials demonstrate robust zero-shot transferability to unseen household scenarios, validating that our sim-to-real-consistent design successfully bridges the reality gap for long-horizon mobile manipulation. Code is available at https://github.com/InternRobotics/REAL.

Keywords: Embodied Agent Framework · Mobile Manipulation · Reinforcement Learning

⋆ Equal contribution; ✉ Corresponding authors: Xing Gao (gaoxing@pjlab.org.cn); Hanqing Wang (hanqingwang.c@gmail.com).

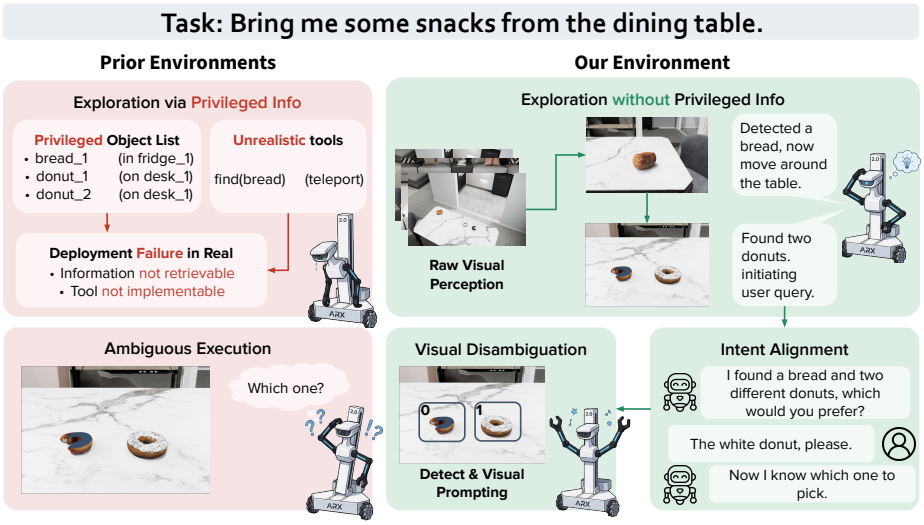

Fig. 1: Comparison of embodied agent frameworks. Left: Reliance on oracle perceptual APIs and open-loop execution in prior environments hinders ambiguity resolution and real-world deployment. **Right:** REAL removes oracle perceptual APIs while retaining physically obtainable scene priors, enabling RGB-based exploration and natural language interaction for ambiguous instructions.

1 Introduction

With their advanced reasoning and planning capabilities, Large Language Models (LLMs) have become a primary driver for embodied agents [1, 19, 20, 28, 29, 47]. Early frameworks primarily utilized these models to execute diverse household tasks within text-based environments built upon underlying physical simulators [24, 26, 38, 42, 43, 45] via textual perception and tool calling. However, pure-text interactions inherently lack the spatial and physical grounding critical for complex execution. This limitation has driven the widespread adoption of Vision-Language Models (VLMs) to endow agents with rich visual perception [4, 13, 46, 60]. By leveraging high-fidelity rendered imagery, modern vision-driven environments enable precise spatial reasoning and significantly expand the range of executable tasks [11, 51, 54]. Yet, a critical bottleneck persists: existing benchmarks often decouple simulation from real-world complexities across two fundamental dimensions—active exploration and communicative interaction (see Fig. 1, Left). Bypassing these realistic constraints inherently leads to deployment failures during sim-to-real transfer.

Specifically, the first limitation lies in the reliance on simulator-privileged perceptual information during physical exploration. Many existing environments equip agents with idealized tools that access global object lists [11, 28, 54] or locate targets via oracle category-finding APIs [51, 54]. Since these privileged interfaces are difficult to retrieve in physical environments, policies trained within

such setups can face performance discrepancies during real-world deployment. To address this, several recent works [5, 27, 40] adopt a non-privileged paradigm that restricts agents to standard sensor inputs like RGB images. However, their exploration and planning trajectories often operate primarily at the textual level rather than being tightly grounded in visual observations. Consequently, these methods can still struggle with precise visual disambiguation when encountering visually similar objects or distractors.

The second limitation stems from the assumption of explicit user instructions, leaving the need for dynamic intent alignment less addressed. Current benchmarks [6, 27, 51] typically treat execution as an open-loop process, assuming that the user possesses sufficient knowledge of the environment and issues unambiguous commands. In practical deployment, however, human users often operate under incomplete information, leading to initially ambiguous goals that require active, context-aware interaction to clarify intent. While recent studies [35, 36, 52] have introduced simulated users to facilitate interaction, their scope is primarily focused on text-based preference elicitation or memorization, leaving the co-optimization of communication and physical mobile manipulation tasks less explored.

To resolve these issues, we present **REAL** (exploRatory, communicativE, And, deployLable), an agentic framework designed for interactive and deployable open-world mobile manipulation. As illustrated in Fig. 1 (Right), REAL establishes a closed-loop visual environment entirely free of oracle perceptual APIs. The agent is restricted from accessing global object lists, target 3D poses, privileged simulator states, or goal locations. Instead, to ground objects, the agent must actively explore the environment and discover objects from raw RGB observations through a deployable toolchain spanning navigation, object detection, and visual prompting for manipulation [31]. In addition, we integrate a simulated user that issues initially ambiguous instructions and provides dynamic, context-aware natural language feedback. This feedback is tightly grounded in the agent's ongoing exploration results, effectively capturing the information blind spots inherent in real-world human interactions.

Within this environment, we develop an automated, scalable data generation and filtering pipeline to construct diverse exploration and interaction tasks. Utilizing the resulting trajectory data, we align and optimize the REAL agent based on Qwen3-VL-8B-Instruct [39] via a two-stage paradigm: supervised fine-tuning for tool-use alignment, followed by online reinforcement learning to enhance adaptive reasoning and error recovery. To systematically evaluate our agent, we introduce **REAL-Bench**, a comprehensive benchmark comprising 241 task instances across four distinct families: active exploration, visual distraction, articulated manipulation, and interactive disambiguation. Experimental results show that our trained agent successfully outperforms leading commercial closed-source VLMs, achieving a 56.9% success rate on interactive tasks. Finally, we deploy and evaluate our agent on a physical LIFT2 dual-arm mobile robot. The high-level policy achieves a 78.3% end-to-end success rate over 60 real-world

episodes, demonstrating robust zero-shot transferability and validating that our sim-to-real-consistent framework design effectively bridges the reality gap.

In summary, our main contributions are threefold:

1. **REAL framework.** We introduce **REAL**, a sim-to-real-consistent agentic framework without oracle perception, coupled with a simulated user for dynamic human-in-the-loop intent alignment.
2. **Training pipeline.** We develop an automated task-generation pipeline and train a Qwen3-VL-8B-based agent with dual-stage SFT and online RL, enabling robust tool-use alignment and adaptive replanning.
3. **Benchmark and deployment.** We construct **REAL-Bench** with 241 tasks and show that the trained agent outperforms strong commercial baselines in simulation while transferring zero-shot to a physical dual-arm mobile robot with a 78.3% success rate.

2 Related Work

2.1 Embodied Agent Environments and Benchmarks

Most embodied agent environments build on foundational simulators [12, 24–26, 38, 43, 45, 55]. The Embodied Agent Interface [28] and ManiTaskGen [11] evaluate LLMs or generate fine-grained tasks, but simplify exploration by providing ground-truth object lists. VisualAgentBench [30], EmbodiedBench [54], and VoTa-Bench [51] incorporate visual inputs yet rely on unrealistic primitives (*e.g.*, a "find" tool that teleports objects), hindering real-robot deployment. PARTNR [5] explores rooms via Concept Graphs [17] and EMMOE [27] uses low-level policies without privileged states, but both reference objects through text, causing ambiguity with visual distractors. In contrast, our environment eliminates oracle perceptual APIs and grounds discovered objects in visual space, enabling robust disambiguation and real-world-compatible exploration.

2.2 Vision-Driven Embodied Agents

Recent work has explored visually grounded embodied agents, ranging from LLM-based exploration agents such as Voyager [49] to VLA policies that directly condition on visual observations, such as OpenVLA [23] and $\pi_{0.5}$ [37]. In simulation-oriented settings, VoTa-Bench [51] applies dual preference optimization for state prediction and action selection; EMMOE [27] trains a DPO-based planner with lightweight policies [9] for mobile manipulation; ERA [6] distills embodied priors and refines agents through online RL; and OWMM-Agent [7] adapts VLMs to open-world mobile manipulation through simulation-based multimodal agentic data synthesis. However, these methods do not center on oracle-free active exploration and dynamic user interaction for acquiring missing task state. RoboOS [48] and Being-0 [56] both support real-world robot deployment. However, RoboOS does not explicitly train the agentic planning capability, while Being-0 delegates high-level planning to a closed-source

foundation-model API. In contrast, REAL trains its high-level VLM agent end-to-end in simulation, where a procedural environment and a data-generation pipeline generate planning-level training data at scale.

Classical task and motion planning (TAMP) integrates symbolic task planning with continuous motion feasibility and state tracking [14, 22]. LLM planners such as SayCan [1] ground language in predefined skills and affordances, while uncertainty-aware or replanning frameworks such as KnowNo [41] and RePLan [44] decide when to ask for help or recover from execution failures. REAL is complementary to this line of work: it does not solve the TAMP problem or model all uncertainty in low-level motion execution. Instead, REAL studies how to train a deployable visual-interactive policy that acquires missing task state through RGB exploration and user dialogue before dispatching physical skills.

2.3 Simulated User Modeling for Embodied Agents

Multi-agent systems have been widely investigated in embodied tasks to facilitate flexible collaboration. Beyond single-agent configurations, EMOS [8] proposes a heterogeneous multi-robot operating system driven by LLM agents. Similarly, REMAC [57] enables multi-agent long-horizon manipulation through self-reflection and self-evolution. Focusing on communication, CoELA [58] investigates natural language interactions among agents, albeit under the assumption of shared environmental awareness. These works highlight the importance of open-ended agent collaboration in complex environments. TEACh [34] establishes multi-turn dialogue for household tasks, and VL-LN [21] uses simulated users to resolve ambiguities via active communication. For user intent understanding, ADAPT [35] elicits preferences through questions, while FARMER [52] employs theory of mind to infer unstated goals. Our framework goes beyond query generation or habit memorization by enforcing a closed-loop process of active exploration, multi-turn dialogue, and manipulation, simulating realistic scenarios requiring genuine human-agent cooperation.

3 Environment Design

To facilitate the deployment of vision-driven embodied agents from simulation to the real world, we construct a closed-loop embodied environment (Fig. 2). Built upon the GRUtopia [50] platform, the proposed environment features high-fidelity rendering and rigid-body physics. At the macro level, the design adheres to two core principles that distinguish it from previous environments that simplify exploration and communication: providing a toolchain free of oracle perceptual APIs and capable of real-world implementation, and integrating a simulated user for natural language interaction. Within this architecture, scene entities are categorized into fixed macroscopic `Receptacles` (*e.g.*, tables, cabinets) and manipulable `Objects` (*e.g.*, cups, books). For a specific scene, the `Receptacles` remain static across different task instances, whereas the quantities, categories, and spatial placements of manipulable `Objects` vary.

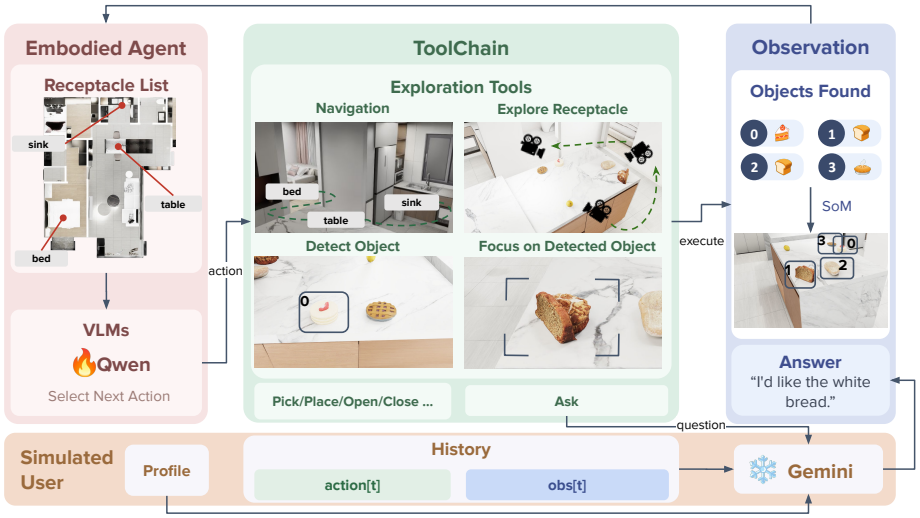

Fig. 2: Overview of the environment design. The Embodied Agent receives physically obtainable receptacle priors and RGB/SoM observations with visual IDs, but no oracle object lists, poses, simulator states, or goal locations. The ToolChain exposes deployable exploration tools (`nav_to`, `walk_around`, `show_object_by_category`, `gaze_at`), `Pick/Place/Open/Close` controls, and `Ask`. The simulated user conditions on the agent's action/observation history and a task profile to answer queries; backend state may be used by handlers but is never exposed to the policy.

3.1 Privilege-Free Environment Modeling

Initial Observation. We assume that the agent is equipped with a long-term 3D occupancy map and semantic segmentation of the receptacles, enabling navigation to a receptacle through path planning. This prior is physically obtainable from a robot's previous scans or mapping stack. Consequently, at task initialization, the agent receives only a prior list of receptacle IDs and their semantic categories (*e.g.*, `table_2`; see the Embodied Agent block in Fig. 2). Despite this structural prior, the agent has no access to global object lists, target 3D poses, simulator object states, or goal locations. It also lacks the fine-grained appearance of receptacles and the distribution of objects upon or within them. To ground target objects, the agent must employ the active exploration toolchain described below to acquire observations.

Multi-level Active Exploration. To enable exploration, we introduce a multi-level active exploration toolchain (see the Exploration Tools block in Fig. 2). These tools avoid oracle perceptual APIs at the policy interface and can be transferred to the real world (details are provided in the supplementary). The overall process is structured into four sequential actions. The inter-receptacle navigation tool (`nav_to`) enables movement between different receptacle nodes based on structural priors. Subsequently, the intra-receptacle scanning tool (`walk_around`)

functions as a broad-spectrum scanner driven by open-vocabulary 2D perception models. This tool navigates the agent around a specific receptacle to bypass physical occlusions and detect common object categories from multiple viewpoints. Following this, the object detection tool (`show_object_by_category`) actively isolates objects of a user-defined category within the current ego-centric view. Ultimately, the target approach and alignment tool (`gaze_at`) advances the agent toward a selected target to center it within the view of the camera, while accounting for physical distance constraints.

Vision-Grounded Interaction. To maintain spatial memory without relying on privileged 3D coordinates, the detected objects are dynamically back-projected from the 2D segmentation masks into the 3D space. This geometric projection constructs a temporary exploration map that records all discovered object categories and their spatial anchors during the most recent invocation of the exploration tool. Objects recorded in this map are highlighted within the RGB/SoM observation of the agent [53] (see the Observation block in Fig. 2). This mechanism bridges perception with physical interaction, enabling the agent to distinguish objects of the same category and parameterize downstream manipulation skills (*e.g.*, `pick`, `open`; see the `Pick/Place/Open/Close` controls in the ToolChain block in Fig. 2) by referencing the assigned visual IDs.

Simulated User for Collaborative Tasks. To simulate collaborative tasks, we integrate a simulated user server driven by `gemini-3-flash` [15] directly into the dynamics of the environment (see the Simulated User block in Fig. 2). At the top level, this module functions as an interactive supervisor that directs the overall task flow. It provides ambiguous initial instructions, monitors the task progress of the agent remotely, and progressively clarifies objectives by answering queries based on real-time exploration results.

To operationalize this top-level design, the module relies on specific mechanisms for configuration, synchronization, and communication. A task-specific profile is initialized for the simulated user to establish the boundaries of preferences and choices, such as restricting targets to objects currently present in the scene. This configuration ensures evaluation stability and prevents the generation of instructions involving non-existent or unexplored items. Furthermore, the server maintains continuous synchronization with the execution trajectory of the embodied agent. By accessing the interaction history and physical observations of the agent, the system constructs a consistent shared context for multi-turn alignment, accurately replicating a human user with limited perception of the initial scene layout. Finally, a bilateral communication interface facilitates interaction. When physical exploration alone fails to resolve ambiguity, the agent can issue an `ask` action to query the user (see the `Ask` control and question flow in Fig. 2). Leveraging the shared context and predefined profile, the simulated user subsequently generates natural language guidance to clarify intent and drive the completion of the task.

4 Agent Training

To minimize online inference overhead and mitigate risks of context explosion or temporal hallucinations during long-horizon deployment, we formulate the decision-making process as a Partially Observable Markov Decision Process (POMDP). Instead of continuously concatenating raw historical observations, our policy relies on a compact aggregated state paired with the current visual input to derive discrete tool actions, maintaining a computationally efficient, localized historical proxy.

Our training pipeline consists of three steps. First, we construct a Markovian state representation and transition pipeline to support efficient data generation (Sec. 4.1). Second, we use expert demonstrations to align the model with the tool interface and structured reasoning format (Sec. 4.2). Finally, we optimize the policy with Group Sequence Policy Optimization (GSPO) [59], using exploration and environmental feedback to improve task-solving performance (Sec. 4.3).

4.1 Problem Formulation and Context Management

Closed-Loop Execution. We formulate the single-step decision-making process of the embodied agent as a POMDP (see the supplementary material for the complete tuple definition). Lacking direct access to the underlying environment state $s_t \in \mathcal{S}$, the agent receives partial observations o_t through an observation function $\Omega(o_t \mid s_t)$ extracting perceivable information from the true state. The agent operates in a continuous perception-reasoning-action loop. Given a task instruction T, the VLM policy maps the current observation o_t to a formatted tool call $a_t \in \mathcal{A}$, using environment tools free of oracle perceptual APIs. A tool executor validates format and dispatches the call, immediately returning invalid actions as environmental feedback. This architecture decouples high-level decision-making from low-level safety verification and environmental dynamics.

Observation and State Aggregation. To address context explosion and temporal hallucinations inherent in long-horizon tasks, we partition the agent's observation into an environmental component and a historical component:

$$o_t = \langle o_t^{env}, o_t^{hist} \rangle \tag{1}$$

The environmental observation o_t^{env} encapsulates immediate perceptual results (*e.g.*, ego-centric visual features, structural world graph changes) and the execution feedback e_t corresponding to a_{t-1}.

To avoid linear context growth from concatenating full interaction histories, we manage o_t^{hist} through self-summarization. We define this compressed observation as:

$$o_t^{hist} = \langle o_{t-1}^{env}, a_{t-1}, h_{t-1} \rangle \tag{2}$$

where h_t represents a structured historical state maintained by the VLM. Initialized by T, the VLM decomposes the global task into sub-goals. At each step,

alongside Chain-of-Thought (CoT) reasoning and tool invocation, the VLM generates a summary σ_t of completed work and an analysis of the active phase $task_t$.

The history is aggregated as $h_t = (\Sigma_t, task_t)$, where $\Sigma_t = [\sigma_0, \sigma_1, \ldots, \sigma_t]$, and is updated online via:

$$h_t \leftarrow \mathrm{VLM}(o_t^{env}, a_t, h_{t-1}) \tag{3}$$

Note that all components of o_t^{hist} (namely o_{t-1}^{env}, a_{t-1}, and h_{t-1}) are fully determined by information available up to step $t-1$, *i.e.*, o_t^{hist} is $\mathcal{F}_{t-1}$-measurable with respect to the natural filtration $\{\mathcal{F}_t\}_{t\geq 0}$. This ensures that the policy $\pi_\theta(a_t \mid o_t)$ conditions only on legitimately available information, preserving the causal structure required by the POMDP formulation.

This bounds the per-step prompt to a fixed structure (current observation, previous action, and compressed history) rather than replaying the interaction trace. Crucially, Σ_t records only the actions confirmed as successful by the tool returns e_t. Tracking execution failures solely in o_t^{env} rather than persistent history provides a deterministic basis for subsequent reasoning and effectively mitigates temporal hallucinations.

Scalable Asset Library and Task Instantiation. Our data engine leverages 7 high-fidelity scenes from GRScenes [50] (11 room types, 100 interactive receptacles) and over 2,500 object instances spanning 639 categories from MesaTask [18]. We represent each scene state as a **World Graph (WG)**, formally $\mathrm{WG} = \{(r_j, \mathcal{E}_j)\}$, mapping each receptacle r_j to its contained object instances $\mathcal{E}_j$. The WG serves as the canonical state representation throughout the framework: it grounds task definitions, enables state-differential reward computation (Sec. 4.3), and provides the success criterion for evaluation.

Tasks center on cross-receptacle object rearrangement: we prompt LLMs to generate semantically plausible *(source, destination, object)* triplets conditioned on the scene layout, which are instantiated as task configurations by specifying an initial WG and a goal WG in the simulator.

To systematically challenge the agent, we introduce distractors at both the object and receptacle levels. Object-level distractors inject visually distinct instances of the same category to require fine-grained discrimination. At the receptacle level, each receptacle is annotated with *uniquely identifying descriptions* covering functional purpose and visual appearance. LLMs then generate unambiguous natural-language instructions from these annotations. Further details and dataset statistics are provided in the supplementary materials.

SFT Data Synthesis. For each task, a rule-based planner decomposes the objective into an action sequence $\{a_t\}_{t=1}^{L} \subset \mathcal{A}$ covering navigation, exploration, and manipulation, and executes it in the simulator. At each step, the observation extractor Ω yields o_t^{env}, including ego-centric RGB images with visual IDs and execution feedback e_t; all manipulation operates exclusively on *visual IDs* from 2D perception. In an automated **Social Annotation** phase for collaborative trajectories, LLMs rewrite instructions to be deliberately vague, injecting communicative actions (`ask`) with simulated user responses into trajectories to train

the agent to communicate actively under uncertainty. The resulting SFT dataset is $\mathcal{D} = \{(T^{(i)}, \tau^{(i)})\}_{i=1}^{N}$, where each trajectory $\tau = \{(o_t, a_t, h_t)\}_{t=1}^{L}$ of length L records the observation, action, and historical state at each step. Actions and observations come from executed trajectories, while historical states are annotated by `gemini-3-pro` [15]. See the supplementary material for annotation details.

4.2 Paradigm Alignment via SFT

Unified Input-Output Paradigm. We select Qwen3-VL-8B as the backbone VLM to support deployment on computationally constrained robotic systems, implementing a unified input-output structure for both training and inference.

At each time step t, the VLM receives a prompt containing the available tool APIs $\mathcal{A}$, the task instruction T, and the observation $o_t = \langle o_t^{env}, o_t^{hist} \rangle$. The model generates internal reasoning (e.g., `<think>...</think>`) to analyze the visual state and plan, then outputs a JSON specifying the updated historical state h_t and next tool call $a_t \in \mathcal{A}$.

Tool-Calling Alignment via Behavior Cloning. In the first stage, we adapt the pre-trained VLM to the tool interface by behavior cloning on $\mathcal{D}$. Let π_θ denote the VLM policy. We minimize the negative log-likelihood of expert trajectories:

$$\mathcal{L}_{\text{SFT}} = -\sum_{(T,\tau)\in\mathcal{D}} \sum_{t=1}^{L} \log \pi_\theta(a_t, h_t \mid T, o_t)\,, \tag{4}$$

where o_t is the agent's observation at step t. We freeze the vision encoder to preserve pre-trained visual representations.

4.3 Reinforcement Learning for Skill Optimization

Deploying the SFT-aligned initial policy π_{init} (Sec. 4.2) in closed-loop inference exposes two limitations: (1) *distribution shift*, where behavior cloning on expert demonstrations does not equip the model to replan or recover from execution failures (*e.g.*, occluded targets, invalid API calls); and (2) *scarce interactive data*, where static datasets cannot cover the diversity of proactive human-robot interaction, causing the policy to overfit to templates rather than querying the user under ambiguity. RL addresses both issues by enabling the agent to explore dynamic environments and actively acquire information from users.

Algorithm Selection. We aim to maximize the expected discounted return over variable-length trajectories under the POMDP formulation (Sec. 4.1). Accurately modeling a critic for VLM-based RL is computationally prohibitive, making critic-free methods such as Group Relative Policy Optimization (GRPO) a natural choice. However, GRPO assigns sequence-level credit via step-wise importance ratios whose cumulative log-ratios grow in variance with trajectory length, frequently triggering clipping and destabilizing training. We therefore

adopt GSPO, which replaces the step-wise ratios with a length-normalized sequence ratio, compressing variance by $1/L_i^2$ and keeping proximal updates valid during long rollouts. This drives the model toward adaptive replanning and proactive interaction rather than static template matching. Detailed formulations are provided in the supplementary material.

Environment-Grounded Reward Shaping. Instead of sparse task-completion signals, we design a step-wise reward r_t grounded in the WG (Sec. 4.1). We compute Δ_{task}, the set of required state changes between the initial and goal WG (*e.g.*, picking and placing n objects yields $2n$ entries). At each step, the instantaneous difference $\delta_t = \mathrm{WG}_t - \mathrm{WG}_{t-1}$ is matched against Δ_{task}; a match yields a positive reward and the item is removed to prevent reward hacking. The physics engine inherently enforces temporal causality (*e.g.*, pick-before-place), eliminating the need to manually model temporal dependencies. Articulation and navigation sub-goals are appended to Δ_{task} and evaluated online.

Beyond task progress, the agent receives a per-step timing penalty $-\lambda_{time}$ for efficiency and a severity-scaled penalty for failed actions based on e_t traceback logs. For tasks involving distractors, we reward the communicative tool `ask` to resolve ambiguity, subject to a difficulty-conditioned query budget $B_{\texttt{ask}}$ to prevent over-reliance on the user. By formulating the reward as $r_t = R(o_{t-1}^{env}, a_t, o_t^{env})$, the Markov property is preserved and the dense signals align with the GSPO optimization objective (detailed numerical configurations are provided in the supplementary material).

5 Experiments

5.1 Benchmark Construction

To evaluate embodied agents along two key dimensions, *active exploration* and *communicative interaction*, we introduce **REAL-Bench**. Built on the environment and tool interface (Sec. 3) and the data generation pipeline (Sec. 4.1), it covers novel scenarios across REAL rooms. REAL-Bench contains **241** task instances in four families: Furniture-Distractor Pick-and-Place (FDP, 72), Furniture & Object-Distractor Pick-and-Place (FODP, 56), Furniture-Distractor Open/Close (FDO, 48), and Simulator-User-Loop (SUL, 65).

FDP and FODP are cluttered mobile manipulation tasks that require moving a target object from a source to a destination receptacle. FODP further adopts an open-vocabulary, full-category distractor setting at both the furniture and object levels, placing greater demands on active exploration and fine-grained visual disambiguation. FDO evaluates articulation-centric manipulation with openable and closeable furniture. These tasks involve highly dynamic state transitions and challenge temporal consistency and precise sequential tool use (*e.g.*, interleaving `open`/`close` with perception and pick-and-place actions). Finally, SUL introduces instruction ambiguity by design to test whether the agent can proactively query the simulated user via `ask` before execution.

5.2 Results and Analysis of SFT and RL Training

We evaluate (i) strong open-source and commercial VLMs under zero-shot prompting, and (ii) the proposed Qwen3-VL-8B agent after *tool-interface alignment* via SFT and *closed-loop* improvement via GSPO. Table 1 reports *success rates* (higher is better) on the four REAL-Bench task families.

Table 1: Main results on REAL-Bench. Success rates (successful / total episodes) are reported across four task families. **Bold** marks the best result per column within each group, and the highlighted row denotes the teacher model used for SFT annotation. SFT aligns Qwen3-VL-8B to the tool interface free of oracle perceptual APIs, while GSPO further improves the interaction-heavy SUL split and surpasses all zero-shot VLM baselines, including the teacher.

Model Name	FDP	FODP	FDO	SUL
Zero-shot Prompting				
gemini-3-pro-preview [16]	**81.9**	39.3	50.0	53.8
gemini-2.5-pro [10]	66.7	**41.1**	43.8	49.2
gpt-5 [33]	68.1	30.4	47.9	52.3
gpt-4o [32]	30.6	28.6	27.1	40.0
claude-haiku-4-5 [2]	45.8	16.1	**52.1**	47.7
Qwen3-VL-235B-A22B-Instruct [3]	22.2	14.3	14.6	18.5
Qwen3-VL-8B-Instruct [3]	0.0	0.0	0.0	1.5
Ours				
Ours-8B-SFT-only (1 epoch)	45.8	30.4	35.4	36.9
Ours-8B-SFT-only (2 epoch)	65.3	28.6	43.8	50.8
Ours-8B-SFT+RL	58.3	33.9	41.7	**56.9**

Tool Alignment and Behavior-Cloning Overfitting Zero-shot Qwen3-VL-8B fails on manipulation tasks and achieves only 1.5% on interactive scenarios. One epoch of supervised fine-tuning aligns the model with the tool interface, reaching 45.8% on FDP. However, a second epoch exposes the limits of behavior cloning: while execution stability on standard manipulation improves (FDP rises to 65.3%), performance on the distractor-rich FODP split drops from 30.4% to 28.6%. This suggests that strict imitation overfits expert trajectories and weakens robustness under open-vocabulary clutter and visual ambiguity.

Reinforcement Learning Restores Generalization Applying closed-loop group sequence policy optimization to the single-epoch checkpoint circumvents this overfitting. RL recovers the exploration capability lost during extended supervised training, increasing FODP success to 33.9% while maintaining tool-calling performance comparable to the model more tightly fine-tuned on standard tasks. This suggests that environmental feedback promotes adaptive problem-solving over rote trajectory memorization, helping the agent handle edge cases.

Social Reward Improves Proactive Intent Alignment The impact of closed-loop optimization is most pronounced in the interaction-heavy SUL split, where the agent reaches 56.9% success and outperforms the strongest proprietary zero-shot baselines. By integrating a difficulty-conditioned query budget and explicit rewards for resolving ambiguity, the training pipeline shifts the agent from passive instruction following to proactive intent alignment. The policy learns when to pause and query the simulator-user, showing that social interaction emerges from end-to-end environmental feedback rather than static prompt engineering.

Dynamic Manipulation Reveals a Vision Bottleneck Despite the gains from policy optimization, success rates consistently drop in the articulation-heavy FDO split. Operating doors and drawers introduces highly dynamic visual state changes. Because we freeze the vision encoder to preserve pre-trained representations, the agent remains bottlenecked by the base model's visual grounding limits. This suggests that while closed-loop RL improves action and communication, mastering articulated manipulation may require further scaling on its multimodal abilities.

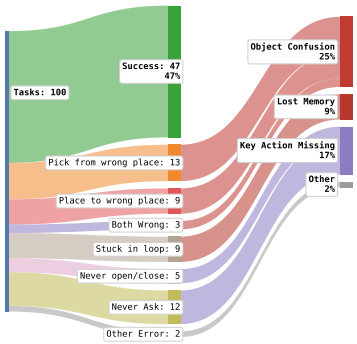

Fig. 3: Failure mode taxonomy. Manual analysis over 100 evaluation episodes from the best SFT+RL checkpoint.

Failure Mode Taxonomy We manually annotate 100 episodes from the best SFT+RL checkpoint to diagnose residual errors (Fig. 3). Of 53 failures, most fall under *Object Confusion* (25), followed by *Key Action Missing* (17), *Lost Memory* (9), and *Other Error* (2). These results identify cluttered visual grounding, procedural completeness, and long-horizon memory as the main remaining bottlenecks.

5.3 Real-World Deployment via Sim-to-Real Transfer

To validate the effectiveness and robustness of our training pipeline, we deploy the simulated agent in a real-world household scenario using the Ark LIFT2 mobile manipulator. This direct sim-to-real transfer tests how capabilities learned in simulation map to physical execution across varying environmental complexity.

Seamless Transfer Through Standardized Interfaces The system uses a Model Context Protocol(MCP) server for tool invocation, keeping the high-level cognitive framework identical between simulation and the physical robot. At each step, agent receives real-time ego-centric observations, combines them with the structured history and immediate past action, and outputs the next discrete tool

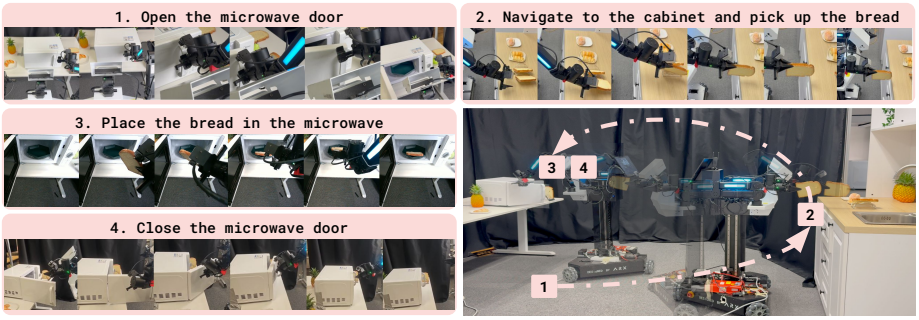

Fig. 4: Real-world deployment on the Ark LIFT2 mobile manipulator. REAL transfers the high-level visual-interactive policy zero-shot to a physical backend with the same MCP tool interface. The execution trajectory demonstrates causal planning (*e.g.*, preparing the microwave before fetching food), human-robot disambiguation under multiple candidate objects, and language-to-physical grounding for closed-loop task completion. Quantitative reliability is reported in Tab. 2.

call. This standardized interface decouples the reasoning algorithm from the environment, allowing the physical world to act as an alternative backend to the simulator without retraining the high-level policy.

Hierarchical Execution with Specialized Physical Policies While high-level planning is transferred zero-shot from simulation, continuous physical dynamics still require robust motor control. We therefore map the abstract tools used in simulation to dedicated real-world implementations. Specifically, atomic manipulation commands are executed by a fine-tuned $\pi_{0.5}$ [37] base model. This hierarchy keeps VLM focused on visual grounding, state tracking, and task planning, while the $\pi_{0.5}$ policy provides stable execution for opening, closing, picking, and placing.

Long-Horizon Causal Reasoning in Physical Environments As shown in Fig. 4, the system successfully executes a multi-stage food preparation task requiring the robot to fetch bread and place it in a microwave. Rather than immediately navigating to the cabinet, the reasoning engine first approaches the table and opens the microwave door. Only after preparing the receptacle does the robot navigate to the secondary location, select the target object, and return to complete placement and closure. This sequence suggests that the structured memory mechanism helps prevent temporal hallucinations, maintain an accurate internal state, and respect causal constraints over long physical executions.

Physical Evaluation and Pipeline Analysis We further evaluate the pipeline in the physical environment across different spatial layouts. Across 60 real-world episodes, REAL achieves 78.3% end-to-end success with zero unrecoverable system crashes (Tab. 2). The low-level primitive layer is executable in 85.3% of 600 VLA primitive executions, indicating that the high-level policy transfers without requiring privileged simulator state while still depending on robust physical

Table 2: End-to-end real-world evaluation. SR denotes task success rate, SPL measures path efficiency, Ask is the average number of user queries, and VLM Lat. reports total VLM inference latency per episode.

Task	SR	Steps	SPL	Ask	VLM Lat.
FDO	80.0%	12.2	67.0%	0.90	71.1s
FODP	100.0%	9.6	86.9%	0.70	53.5s
SUL	55.0%	13.0	35.4%	0.55	80.5s
Overall	**78.3%**	**11.6**	**63.1%**	**0.72**	**68.4s**

skills. A more detailed breakdown of these real-world trials, along with primitive-level evaluation and failure analysis, is provided in the supplementary material.

6 Conclusion and Limitations

We presented REAL, a closed-loop framework for training exploratory and communicative embodied agents that transfer directly to physical robots. By replacing oracle perceptual APIs with a physically realizable exploration toolchain and integrating an LLM-driven simulated user for dynamic intent alignment, REAL addresses two key oversimplifications in prior work. Training a Qwen3-VL-8B agent via SFT followed by closed-loop GSPO on REAL-Bench (241 tasks, four families) shows that tool-interface alignment is a prerequisite for closing the perception-action loop, while RL uniquely improves interactive disambiguation. Deployment on the Ark LIFT2 dual-arm mobile robot further validates sim-to-real transferability for long-horizon household tasks.

Despite these results, several limitations remain. First, the task space centers on cross-receptacle rearrangement and does not yet encode richer compositional constraints such as temporally ordered sub-tasks [5] or spatial relational goals [11] (*e.g.*, "place the cup *to the left of* the plate"). Second, the simulated user operates within a bounded behavioral envelope—its preferences are constrained to in-scene objects and its responses are anchored to the agent's trajectory—whereas real users may introduce new goals mid-task or convey intent through implicit cues. Third, receptacles are modeled as monolithic entities (*e.g.*, `cabinet_1`) without part-level distinction (*e.g.*, individual shelves), limiting spatial precision for both exploration and manipulation. Incorporating part-level segmentation and hierarchical receptacle modeling would address this gap.

7 Acknowledgments

This work was supported in part by Shanghai Artificial Intelligence Laboratory, the National Natural Science Foundation of China under Grant 62401367 and the Shanghai Magnolia Talent Program Pujiang Project under Grant No. 25PJA076.

References

1. Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Ho, D., Hsu, J., Ibarz, J., Ichter, B., Irpan, A., Jang, E., Ruano, R.J., Jeffrey, K., Jesmonth, S., Joshi, N.J., Julian, R., Kalashnikov, D., Kuang, Y., Lee, K.H., Levine, S., Lu, Y., Luu, L., Parada, C., Pastor, P., Quiambao, J., Rao, K., Rettinghouse, J., Reyes, D., Sermanet, P., Sievers, N., Tan, C., Toshev, A., Vanhoucke, V., Xia, F., Xiao, T., Xu, P., Xu, S., Yan, M., Zeng, A.: Do as i can, not as i say: Grounding language in robotic affordances. In: Proceedings of The 6th Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 205, pp. 287–318. PMLR (2023)
2. Anthropic: Claude haiku 4.5 system card. System card, Anthropic (Oct 2025), `https://www-cdn.anthropic.com/7aad69bf12627d42234e01ee7c36305dc2f6a970.pdf`, accessed: July 1, 2026
3. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-VL technical report. arXiv preprint arXiv:2511.21631 (2025)
4. Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Dabis, J., Finn, C., Gopalakrishnan, K., Hausman, K., Herzog, A., Hsu, J., et al.: RT-1: Robotics transformer for real-world control at scale. In: Robotics: Science and Systems (2023)
5. Chang, M., Chhablani, G., Clegg, A., Cote, M.D., Desai, R., Hlavac, M., Karashchuk, V., Krantz, J., Mottaghi, R., Parashar, P., Patki, S., Prasad, I., Puig, X., Rai, A., Ramrakhya, R., Tran, D., Truong, J., Turner, J.M., Undersander, E., Yang, T.Y.: PARTNR: A benchmark for planning and reasoning in embodied multi-agent tasks. In: International Conference on Learning Representations (2025)
6. Chen, H., Zhao, M., Yang, R., Ma, Q., Yang, K., Yao, J., Wang, K., Bai, H., Wang, Z., Pan, R., Zhang, M., Barreiros, J., Onol, A., Zhai, C., Ji, H., Li, M., Zhang, H., Zhang, T.: Era: Transforming vlms into embodied agents via embodied prior learning and online reinforcement learning. arXiv preprint arXiv:2510.12693 (2025)
7. Chen, J., Liang, H., Du, L., Wang, W., Hu, M., Mu, Y., Wang, W., Dai, J., Luo, P., Shao, W., Shao, L.: OWMM-Agent: Open world mobile manipulation with multi-modal agentic data synthesis. In: Advances in Neural Information Processing Systems (2025)
8. Chen, J., Yu, C., Zhou, X., Xu, T., Mu, Y., Hu, M., Shao, W., Wang, Y., Li, G., Shao, L.: EMOS: Embodiment-aware heterogeneous multi-robot operating system with LLM agents. In: ICLR (2025)
9. Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R., Song, S.: Diffusion policy: Visuomotor policy learning via action diffusion. The International Journal of Robotics Research p. 02783649241273668 (2024). `https://doi.org/10.1177/02783649241273668`
10. Comanici, G., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
11. Dai, L., Wang, H., Wan, W., Su, H.: ManiTaskGen: A comprehensive task generator for benchmarking and improving vision-language agents on embodied decision-making. arXiv preprint arXiv:2505.20726 (2025)

12. Deitke, M., VanderBilt, E., Herrasti, A., Weihs, L., Ehsani, K., Salvador, J., Han, W., Kolve, E., Kembhavi, A., Mottaghi, R.: ProcTHOR: Large-scale embodied AI using procedural generation. In: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (eds.) Advances in Neural Information Processing Systems. vol. 35, pp. 5982–5994. Curran Associates, Inc. (2022)
13. Driess, D., Xia, F., Sajjadi, M.S.M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al.: PaLM-E: An embodied multimodal language model. In: International Conference on Machine Learning. pp. 8469–8488 (2023)
14. Garrett, C.R., Chitnis, R., Holladay, R., Kim, B., Silver, T., Kaelbling, L.P., Lozano-Pérez, T.: Integrated task and motion planning. Annual Review of Control, Robotics, and Autonomous Systems **4**(1), 265–293 (2021). https://doi.org/10.1146/annurev-control-091420-084139
15. Google DeepMind: Gemini. https://deepmind.google/models/gemini/ (2025), accessed: June 26, 2026
16. Google DeepMind: Gemini 3 pro model card. Model card, Google DeepMind (2026), https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-Pro-Model-Card.pdf, model release: November 2025; last updated: May 2026. Used to document the Gemini 3 Pro family corresponding to the evaluated gemini-3-pro-preview API identifier. Accessed: July 1, 2026
17. Gu, Q., Kuwajerwala, A., Morin, S., Jatavallabhula, K.M., Sen, B., Agarwal, A., Rivera, C., Paul, W., Ellis, K., Chellappa, R., et al.: ConceptGraphs: Open-vocabulary 3D scene graphs for perception and planning. In: IEEE International Conference on Robotics and Automation. pp. 5021–5028 (2024)
18. Hao, J., Liang, N., Luo, Z., Xu, X., Zhong, W., Yi, R., Jin, Y., Lyu, Z., Zheng, F., Ma, L., Pang, J.: MesaTask: Towards task-driven tabletop scene generation via 3D spatial reasoning. In: Advances in Neural Information Processing Systems (2025)
19. Huang, W., Abbeel, P., Pathak, D., Mordatch, I.: Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In: International Conference on Machine Learning. pp. 9118–9147. PMLR (2022)
20. Huang, W., Xia, F., Xiao, T., Chan, H., Liang, J., Florence, P., Zeng, A., Tompson, J., Mordatch, I., Chebotar, Y., et al.: Inner monologue: Embodied reasoning through planning with language models. In: Conference on Robot Learning. pp. 1769–1782. PMLR (2023)
21. Huang, W., Zhu, S., Wei, M., Xu, J., Liu, X., Wang, H., Wang, T., Zhao, F., Pang, J.: VL-LN bench: Towards long-horizon goal-oriented navigation with active dialogs. arXiv preprint arXiv:2512.22342 (2025)
22. Kaelbling, L.P., Lozano-Pérez, T.: Hierarchical task and motion planning in the now. In: 2011 IEEE International Conference on Robotics and Automation. pp. 1470–1477. IEEE (2011). https://doi.org/10.1109/ICRA.2011.5980391
23. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., Vuong, Q., Kollar, T., Burchfiel, B., Tedrake, R., Sadigh, D., Levine, S., Liang, P., Finn, C.: OpenVLA: An open-source vision-language-action model. In: Conference on Robot Learning (2024)
24. Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Deitke, M., Ehsani, K., Gordon, D., Zhu, Y., Kembhavi, A., Gupta, A., Farhadi, A.: AI2-THOR: An interactive 3d environment for visual ai. arXiv preprint arXiv:1712.05474 (2017)
25. Lei, Z., Yin, S., Xiong, Y., Ding, Y., Huang, W., Wei, Y., Xu, Q., Li, Y., Li, W., Wang, Y., Chen, S.: EmboMatrix: A scalable training-ground for embodied decision-making. arXiv preprint arXiv:2510.12072 (2025)

26. Li, C., Zhang, R., Wong, J., Gokmen, C., Srivastava, S., Martín-Martín, R., Wang, C., Levine, G., Lingelbach, M., Sun, J., et al.: Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In: Conference on Robot Learning. pp. 80–93. PMLR (2023)
27. Li, D., Cai, T., Tang, T., Chai, W., Driggs-Campbell, K.R., Wang, G.: EMMOE: A comprehensive benchmark for embodied mobile manipulation in open environments. arXiv preprint arXiv:2503.08604 (2025)
28. Li, M., Zhao, S., Wang, Q., Wang, K., Zhou, Y., Srivastava, S., Gokmen, C., Lee, T., Li, L.E., Zhang, R., Liu, W., Liang, P., Fei-Fei, L., Mao, J., Wu, J.: Embodied agent interface: Benchmarking LLMs for embodied decision making. In: Advances in Neural Information Processing Systems (2024)
29. Liang, J., Huang, W., Xia, F., Xu, P., Hausman, K., Ichter, B., Florence, P., Zeng, A.: Code as policies: Language model programs for embodied control. In: IEEE International Conference on Robotics and Automation. pp. 9493–9500 (2023)
30. Liu, X., Zhang, T., Gu, Y., Iong, I.L., XiXuan, S., Xu, Y., Zhang, S., Lai, H., Sun, J., Yang, X., Yang, Y., Qi, Z., Yao, S., Sun, X., Cheng, S., Zheng, Q., Yu, H., Zhang, H., Hong, W., Ding, M., Pan, L., Gu, X., Zeng, A., Du, Z., Song, C.H., Su, Y., Dong, Y., Tang, J.: Visualagentbench: Towards large multimodal models as visual foundation agents. In: The Thirteenth International Conference on Learning Representations (2025)
31. Nasiriany, S., Xia, F., Yu, W., Xiao, T., Liang, J., Dasgupta, I., Xie, A., Driess, D., Wahid, A., Xu, Z., Vuong, Q., Zhang, T., Lee, T.W.E., Lee, K.H., Xu, P., Kirmani, S., Zhu, Y., Zeng, A., Hausman, K., Heess, N., Finn, C., Levine, S., Ichter, B.: PIVOT: Iterative visual prompting elicits actionable knowledge for VLMs. In: International Conference on Machine Learning (2024)
32. OpenAI: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024)
33. OpenAI: Openai gpt-5 system card. arXiv preprint arXiv:2601.03267 (2026)
34. Padmakumar, A., Thomason, J., Shrivastava, A., Lange, P., Narayan-Chen, A., Gella, S., Piramuthu, R., Tür, G., Hakkani-Tür, D.: TEACh: Task-driven embodied agents that chat. In: AAAI. pp. 2017–2025 (2022). https://doi.org/10.1609/AAAI.V36I2.20097
35. Patel, M., Puig, X., Desai, R., Mottaghi, R., Chernova, S., Truong, J., Rai, A.: ADAPT: Actively discovering and adapting to preferences for any task. In: Conference on Language Modeling (2025)
36. Philipov, D., Dongre, V., Tür, G., Hakkani-Tür, D.: Simulating user agents for embodied conversational-ai. arXiv preprint arXiv:2410.23535 (2024)
37. Physical Intelligence, Black, K., Brown, N., Darpinian, J., Dhabalia, K., Driess, D., Esmail, A., Equi, M., Finn, C., Fusai, N., et al.: $\pi_{0.5}$: a vision-language-action model with open-world generalization. arXiv preprint arXiv:2504.16054 (2025)
38. Puig, X., Ra, K., Boben, M., Li, F., Wang, T., Fidler, S., Torralba, A.: VirtualHome: Simulating household activities via programs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8494–8502 (2018)
39. Qwen, Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., Lin, H., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Lin, J., Dang, K., Lu, K., et al.: Qwen2.5 technical report. arXiv preprint arXiv:2412.15115 (2024)
40. Ren, A.Z., Clark, J., Dixit, A., Itkina, M., Majumdar, A., Sadigh, D.: Explore until confident: Efficient exploration for embodied question answering. In: Robotics: Science and Systems (2024)

41. Ren, A.Z., Dixit, A., Bodrova, A., Singh, S., Tu, S., Brown, N., Xu, P., Takayama, L., Xia, F., Varley, J., Xu, Z., Sadigh, D., Zeng, A., Majumdar, A.: Robots that ask for help: Uncertainty alignment for large language model planners. In: Conference on Robot Learning (2023)
42. Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., Fox, D.: 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)
43. Shridhar, M., Yuan, X., Côté, M.A., Bisk, Y., Trischler, A., Hausknecht, M.: ALFWorld: Aligning text and embodied environments for interactive learning. In: International Conference on Learning Representations (2021)
44. Skreta, M., Zhou, Z., Yuan, J.L., Darvish, K., Aspuru-Guzik, A., Garg, A.: Replan: Robotic replanning with perception and language models. arXiv preprint arXiv:2401.04157 (2024)
45. Szot, A., Clegg, A., Undersander, E., Wijmans, E., Zhao, Y., Turner, J., Maestre, N., Mukadam, M., Chaplot, D., Maksymets, O., Gokaslan, A., Vondrus, V., Dharur, S., Meier, F., Galuba, W., Chang, A., Kira, Z., Koltun, V., Malik, J., Savva, M., Batra, D.: Habitat 2.0: Training home assistants to rearrange their habitat. In: Advances in Neural Information Processing Systems (2021)
46. Szot, A., Mazoure, B., Attia, O., Timofeev, A., Agrawal, H., Hjelm, D., Gan, Z., Kira, Z., Toshev, A.: From multimodal llms to generalist embodied agents: Methods and lessons. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 10644–10655 (2025)
47. Szot, A., Schwarzer, M., Agrawal, H., Mazoure, B., Metcalf, R., Talbott, W., Mackraz, N., Hjelm, R.D., Toshev, A.: Large language models as generalizable policies for embodied tasks. In: International Conference on Learning Representations (2024)
48. Tan, H., Hao, X., Chi, C., Lin, M., Lyu, Y., Cao, M., Liang, D., Chen, Z., Lyu, M., Peng, C., He, C., Ao, Y., Lin, Y., Wang, P., Wang, Z., Zhang, S.: RoboOS: A hierarchical embodied framework for cross-embodiment and multi-agent collaboration. arXiv preprint arXiv:2505.03673 (2025). https://doi.org/10.48550/arXiv.2505.03673
49. Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., Anandkumar, A.: Voyager: An open-ended embodied agent with large language models. Transactions on Machine Learning Research (2024)
50. Wang, H., Chen, J., Huang, W., Ben, Q., Wang, T., Mi, B., Huang, T., Zhao, S., Chen, Y., Yang, S., Cao, P., Yu, W., Ye, Z., Li, J., Long, J., Wang, Z., Wang, H., Zhao, Y., Tu, Z., Qiao, Y., Lin, D., Pang, J.: GRUtopia: Dream general robots in a city at scale. arXiv preprint arXiv:2407.10943 (2024)
51. Wang, S., Fei, Z., Cheng, Q., Zhang, S., Cai, P., Fu, J., Qiu, X.: World modeling makes a better planner: Dual preference optimization for embodied task planning. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 21518–21537. Vienna, Austria (2025). https://doi.org/10.18653/v1/2025.acl-long.1044
52. Wang, Y., Huang, X., Zhong, F., Yang, Y., Wang, Y., Chen, Y., Dong, H.: Communication-efficient desire alignment for embodied agent-human adaptation. arXiv preprint arXiv:2505.22503 (2025)
53. Yang, J., Zhang, H., Li, F., Zou, X., Li, C., Gao, J.: Set-of-mark prompting unleashes extraordinary visual grounding in GPT-4V. arXiv preprint arXiv:2310.11441 (2023)

54. Yang, R., Chen, H., Zhang, J., Zhao, M., Qian, C., Wang, K., Wang, Q., Koripella, T.V., Movahedi, M., Li, M., Ji, H., Zhang, H., Zhang, T.: EmbodiedBench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. In: International Conference on Machine Learning (2025)
55. Yang, Y., Sun, F.Y., Weihs, L., VanderBilt, E., Fox, J.S., Kembhavi, A., Mottaghi, R.: Holodeck: Language guided generation of 3D embodied AI environments. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2024)
56. Yuan, H., Bai, Y., Fu, Y., Zhou, B., Feng, Y., Xu, X., Zhan, Y., Karlsson, B.F., Lu, Z.: Being-0: A humanoid robotic agent with vision-language models and modular skills. arXiv preprint arXiv:2503.12533 (2025)
57. Yuan, P., Ma, A., Yao, Y., Yao, H., Tomizuka, M., Ding, M.: REMAC: Self-reflective and self-evolving multi-agent collaboration for long-horizon robot manipulation. arXiv preprint arXiv:2503.22122 (2025)
58. Zhang, H., Du, W., Shan, J., Zhou, Q., Du, Y., Tenenbaum, J.B., Shu, T., Gan, C.: Building cooperative embodied agents modularly with large language models. In: International Conference on Learning Representations (2024)
59. Zheng, C., Liu, S., Li, M., Chen, X.H., Yu, B., Gao, C., Dang, K., Liu, Y., Men, R., Yang, A., Zhou, J., Lin, J.: Group sequence policy optimization. arXiv preprint arXiv:2507.18071 (2025)
60. Zitkovich, B., Yu, T., Xu, S., Xu, P., Xiao, T., Xia, F., Wu, J., Wohlhart, P., Welker, S., Wahid, A., Vuong, Q., Vanhoucke, V., Tran, H., Soricut, R., Singh, A., Singh, J., Sermanet, P., Sanketi, P.R., Salazar, G., Ryoo, M.S., Reymann, K., Rao, K., Pertsch, K., Mordatch, I., Michalewski, H., Lu, Y., Levine, S., Lee, L., Lee, T.W.E., Leal, I., Kuang, Y., Kalashnikov, D., Julian, R., Joshi, N.J., Irpan, A., Ichter, B., Hsu, J., Herzog, A., Hausman, K., Gopalakrishnan, K., Fu, C., Florence, P., Finn, C., Dubey, K.A., Driess, D., Ding, T., Choromanski, K.M., Chen, X., Chebotar, Y., Carbajal, J., Brown, N., Brohan, A., Arenas, M.G., Han, K.: RT-2: Vision-language-action models transfer web knowledge to robotic control. In: Proceedings of The 7th Conference on Robot Learning. Proceedings of Machine Learning Research, vol. 229, pp. 2165–2183. PMLR (2023)