TAIROS: An Embodied AI Platform for Robotics Applications

Tencent Robotics X Team & Futian Laboratory, Shenzhen

https://tairos.tencent.com

Abstract

Recent advancements in embodied intelligence and robotics have witnessed groundbreaking innovations across hardware and AI model architectures. While significant progress has been made in specialized foundation models for reasoning, multi-modal perception, manipulation and locomotion, there remains a critical gap in unified platforms capable of seamless cross-embodiment deployment for real-world robot applications. We present TAIROS, a comprehensive embodied AI platform that integrates multi-modal perception, long-horizon planning, and dexterous action capabilities into a unified modular architecture. Building upon state-of-the-art LLM, VLM, and VLA models, TAIROS features three interoperable modules: Embodied Perception, Embodied Planning, and Perception-Action, designed for both integrated agent deployment and standalone functionality. Our platform demonstrates exceptional generalization across diverse robotic embodiments (humanoids, quadrupeds, bi-manual manipulators) and real-world tasks including complex manipulation, dynamic locomotion, and multi-modal interaction. Extensive validation on industrial and domestic scenarios confirms TAIROS's capabilities in bridging the gap between AI advancements and physical-world applications.

1. Introduction

The advent of foundation models has ushered in a new paradigm for artificial intelligence systems, with transformative impacts across vision, language, and decision-making domains. These models, trained in Internet-scale datasets that encompass trillions of tokens and millions of images, have demonstrated unprecedented generalization and adaptation capabilities. Seminal works like GPT-4 [1] and Gemini [2] have shown how large-scale pretraining can yield models that transfer effectively to downstream tasks with minimal fine-tuning. Particularly in embodied intelligence and robotics, foundation models offer the promise of overcoming longstanding challenges in generalization, sample efficiency, and multi-modal understanding that have constrained traditional approaches.

Embodied intelligence represents an interdisciplinary field that integrates mechanical engineering, embodiment design, control theory, and AI. The rapid advancement of foundation models in AI has recently led to the emergence of numerous specialized models addressing

Email address: roboticsx@tencent.com (Tencent Robotics X Team & Futian Laboratory, Shenzhen)

different aspects of embodied intelligence, which can be broadly categorized into four types: multi-modal foundation models for embodied perception and navigation, large language models for embodied reasoning, vision-language-action (VLA) models for manipulation, and simulation-based reinforcement learning for locomotion and whole-body control (WBC).

1.1. Multi-modal Foundation Models

In recent years, multi-modal foundation models have made significant advances, driven by breakthroughs in cross-modal semantic understanding. The introduction of CLIP [3] marked a milestone by mapping images and text into a shared embedding space through contrastive learning, laying the groundwork for unified multi-modal representations. Building on this foundation, OpenAI launched DALL·E [4], which pioneered the use of diffusion models for text-to-image generation and opened a new chapter in generative multi-modal modeling. Concurrently, Google introduced the Vision Transformer (ViT) [5], which brought the Transformer architecture to the visual domain, replacing traditional CNNs and providing a unified backbone for multi-modal integration. This architectural innovation paved the way for more scalable and flexible multi-modal models. Subsequently, Google released PaLM-E [6], a large-scale model that integrates text, images, and robotic sensor data, scaling up to 562 billion parameters. PaLM-E represents a significant step toward embodied intelligence by enabling a closed loop from perception to action within a single model.

In the field of perception, foundation models have also achieved remarkable progress. Meta's Segment Anything Model (SAM) [7] was the first general-purpose image segmentation foundation model, demonstrating strong zero-shot generalization and enhancing object segmentation in areas such as autonomous driving and robotics. Its successor, SAM2 [8], further improved efficiency and accuracy, enabling object segmentation from video streams and showing great potential in real-world robotic perception. However, both SAM and SAM2 lack comprehensive scene-level semantic understanding, which limits their application in more complex tasks and necessitates integration with visual classification models. To address the need for open-vocabulary and text-guided object detection, models such as Grounding DINO [9] and YOLO-World [10] have emerged. Grounding DINO leverages a multi-modal transformer architecture to achieve deep cross-modal fusion, enabling zero-shot object localization based on textual descriptions without additional training. In contrast, YOLO-World extends the traditional YOLO framework with vision-language pretraining, focusing on real-time, open-vocabulary object detection for practical deployment.

As detection and segmentation models evolve, a new trend has emerged: directly integrating visual information into large language models to create multi-modal foundation models. This integration enables richer visual-language interaction and reasoning, pushing VLMs toward general artificial intelligence and embodied intelligence applications. For example, OpenAI's GPT-4o [1] extends language models with visual input capabilities, supporting complex reasoning and generation that combines images and text. Similarly, Qwen2.5-VL [11] emphasizes comprehensive vision-language understanding across images, videos, text, and structured layouts, while VLN-Game [12] combines pretrained vision-language features with 3D mapping and game-theoretic target matching to enable zero-shot visual-language navigation.

Building on these advances, recent research has begun to explore the application of multi-modal foundation models in embodied intelligence scenarios. For instance, ConceptGraphs [13] proposes an efficient open-vocabulary 3D scene graph representation that optimizes storage and scalability by focusing features on object nodes. Werby et al. [14] further developed a hierarchical open-vocabulary 3D scene graph approach, enabling robots to understand objects and their spatial relationships in complex environments and to follow natural language instructions more effectively.

Within the TAIROS platform, our Embodied Perception Module builds on these state-of-the-art multi-modal foundation models to deliver enhanced visual-language understanding and memory. This unified framework bridges perception and action, enabling more sophisticated scene interpretation and task execution across a wide range of robotic applications.

1.2. Embodied Reasoning using LLMs

In the domain of embodied reasoning using large language models (LLMs), current research has evolved along several distinct yet complementary technical pathways. The first category adopts hierarchical architectures that decouple high-level planning from low-level execution, exemplified by frameworks like DEDER [15] which distills complex reasoning from LLMs into smaller, resource-efficient models through a two-tier policy structure and embodied knowledge graph. Similarly, Environment Preference Optimization (EPO) [16] introduces a novel hierarchical framework that decomposes long-horizon tasks into sub-goals while leveraging multi-modal environment feedback to generate automated training signals, achieving state-of-the-art performance on established benchmarks like ALFRED. Another notable approach, EmbodiedAgent [17], addresses multi-robot coordination challenges through a structured memory system that validates actions against environmental constraints, supported by the MultiPlan+ dataset and RPAS assessment schema.

Many another approaches focus on enhancing multi-modal understanding through tighter vision-language integration. PlanLLM [18] pioneers cross-modal joint learning by connecting world-level common sense with visual states via mutual information maximization, demonstrating robust performance in both closed-set and open-vocabulary scenarios. Parallel efforts have developed frameworks that concurrently process visual and linguistic planning signals to overcome spatial imagination limitations in pure LLM-based approaches [19]. The TaPA [20] framework further advances this direction by grounding LLM-generated plans in physical scene constraints through visual perception integration, while Robo2VLM [21] contributes a data generation pipeline that derives VQA queries from real robot trajectories to improve spatial reasoning in vision-language models.

Task decomposition and adaptive planning constitute another focus of research. Recent innovations include multi-modal grounded planning systems that achieve data-efficient learning in complex environments [22], and Egocentric Planning which combines symbolic planning with Object-oriented POMDPs for scalable task achievement [23]. The InterPreT [24] framework enables robots to learn symbolic predicates from non-expert language feedback, facilitating generalization to novel tasks. SMART-LLM [25] demonstrates how LLMs can coordinate multi-robot systems through programmatic task decomposition and coalition formation, while MPO [26] introduces meta-plans reusable high-level templates optimized

via execution feedback. The Embodied-Reasoner [27] extends visual search and reasoning to interactive tasks through a three-stage training pipeline, and PRED [28] enhances robustness by preemptively revising actions based on environmental discrepancy detection.

Benchmark development remains critical for evaluating progress in embodied planning. The Embodied Agent Interface [29] establishes standardized evaluation using Linear Temporal Logic to systematically assess 18 LLMs across key tasks such as goal interpretation and action sequencing. However, there remains a notable scarcity of large-scale benchmarks in this domain. To address this critical gap, we propose a novel benchmark specifically designed for evaluating complex long-horizon planning tasks, which will serve as a comprehensive testbed for assessing various foundational planning models of embodied intelligence.

The most analogous to the Embodied Planning Module in the present work is Cooperative Embodied Language Agent (CoELA) [30], a modular framework that integrates perception, memory, and communication modules for decentralized multi-agent collaboration. These advancements collectively push the boundaries of embodied AI by addressing fundamental challenges in reasoning, perception, and adaptive execution across diverse real-world scenarios.

1.3. Vision-Language-Action Models

Another significant line of research adopts an end-to-end approach to embodied intelligence through the Vision-Language-Action (VLA) paradigm, which heavily relies on robotics data typically collected via teleoperation or similar methods. RT-1 [31] pioneered transformer-based robot control through its discretized action transformer architecture, utilizing EfficientNet for visual processing and demonstrating scalable multi-task kitchen manipulation. Building upon this foundation, RT-2 [32] achieved breakthrough capabilities as the first vision-language-action model co-finetuned on both internet-scale visual question answering data and robotic manipulation data, employing PaLI-X architecture components.

Alternative approaches have demonstrated complementary strengths. SayCan [33] established a paradigm combining large language model planning with value function grounding, using PaLM [6] for high-level goal interpretation. ACT [34] introduced temporal ensembling and action chunking to achieve sub-millimeter precision in bimanual manipulation through its CVAE-Transformer architecture. The emergence of diffusion-based methods began with Diffusion Policy, which modeled multimodal action distributions and later incorporated UMI [35] framework improvements. Octo [36] set new benchmarks as a generalist diffusion policy trained on over 4 million trajectories across 22 platforms using the Open X-Embodiment dataset. OpenVLA [37] demonstrated efficient transfer through LLaMA-2 adaptation with DINOv2/SigLIP visual encoders. RDT-1B [38] advanced diffusion models through its 1.2B-parameter architecture featuring a unified action space representation.

More recently, π_0 [39] implemented flow matching for high-frequency control using PaliGemma components and demonstrated exceptional cross-platform deployment capabilities for robotic manipulation tasks. FAST[40] introduced frequency-space action tokenization for 15x inference acceleration. Gemini Robotics leveraged Gemini 2.0 foundation model capabilities for dexterous manipulation. Helix [41] achieved 200Hz humanoid control through optimized transformer policies, while GR00T [42] developed a unified diffusion framework for

humanoid systems using Eagle-2 VLM components based on real-world robot data and extensive IsaacSim data. These advances collectively demonstrate rapid progress along multiple dimensions: scaling through foundation model approaches, specialization for particular control regimes, and novel architectural innovations in action representation and policy learning.

Our Perception-Action Module adopts π_0 as its foundational architecture. The module's implementation involves a two-phase training approach: initializing with the pre-trained π_0 model's parameters followed by domain-specific post-training using our proprietary dataset collected through extensive teleoperation and simulation experiments. This dataset comprises multi-modal observations paired with corresponding action trajectories across diverse manipulation tasks, enabling the model to maintain π_0's robust generalization capabilities while adapting to our target operational environments and task requirements.

1.4. Locomotion and Whole-Body Control

For locomotion and whole-body control (WBC), the dominant technical route involves simulation-based learning with sim2real transfer. Lifelike [43] demonstrated this by training locomotion policies by tracking motion capture data before deploying to quadruped robots. OmniH2O [44] enabled both teleoperation and autonomous control of full-size humanoids through GPT-4o or learned policies. BeamDojo [45] introduced specialized rewards for polygonal feet locomotion, and various frameworks like Exbody2 [46], HoST [47], and GMT [48] advanced whole-body control through innovative training methodologies combining RL, behavior cloning, and motion prior integration. HOVER [49] and ASAP [50] further pushed the boundaries of agile humanoid motion through unified policy distillation and delta action learning for easier robot deployment. Our locomotion model follows the simulation-based learning route, with the primary objective of developing a more universal training pipeline capable of rapid cross-platform adaptation without requiring robot-specific parameter tuning.

1.5. Summary

While the field of embodied intelligence has witnessed significant progress across multiple research directions, there remains a notable absence of comprehensive systems capable of addressing all these aspects in an integrated manner. The TAIROS platform proposed in this work represents a holistic system encompassing perception, planning, and execution capabilities through three core functional modules: the Embodied Perception Module, Embodied Planning Module, and Perception-Action Module. These modules are seamlessly integrated through standardized interfaces to form a unified embodied intelligence agent capable of executing end-to-end robotic tasks with cross-platform adaptability. Robot platforms meeting hardware specifications can directly access TAIROS services through API calls or SDK-based edge deployment. Importantly, the platform maintains flexible modularity, allowing each component to be independently invoked for specific functions such as visual question answering (VQA) in perception, query-based planning, or edge deployment of VLA and WBC execution. The TAIROS platform has already been successfully deployed across multiple robotic platforms from Unitree, PaXini, Leju, Dobot, and Engine AI, etc., demonstrating its practical applicability and versatility in real-world scenarios.

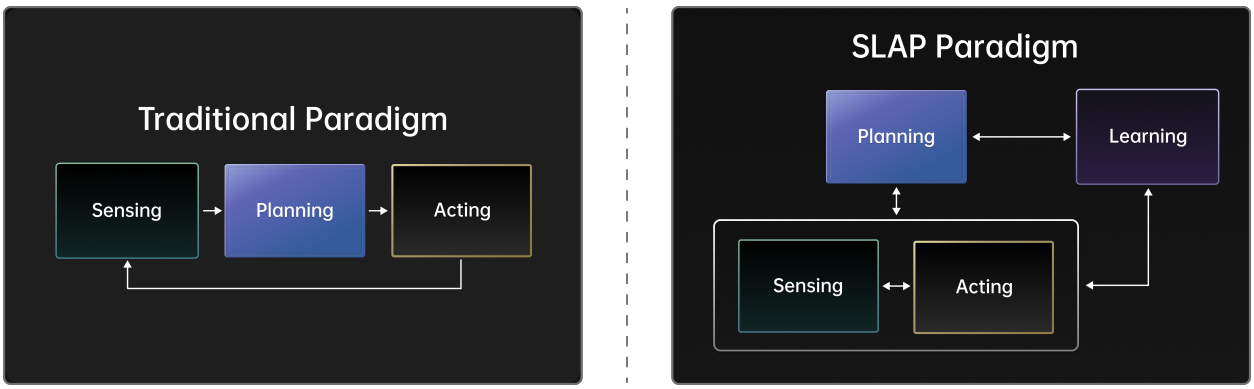

Figure 1: Paradigm shift from sensing-planning-acting to SLAP in the field of robotics.

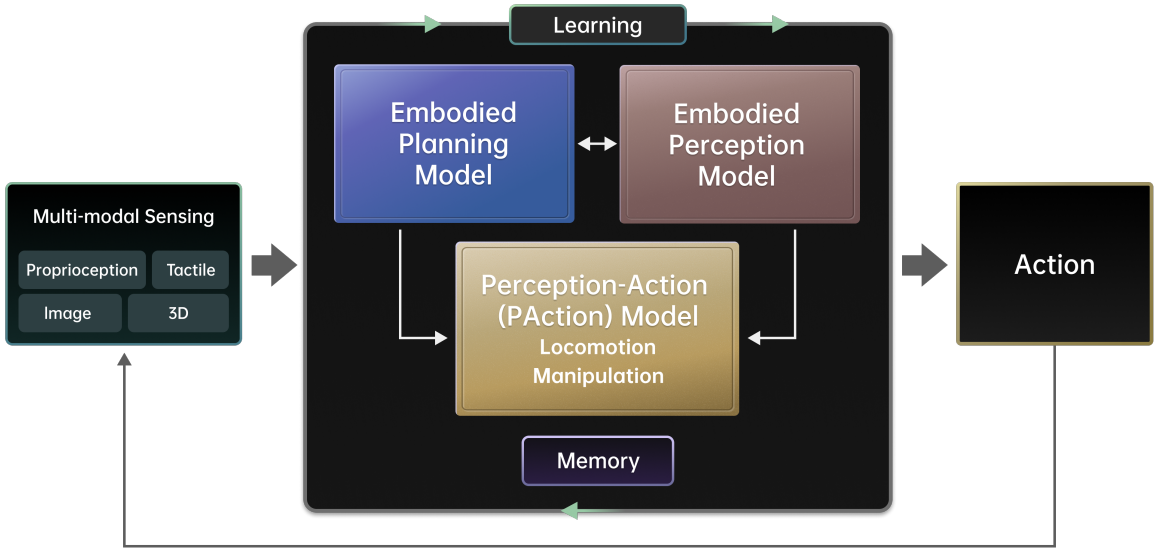

Figure 2: A framework overview of the TAIROS platform.

2. TAIROS Framework Overview

The field of robotics has undergone a fundamental transformation in its architectural paradigm, evolving from the classical sensing-planning-action loop to the new SLAP framework as depicted in Figure 1. The classical sensing-planning-action loop cannot deal with fast environmental incidents such as tripping by stones during walking and a slipping cup during grasping, and is thus lack of reactive autonomy. The SLAP framework we proposed in 2018 consists of Sensing, Learning, Action, and Planning. The notable difference is the tight coupling of Sensing and Action at the lower level, allowing fast reaction to the changing environment. This is consistent with System 1 in human cognition [51]. Only when dealing complex tasks, the Planning is called upon, which is consistent with System 2 in human cognition. The Learning infiltrates every module of Sensing, Action, and Planning.

After years of continuous research and development, our colleagues at Tencent Robotics

X Lab have refined this framework through persistent iteration. Now it has evolved into a more comprehensive and robust core technological framework, which we call the **SLAP**^{3} system (Sensing-Learning-Action: Perception, Planning, PAction, where PAction stands for Perception-Action). The TAIROS platform is built upon the **SLAP**^{3} framework. See Figure 2 for an overview. TAIROS consists of three main modules that focus on perception, planning, and execution, respectively. The Embodied Perception Module ingests multi-modal data from a range of sensors, including robot proprioceptive signals, camera images, depth maps or point clouds from depth cameras or LiDAR, as well as tactile and force sensor inputs. Using these inputs, the module reconstructs a dense 3D point cloud, performing object-level geometric fusion and semantic annotation to generate a hierarchical scene graph. This scene graph functions as the robot's long-term memory, enabling efficient information summarization, querying, and retrieval. By integrating multi-modal sensory information into a hierarchical and structured format, the robot can continuously and systematically perceive and update its environment, which in turn provides robust support for advanced reasoning and decision-making over extended periods. The Embodied Planning Module is an LLM-based reasoning agent that receives user prompt and environment context from the Embodied Perception Module, and then performs long-horizon reasoning through MCTS [52], CoT [53], and tool calling [54], etc., to decompose a difficult task into sub-tasks, each of which can be completed by calling the Perception-Action (*PAction*) Module. The PAction Module receives commands from the Embodied Planning Module and vision-tactile-force-language embeddings from the Embodied Perception Module to output robot actions. The Perception-Action module currently contains two specific models for legged robot locomotion and gripper/dexterous-hand manipulation separately. The locomotion model is trained in simulation using RL and deployed on real robots through a general sim2real pipeline. The manipulation policy is a VLA model based on an architecture similar to π_0 [39]. In the future, the locomotion and manipulation models will be unified. The three modules compose the complete embodied agent for end-to-end deployment over any robot hardware platform that meets a certain requirement. Meanwhile, each of the three modules can be called independently via self-contained APIs (service from the cloud) or SDKs (for edge deployment). For example, the Embodied Perception Module enables text prompt interaction with users, acting like a VLM for question answering and scene understanding; the Embodied Planning Module can chat with users and help solving long-horizon decision problem via text responses; the Perception-Action Module can be deployed in robot hardware for direct locomotion and manipulation tasks. Please refer to the official site for direct usage: `https://tairos.tencent.com/docs`. In the following sections, we will elaborate each module in technical detail.

3. Embodied Perception Module

The Embodied Perception Module is designed to equip embodied robots with advanced environmental perception and memory capabilities. To this end, we maintain a hierarchical scene graph that is updated in real time and online, continuously capturing and organizing information about the robot's surroundings. This enables the robot to construct and dynamically update a structured, semantic 3D representation of complex and changing environments

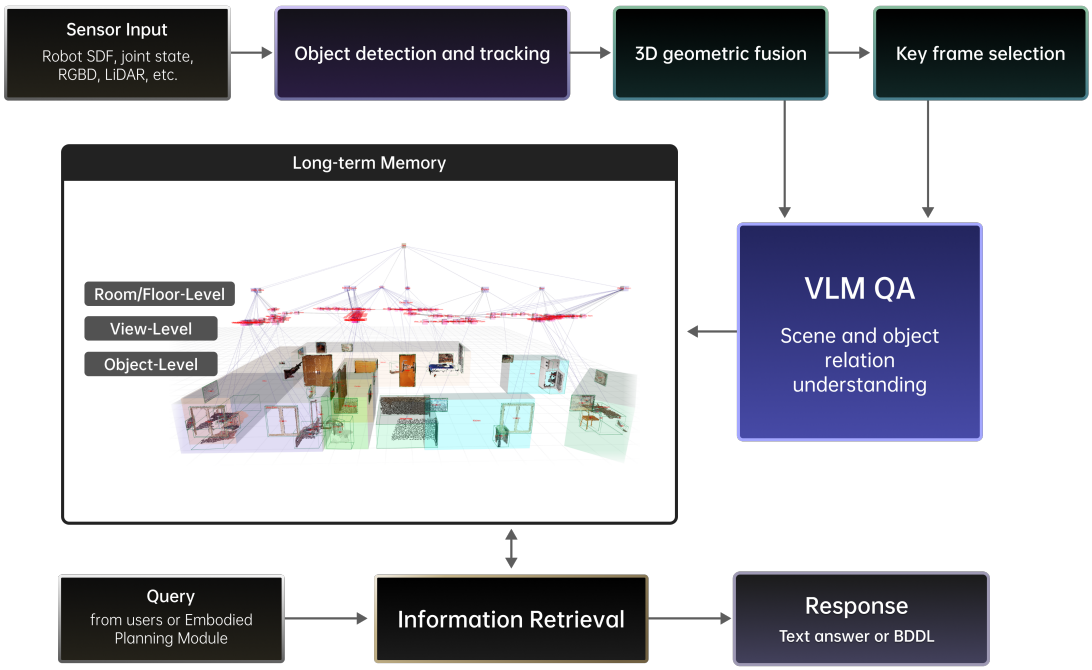

Figure 3: The pipeline of the Embodied Perception Module.

as they are perceived. Through this real-time, online updating process, the robot achieves robust, up-to-date understanding and interaction within dynamic scenes. The entire pipeline is depicted in Figure 3.

3.1. Key Frame Selection

The robot acquires environmental observations as continuous temporal signals, such as sequential RGB-D video frames. However, continuous sampling results in substantial data redundancy, and the quality of individual frames may be degraded by factors such as motion blur during robot movement. Processing every frame in real time is both computationally inefficient and unnecessary. To address these challenges and optimize computational efficiency, we implement a real-time key frame selection mechanism that identifies and retains only the most informative and high-quality frames from the observation stream. Specifically, frames are first considered as key frame candidates if there is significant camera motion—defined as a translational displacement greater than 0.5 meters or a rotational change exceeding 30 degrees relative to the last selected key frame—or if significant dynamic changes are detected during periods of static or slow movement, such as the movement of objects or people within the scene. Each candidate frame then undergoes a rigorous quality assessment, where image sharpness is evaluated using Laplacian variance analysis to filter out blurred frames, and content relevance is verified through object detection outputs. Only frames that meet all motion, dynamic change, and quality criteria are ultimately selected as key frames.

This multi-stage selection strategy ensures that the system processes only high-value visual data, thereby maintaining robust and up-to-date environmental perception while significantly improving the efficiency of the perception pipeline.

3.2. 3D Reconstruction

Based on the selected key frame observations, we first perform real-time 3D reconstruction of the robot's surrounding environment, generating a spatially consistent representation of the observed scene. In parallel, we utilize open-vocabulary 2D object detection models, such as YOLO-World [10] and Grounding DINO [9], to semantically identify objects within each frame. The detected objects are then precisely segmented using the Segment Anything Model (SAM) [7], enabling the extraction of accurate object masks. To ensure robust multi-view tracking and maintain object identity across different perspectives, we further employ instance association techniques based on SAM2 [8]. Throughout this process, robot self-occlusions are systematically filtered out, ensuring that only external scene elements are considered and thereby enhancing the reliability of the environmental model.

Subsequently, the detection and segmentation results are integrated with the 3D reconstruction to achieve spatiotemporal semantic consistency across the observed scene. Specifically, we leverage depth information in conjunction with the camera's intrinsic and extrinsic parameters to project detected objects into the world coordinate system, thereby aligning object-specific point clouds within the 3D reconstruction and estimating the corresponding Z-axis-aligned oriented 3D bounding boxes. For objects identified as the same instance across multiple views, we perform point cloud fusion based on both spatial proximity in the world coordinate system and semantic similarity, consolidating redundant observations into unified object representations. Through this comprehensive pipeline, the robot is able to extract and represent the semantic distribution and precise spatial locations of objects in its surrounding environment, providing a robust foundation for the subsequent construction of a hierarchical scene graph.

3.3. Scene Understanding

To further enhance the robot's ability to perceive object attributes, spatial relationships, and scene types within its environment—and to facilitate the formation of effective scene memory—we incorporate large-scale vision-language models (VLMs) to provide rich semantic information at both the object and scene levels. Specifically, we leverage VLMs to query a variety of object properties, including color, on/off state (such as for lights or refrigerators), category, orientation, and functional usage. In addition, we utilize the VLM to infer spatial relationships between objects, such as "upon," "under," "near," "in," and "contain," thereby establishing semantic associations among objects in the scene. For example, we prompt the VLM to identify functional objects that can serve as surfaces (such as tables or sofas, which may have other objects placed "on" them) and containers (such as refrigerators, which may have objects placed "in" them). The VLM is then used to detect objects that exhibit "on" relationships with these surfaces (e.g., apples and bread on a table) and "in" relationships with containers (e.g., apples and bread inside a refrigerator). Furthermore, we prompt the VLM to perform scene clustering, generating scene labels such as "Living Room," "Bedroom,"

or "Kitchen," along with a concise one-sentence description of the scene. This comprehensive semantic analysis enables a detailed understanding of the spatial organization and object interactions within the environment.

3.4. Hierarchical Scene Graph

Building upon the extracted spatial relationships and the fused point cloud data, we construct a comprehensive multi-layered scene graph that captures the hierarchical structure of the environment. At the object level, 2D object detections are integrated with point cloud back-projection to generate precise 3D bounding boxes for each detected entity. The vision-language model (VLM) is employed to extract detailed object attributes, such as category, color, and functional state, which are combined with the 3D bounding box information to form the data structure of each object node. In addition, the VLM infers spatial relationships between objects—such as "on" or "in"—which are encoded as semantic edges connecting the relevant object nodes, thereby enriching the graph with contextual information about object interactions.

At the view level, each key frame is represented as a view node, encapsulating information such as camera pose and associated observations. Object nodes that are visible within a particular key frame are linked to the corresponding view node through hierarchical edges, establishing parent-child relationships that connect the object and view levels. Moving up the hierarchy, the room level abstracts spatial regions or rooms within the environment as room nodes. The VLM facilitates the clustering of temporally and spatially related views—such as a sequence of key frames all depicting a kitchen—by linking their respective view nodes to a single room node via hierarchical edges. At the highest level currently supported, the floor level, all room nodes are connected to a single floor node, reflecting the assumption of a single-floor environment in the present implementation.

This multi-layered scene graph provides a structured and hierarchical representation of the environment, seamlessly integrating object detections, spatial relationships, and semantic context across multiple levels of granularity. Such a representation not only supports efficient scene understanding and memory, but also lays a robust foundation for advanced reasoning and decision-making tasks in robotic applications.

3.5. Downstream Tasks

The hierarchical scene graph is updated online as the robot interacts with the environment and is kept as long-term memory. The memory supports both direct user interaction via text prompt and calling by the Embodied Planning Module. For downstream task support, a hybrid retrieval module processes user or Embodied Planning Module's queries via efficient retrieval (including both spatial retrieval and semantic reasoning to infer implicitly related objects). Finally, the retrieved entities and their spatial relations are formatted in BDDL [55], which facilitates integration with the planning system. An example of the BDDL format is as below.

```
(:objects
    box.n.01_1 - box.n.01
```

```
    chair.n.01_1 chair.n.01_2 chair.n.01_3 chair.n.01_4 - chair.n.01
    coffeetable.n.01_1 - coffeetable.n.01
    creditcard.n.01_1 - creditcard.n.01
    diningtable.n.01_1 - diningtable.n.01
    drawer.n.01_1 - drawer.n.01
    floor.n.01_1 - floor.n.01
    ......
    television.n.01_1 - television.n.01
    vase.n.01_1 - vase.n.01
    watch.n.01_1 - watch.n.01
    window.n.01_1 window.n.01_2 window.n.01_3 - window.n.01
)

(:init
    (open box.n.01_1)
    (toggled_on floorlamp.n.01_1)
    (open laptop.n.01_1)
    (toggled_on lightswitch.n.01_1)
    (ontop sofa.n.01_2 chair.n.01_4)
    (ontop remotecontrol.n.01_1 coffeetable.n.01_1)
    (ontop box.n.01_1 coffeetable.n.01_1)
    (ontop keychain.n.01_1 coffeetable.n.01_1)
    (ontop watch.n.01_1 coffeetable.n.01_1)
    .....
    (inroom window.n.01_1)
    (inroom window.n.01_2)
    (inroom window.n.01_3)
)
```

This workflow bridges low-level perception with high-level scene abstraction, enabling robots to reason about environments in both geometric and semantic dimensions.

The system processes user queries through a structured pipeline where the LLM first interprets the input instruction to determine whether it falls within the scope of multi-modal perception capabilities. If the query lies outside this operational domain, the system directly generates an appropriate rejection response. For valid queries, the LLM dynamically selects the optimal query modality, currently including options such as current field of view search, directional/distance-based search, room-specific search (either current or designated), or global environment search while simultaneously determining the corresponding query parameters. The retrieved results, combined with the original user instruction, are then formatted into a comprehensive prompt for the LLM, which subsequently generates both a natural language response and visualizable outputs (such as target object IDs for 3D visualization or navigation point computation). This integrated approach enables context-aware information retrieval while maintaining robust rejection handling for out-of-distribution requests.

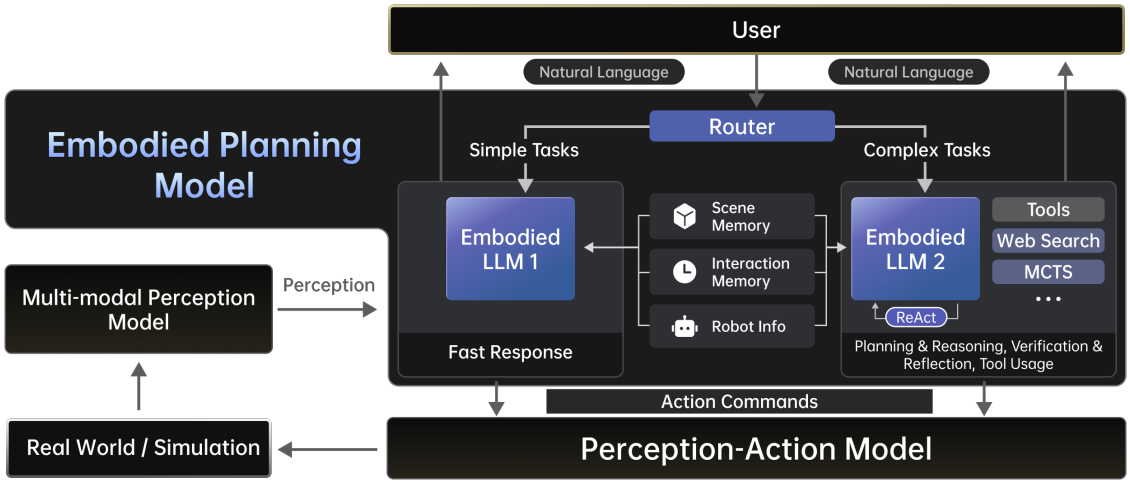

Figure 4: The pipeline of the Embodied Planning Module.

4. Embodied Planning Module

The Embodied Planning Module plays a crucial role in the overall system, serving as the interface that directly receives user instructions through voice interaction while simultaneously processing real-time multi-modal perceptual data from the Embodied Perception Module for semantic understanding. Meanwhile, it coordinates task execution by commanding the Perception-Action Module to ensure successful robotic operation, with its complete workflow illustrated in Figure 4.

4.1. Router

Upon receiving a voice instruction, the module first converts it into natural language text via automatic speech recognition (ASR), then employs a router LLM model for binary classification with linguistic output (e.g., generating "simple" or "hard" labels) to categorize task complexity. Simple tasks, such as direct verbal responses or basic action commands for the Perception-Action Module, are handled by the Fast Embodied LLM, while complex tasks requiring long-horizon planning are delegated to the Planning Embodied LLM. This bifurcated architecture optimizes computational efficiency and interaction delay by allocating resources according to task demands.

4.2. Planning Embodied LLM

The Planning Embodied LLM functions as a sophisticated reasoning agent, integrating several state-of-the-art LLM-based techniques, including tool calling [54], Chain of Thought (CoT) [53], MCTS, RL, etc. We elaborate each valuable technique used in the system in the following.

Tools. The tool-based approach plays a pivotal role in LLM-based agent, where we propose multiple specialized functions. The *Plan* tool decomposes complex tasks into executable sub-tasks for the Perception-Action Module, while the *Action* tool generates meta actions at the

sub-task level, such as navigation commands for locomotion models or language instructions for VLA models. The *Error Handle* tool triggers re-planning or re-acting when sub-task execution fails, ensuring robustness. Additionally, the *Visual QA* tool facilitates interactive queries with the Embodied Perception Module to retrieve relevant visual information, and the *Web Search* tool fetches task-related knowledge from the internet. For environmental interaction, the *Explore* tool enables active exploration (e.g., object search), and the *Translation* tool handles multilingual communication. A *Critic* tool, implemented as a VLM module, evaluates task progress and robot states to guide decision-making. Finally, the *Terminate* tool signals task completion. Together, these tools enable the agent to perform long-horizon planning with adaptive reasoning and recovery mechanisms.

Active Exploration. To further enhance active exploration capability in long-horizon tasks, we integrate a multi-turn multimodal reinforcement learning framework into the Planning Embodied LLM, specifically optimizing the strategy generation of the *Explore* tool. This approach significantly boosts the agent's active search and memory retrieval capabilities in unknown environments through two key designs. First, multi-turn interactive exploration: upon receiving initial task instructions, the agent can independently determine multiple rounds of exploratory actions (e.g., <get_memory> to retrieve historical observations, <action> to perform physical interactions). The agent dynamically adjusts the direction of each round of actions based on the current scene graph and visual observations, continuing until the termination condition is triggered. This mechanism transforms exploration from a one-time blind search into an adaptive process with contextual memory. Second, reward-driven exploration optimization: we have designed a fine-grained reward function that incorporates multi-dimensional metrics such as object-matching F1-score, exploration path efficiency, and format compliance. In particular, an "exploration reward" is introduced to quantitatively evaluate the environmental feedback from each round of actions, encouraging the agent to prioritize interactions with regions that maximize information gain.

Interruption. There are two types of interruptions: instruction interruption and action interruption. Instruction interruption occurs when the current instruction is in the process of tool invocation or queuing, but action execution has not yet started. At this point, the instruction processing is interrupted, and the result is stored in the historical instructions. Action interruption occurs when the task initiated by the instruction sends an action sequence and is awaiting the result. An additional interruption action is sent to stop the robot's actions promptly.

Instruction Tracking. The framework systematically manages the user input instruction process, recording reflection content, tool invocation status, task decomposition, and execution results. For any output action sequence, text response, or system message, the corresponding instruction ID is bound. Once instruction processing is complete or interrupted, the data is stored in the historical memory.

Agentic LLM. We propose two agents: a reactive agent based on a 32B base model, which processes historical information following instructions, selects the appropriate tools, and generates the corresponding tool parameters. The results of tool calls, as well as any exceptions encountered during tool invocation, are handled through reflection by this agent.

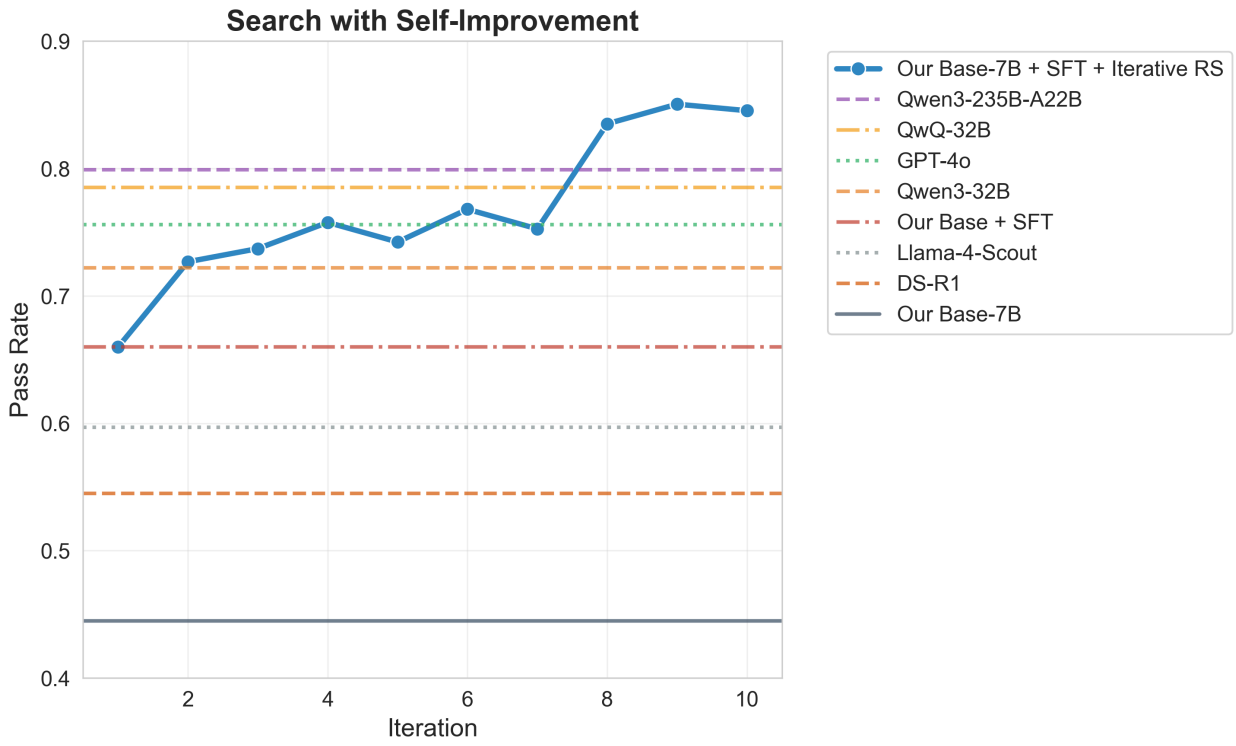

Figure 5: Ablation study of incorporating MCTS.

Any user's instruction is processed until the Terminate tool is invoked or the instruction is interrupted. A standard process-oriented (SOP) agent invokes tools following a defined procedure. When the current task needs to be broken down into actions from mid-level task, the Action tool is invoked. If errors occur during the execution of the actions, the Error-Handle tool is called. Upon completion of the task, the Critic tool is invoked to assess whether the task execution aligns with expectations.

Search. In embodied tasks, particularly execution in real-world or simulated environments, obtaining ground truth trajectories is extremely difficult, and manual labeling is costly. As a result, acquiring a large-scale supervised fine-tuning (SFT) dataset becomes a major challenge. Once a model has acquired basic capabilities with a small amount of data, self-improvement through generating its own training data becomes a reasonable approach. However, planning problems can still be complex, especially when dealing with long sequences or rare actions, as simple random sampling may fail to yield successful trajectories. MCTS addresses this by simulating future states and evaluating action paths through a tree structure, allowing for more informed decision-making. In the evaluation stage, the value is calculated via a trained value model and using MC rollouts. The action model, together with MCTS, can generate trajectories with higher quality. Furthermore, by establishing a self-improving loop where MCTS produces better trajectories, which are then filtered and used to train the model, leading to even better trajectories. We have observed that after 5 to 10 iterations, the model's performance continues to improve. We conduct experiments with a 7B Base model and on

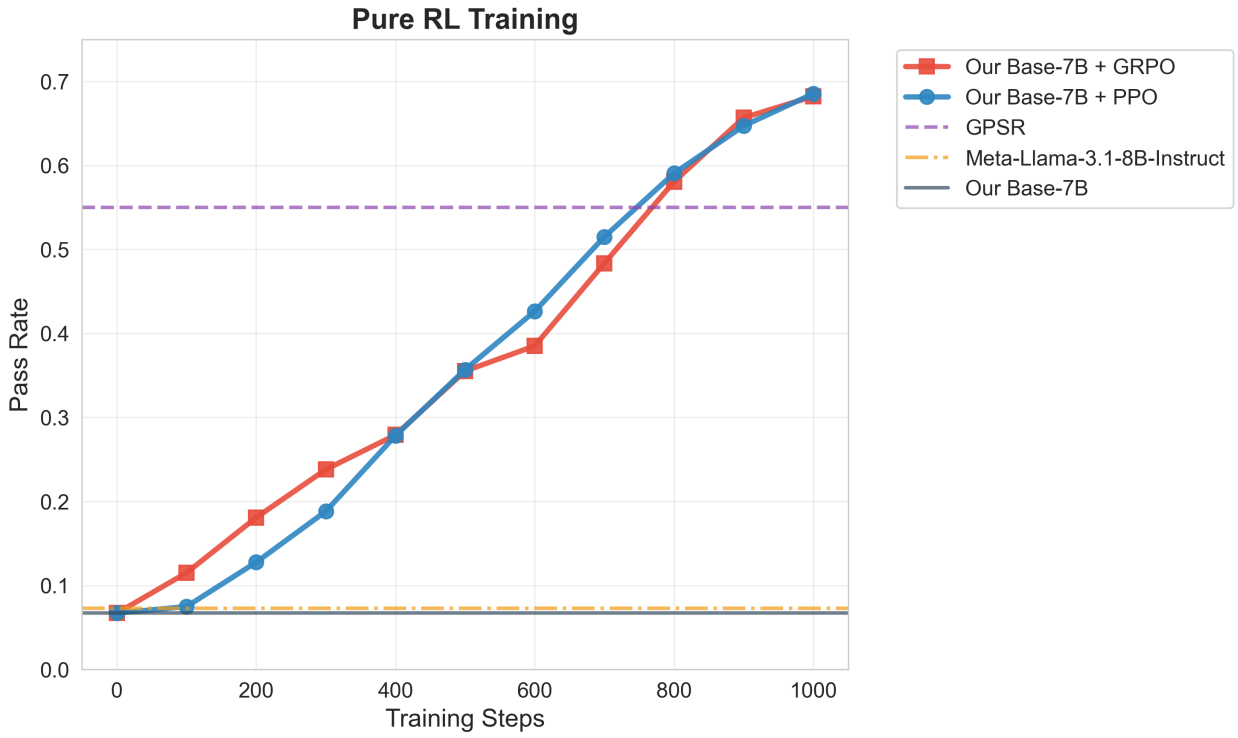

Figure 6: Ablation study of incorporating RL training.

an embodied symbolic simulation environment. With the same tasks, this final performance in terms of pass rate outperforms all the baselines, including closed-source LLMs and as well as the latest reasoning models. Figure 5 shows the comparison results of various models.

RL. The emergence of pure RL training paradigms on LLMs, such as DeepSeek-R1 [56], provides another path for training embodied agents. It learns to understand the environment (i.e., transitions) and solve tasks through exploration, interaction with the environment, and reward feedback. We started directly from a general model without going through embodied task-specific SFT before proceeding with PPO/GRPO. For reward feedback, we provided sparse outcome rewards generated by a CoT-Reward Model. The trained reward model scores based on the effectiveness, rationality, and efficiency of the final execution, producing an overall score. Additionally, if an error occurs during execution, there is an extra penalty. We used the Verl framework [57] and integrated a large number of cloud-based simulations, providing execution status and results via headless execution for RL rollouts. We used a 7B model as the base model and AI2Thor (ALFRED) as the environment. Experimental results in Figure 6 show that RL training can effectively improve the model's capabilities, and outperforms other closed-source and open-source models, as well as some prompting based method that specifically designed for ALFRED.

Reward Model. The Reward Model plays a crucial role in the training process of RL. We leverage the CoT Reward Model to score the execution trajectories of agents. A trained 7B model, through deep thinking, primarily evaluates a trajectory based on three key aspects:

1) Effectiveness (task completeness - for example, if the task is to fry an egg, the egg needs to be cooked in the pan); 2) Rationality, such as deeming the trajectory irrational if it contains illegal actions during execution; 3) Efficiency, which involves avoiding redundant and ineffective actions or explorations. Ultimately, we generate an overall score based on these three perspectives, serving as an evaluation of the trajectory's execution effect.

4.3. Bridging Sub-task and Action

The action model, after extensive data and RL training, is capable of executing both simple tasks (e.g., "Navigate to the fridge") and more complex ones (e.g., "Take all the fruits from the fridge and place them on the table"). In the process of sub-task decomposition, the planning model must determine an appropriate granularity for the tasks (along with descriptive text), ensuring that the action model can execute them effectively while avoiding excessive decomposition. A task that is too finely broken down could lead to unnecessary complexity and suboptimal performance. Inspired by the Meta Plan Optimization (MPO) approach [26], we optimize the planning model by aligning it with a fixed, trained action model. Initially, we manually curate a small set of planning data (a few dozen examples) and train the planning model via SFT to develop basic planning capabilities. Following this, the planning model, operating at a higher temperature, performs extensive sampling for each task. These generated plans are then executed in simulation by the action model. A critic evaluates the execution's success, and based on the results, we refine the planning model through DPO. The criteria for positive and negative examples are as follows:

```
Positive Example: Successful execution with fewer steps.
Negative Example: Execution failure
```

Through this training process, our planning model integrated with the action model in benchmark object-interaction tasks improves success rate significantly. Compared to the baseline SFP version initialized with the same 32B base model, the success rate improved from 37.3% to 45.7%. The trained planning model's thinking process is as follows:

```
<think>
The task is to move the book to the bed and then move the phone to where
the book was originally located.

KEY OBJECTS:
- book.n.01_1 (ontop desk.n.01_1)
- bed.n.01_1 (ontop floor.n.01_1)
- cellphone.n.01_1 (ontop desk.n.01_2)

The robot needs to:
1. Pick up the book.n.01_1 from desk.n.01_1 and place it on bed.n.01_1.
2. Pick up the cellphone.n.01_1 from desk.n.01_2 and place it on desk.n.01_1
where the book was originally located.
```

```
The robot must ensure that the book is moved to the bed before moving the
phone to the desk.
</think>

<mid_tasks>
1. Pick up the book.n.01_1 from desk.n.01_1 and place it on bed.n.01_1.
2. Pick up the cellphone.n.01_1 from desk.n.01_2 and place it on desk.n.01_1
where the book was originally located.
</mid_tasks>
```

4.4. Memory and Context

In Figure 4, the scene memory is obtained from the Embodied Perception Module and kept in RAG as persistent memory. The interaction memory refers to the history of interactions, including dialogues, actions, tools used, and some special feedback (e.g., the task failure information), along with the timestamp of each event. Robot info includes robot types (e.g., humanoid robots with two arms, quadrupedal robots, etc.), robot functions (such as navigation, grasping, etc.), and robot descriptions (e.g., name and owner). The interaction memory and robot info are organized as prompt context.

4.5. Benchmark

Furthermore, we have developed a comprehensive benchmark specifically designed for evaluating long-horizon and challenging embodied decision-making tasks. This benchmark comprises 1,011 task samples distributed across seven primary categories: Object-Interaction (363 samples, 35.9%), QA-Attribute (144 samples, 14.2%), QA-Alignment (131 samples, 13.0%), QA-Self-awareness (122 samples, 12.1%), QA-Spatial (104 samples, 10.3%), Navigation (80 samples, 7.9%), and QA-Temporal (67 samples, 6.6%). It supports three distinct robot configurations: Single-armed Robot, Dual-armed Robot, and Mobile Base (2.8%), and is primarily tested in four typical indoor environments: Kitchen (39.9%), Bedroom (29.0%), Living Room (22.3%), and Bathroom (8.7%). The evaluation framework is designed for end-to-end assessment, providing a multi-dimensional analysis that includes task comprehension, action sequence generation, reflective capability, task success rate, and user satisfaction. Additionally, it incorporates a hierarchical scoring system tailored to different task categories, employing metrics such as exact match, answer similarity, LLM-based scoring, and success rate to ensure a thorough and nuanced evaluation of embodied AI systems.

4.6. Overall Evaluation

We conducted evaluations using a comprehensive pipeline specifically designed for embodied tasks, covering the entire process from task understanding to user satisfaction. This end-to-end evaluation encompasses key phases including goal state prediction, task rejection, action sequence generation, reflective abilities, tool usage, and task completion. Evaluation is performed only on the dimensions listed in each sample's `evaluation dimensions`. The scoring metrics are described as follows:

Task Understanding. Goal Task Understanding: compute the semantic similarity between model-predicted `task` and ground truth `task`; Goal State Prediction: semantic similarity between generated `goalState` and ground truth; Task-Related Scene Graph Accuracy: exact match comparison (ignoring order) after normalization and regular expression extraction of scene graph entries.

Let $\text{sim}(x, y)$ be the cosine similarity between two text embeddings x and y, and $\text{EM}(x, y)$ the exact match score (1 if match, else 0). Define task understanding score S_{under} as:

$$\begin{aligned} S_{\text{under}} = & \frac{1}{3}\left(\text{sim}(\texttt{task}, \texttt{gt\_task})\right) \\ & + \frac{1}{3}\left(\text{sim}(\texttt{goalState}, \texttt{gt\_goalState})\right) \\ & + \frac{1}{3}\left(\text{EM}(\texttt{sceneGraph}, \texttt{gt\_sceneGraph})\right) \end{aligned} \tag{1}$$

Task Rejection Accuracy. Compare the boolean field `acceptTask` with the ground truth. The score S_{rej} is:

$$S_{\text{rej}} = \text{EM}(\texttt{acceptTask}, \texttt{gt\_acceptTask}) \tag{2}$$

Tool Usage. Match the correct tool call (e.g., `get_time()`, `vqa(text)`, `weather()`) and verify parameter correctness and simulation success:

$$S_{\text{tool}} = \frac{1}{2}\left(\text{Match(tool)} + \text{Success(execution)}\right) \tag{3}$$

Action Sequence and Task Completion. Let A be the predicted action sequence and G the ground truth sequence. Define:

- Success Rate (Succ.): Fraction of tasks completed successfully.
- Goal Condition Success (GcS): Fraction of predicates in final state matched to the goal.
- Success weighted by Path Length (SPL):

$$\text{SPL} = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{S_i L_i}{\max(P_i, L_i)} \tag{4}$$

where S_i is success (1 or 0), L_i is the length of optimal path, and P_i is path length taken. These scores are averaged to form S_{action}:

$$S_{\text{action}} = \frac{1}{3}\left(\text{Succ.} + \text{GcS} + \text{SPL}\right) \tag{5}$$

Certain subcategories require LLM-based grading for S_{action}, using structured prompts incorporating ground truth and prediction.

Answer Similarity (QA). Depending on category and subcategory:

$$S_{\text{qa}} = \begin{cases} \text{sim}(\texttt{answer}, \texttt{gt\_answer}) & \text{if similarity-based} \\ \text{LLMScore}(\texttt{answer}, \texttt{gt\_answer}) & \text{if model-based} \end{cases} \tag{6}$$

For dimensions using LLM evaluation, the following prompt is employed:

```
You are a careful evaluator. Please rate the following response (score 0 to
1) with respect to the reference answer, based on correctness, relevance,
and completeness. \textbackslash n\textbackslash nReference Answer: [ground
truth]\textbackslash n\textbackslash nModel Answer: [prediction]
\textbackslash n\textbackslash nScore:
```

Reflective Ability. Defined as the proportion of failed actions that were corrected with a successful follow-up, forming the score S_{ref}:

$$S_{\text{ref}} = \frac{\text{Num(Effective Reflections)}}{\text{Num(Failure Events)}} \tag{7}$$

Alternatively, LLM-based grading can assess whether recovery was adequate.
Total Score. We define a weighted total score for each sample:

$$S_{\text{total}} = \sum_{i \in \mathcal{D}_{\text{enabled}}} w_i S_i \tag{8}$$

where $\mathcal{D}_{\text{enabled}}$ is the set of active evaluation dimensions, and w_i the weight (equal weight by default or set via config). All scores are reported per subcategory and aggregated by category. For dimensions not applicable (e.g., $w_i = 0$), they are omitted from S_{total}. This structured framework enables in-depth evaluation and fair comparison across baselines such as ReAct and plan-and-execute, offering insight into performance across all critical embodied reasoning dimensions.

We compared the ReAct and plan-and-execute planning frameworks using different models. ReAct is a framework that combines reasoning and action in a recursive manner, where the model is able to adaptively react to the environment based on intermediate feedback. Plan-and-execute refers to a framework that focuses on first generating a high-level plan, followed by execution of the individual steps in the plan, often leveraging symbolic reasoning and task decomposition. This framework is more structured and hierarchical, in contrast to the flexibility of ReAct's recursive approach. We tested models such as GPT-4o, DeepSeek-R1, Qwen-Max-Latest, Robobrain-7B/32B, among others, within both frameworks. For various tasks, such as Object Interaction, Alignment, etc., we conducted evaluations using a comprehensive pipeline specifically designed for embodied tasks, covering the entire process from task understanding to user satisfaction. This end-to-end evaluation encompasses key phases such as goal state prediction, task rejection, action sequence generation, reflective abilities, tool usage, and task completion. The framework also incorporates a rich and detailed scoring system that evaluates the models' performance across multiple dimensions. For instance, LLM-based scoring assesses how well the model's responses align with the expected outputs. Additionally, similarity metrics are used to compare generated responses to reference answers, including exact match and partial match evaluations. Task completion rates, including sub-task completion rates and execution path efficiency, are also measured. These metrics provide a quantitative assessment of how effectively the model generates

coherent, accurate, and efficient responses. Furthermore, the task rejection component ensures that models can accurately identify tasks outside their capabilities, a crucial skill for real-world applications. The overall performance comparison is given in Table 1.

Table 1: Overall performance of all compared models. The scores represent the weighted total scores of all active evaluation dimensions of each model (row) under each task (column).

Models	Performance Metrics ↑						
	Object Interaction	Alignment	Self-awareness	Attribute	Spatial	Temporal	Navigation
Tairos-Planing	**60.82**	66.74	**74.38**	**66.07**	**52.62**	**53.44**	**62.09**
GPT-4o+ReAct	44.06	70.00	71.69	60.70	39.10	48.13	58.00
DeepSeek-R1+ReAct	45.93	67.54	71.66	60.99	38.24	46.51	60.82
Claude-4.0-Sonnet+ReAct	50.56	69.19	68.14	65.87	48.90	51.87	57.98
Gemini-2.5-Pro+ReAct	48.01	37.33	51.95	64.38	49.07	48.09	53.16
Qwen-max-Latest+ReAct	43.90	58.60	63.91	56.90	38.90	47.60	60.50
Robobrain-7B+ReAct	36.85	20.76	25.33	49.90	50.70	41.02	52.69
Robobrain-32B+ReAct	37.98	21.43	26.12	51.30	50.92	42.35	54.40
GPT-4o+plan-and-execute	40.30	**70.70**	73.90	57.90	35.90	46.10	58.80
DeepSeek-R1+plan-and-execute	45.80	67.40	71.10	60.80	37.90	46.30	57.80
Claude-4.0-Sonnet+plan-and-execute	49.54	69.88	68.74	64.93	48.89	50.25	56.36
Gemini-2.5-Pro+plan-and-execute	47.09	37.61	50.89	62.17	49.24	47.72	52.87
Qwen-max-Latest+plan-and-execute	42.50	58.30	64.14	55.60	36.80	47.20	59.10
Robobrain-7B+plan-and-execute	35.47	20.52	25.68	49.43	49.02	39.63	51.64
Robobrain-32B+plan-and-execute	36.93	22.21	27.31	49.63	50.40	43.23	53.38

5. Perception-Action Module

The Perception-Action Module adopts a dual-model architecture consisting of a VLA model for manipulation tasks and a simulation-based RL training pipeline for locomotion tasks, reflecting the prevailing technical approaches for these distinct task categories in the field. In our current system implementation, the manipulation model and locomotion model remain decoupled. This design ensures operational clarity while requiring careful coordination at the command level. The Embodied Planning Module addresses this by sequentially outputting corresponding commands for each model in its task execution pipeline, thereby preventing conflicting calls and maintaining system stability during concurrent manipulation and mobility operations. This architecture provides modular flexibility while ensuring reliable task execution through explicit command sequencing.

5.1. VLA for Manipulation

Our VLA model builds upon the foundational architecture of the π_0 model [39], incorporating improvements with data pipeline augmentation, 3D information grounding, and adapting to more heterogeneous application scenarios, including industrial manipulation and domestic service tasks.

Data Acquisition. We aim to train VLA models using data collected via both teleoperation and UMI [35] handheld gripper. We mainly introduce the UMI data training below. A critical goal is achieving cross-embodiment generalization, where policies learned from UMI data could transfer effectively to any other robotic arms. However, a significant visual domain

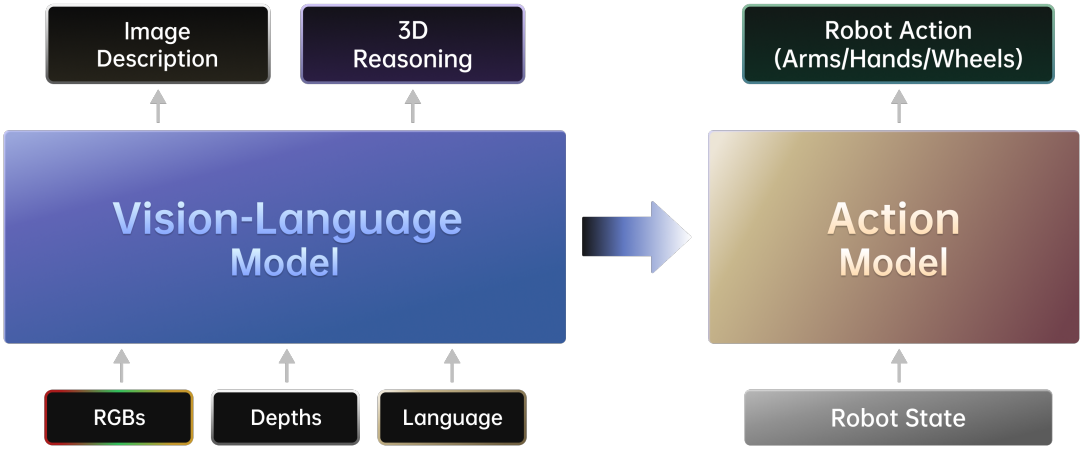

Figure 7: An illustration of the VLA model for manipulation.

gap exists between the UMI overhead-camera data and data captured by a target robot's overhead camera. This gap stems primarily from the prominent presence of the human arm in UMI data, compared to the robotic arm seen in target robot data, which makes them visually distinct from each other. Furthermore, the distinct kinematic configurations, the human arm typically operating horizontally versus many robotic arms moving vertically, create fundamental differences in background appearance and arm orientation. This visual discrepancy poses a challenge for visuomotor policies to transfer across embodiments. To mitigate this gap, we adopt a data editing pipeline similar to [58]. Our approach first detects and segments the human arms in UMI video frames. We achieve this using text-prompt open-vocabulary detection (Grounding DINO) followed by precise instance segmentation (SAM2). Then, we remove the arm region and restore the background plausibly, using the video inpainting model ProPainter [59]. Crucially, we use the gripper pose from UMI data to calculate the joint angles of the target robotic arm by solving inverse kinematics. Then, we leverage the known overhead camera extrinsics and calculated joint angles to render this virtual robotic arm composited onto the inpainted background, aligning its end-effector pose precisely with the recorded UMI gripper pose. This process generates synthetic overhead-view sequences that visually simulate the robots' view, significantly enhancing visual consistency for cross-embodiment policy transfer.

3D Alignment. Multi-view images are widely used in recent VLA approaches due to their implicit encoding of 3D information, which is crucial for spatial manipulation. However, learning robust multi-view representations typically requires large-scale real-world teleoperation data, which is often limited in robotics. To inject stronger cross-view spatial understanding into VLA models, we leverage external 3D visual representations, rather than relying solely on the VLA models to learn them independently. Specifically, we adopt the 3D foundation model VGGT [60], which has shown strong 3D perception capabilities from 2D images, as a teacher model to guide VLA in learning powerful 3D visual correspondence. Nonetheless, VGGT is originally trained on scene-level datasets with moderate pose variation and overlapping views,

while robotic settings, particularly those using head-mounted and wrist-mounted cameras, involve much greater variation in pose and appearance. This domain gap hinders the direct applicability of VGGT to embodied tasks.

To bridge this gap, we generate a multi-view dataset of 58K photorealistic synthetic images, where a simulated Franka robot manipulates various objects in diverse indoor scenes. The dataset provides precise labels for multi-view camera poses and point cloud alignment. We use this high-quality dataset to fine-tune VGGT, enabling it to adapt to the head–wrist camera configuration and demonstrate zero-shot generalization in our real-world dual-arm robot scenarios. The fine-tuned VGGT is then used to generate cross-view-consistent features, which supervise the output hidden states of the prefix VLM model via an alignment loss. This guidance enables the VLM model to efficiently learn more powerful 3D visual representations from the limited robotics dataset. However, pre-trained VLM models are typically trained on large-scale internet data and encode strong semantic alignment between images and text. Directly aligning VLM features with VGGT features may lead to a loss of this large-scale pre-trained knowledge. To mitigate semantic forgetting during training, we therefore continue to train the VLM on VQA and object localization tasks using a next-token-prediction loss in parallel. Optionally, our method can leverage depth images as a known prior when available, providing additional guidance for the network to produce more accurate predictions with auxiliary information. Depth modality is processed by a block-specific MLP and is added token-wise at the middle of the transformer block.

The entire framework is trained end-to-end. VGGT is used only during training and removed at inference. This design enables the robot to better reason over diverse image streams (e.g., stereo, head, and wrist views), enhancing its understanding of 3D spatial relationships in complex manipulation tasks.

Applications. We utilize the Dobot X-Trainer robot, a dual-arm system equipped with wrist-mounted cameras on each gripper and an externally mounted overhead camera. The task is to enable the robot to accurately grasp essence bottles and insert them vertically into a container that features a hole at its bottom, with a diameter closely matching that of the bottle. This setup poses a significant challenge due to the tight clearance of the hole, requiring high-precision manipulation. Moreover, the initial positions of both bottles and containers are randomized, demanding strong spatial generalization from the model. To support post-training, we collect 1,000 demonstration trajectories. The deployed VLA model can achieve over 80% success rate. We also employ the PaXini Tora One humanoid robot as our experimental platform to address a representative industrial task involving the packing of multiple bottles with varying sizes and appearances (including laundry detergent and water bottles) on a moving conveyor belt assembly line. For this specific task scenario, we collected a dataset comprising 300 complete execution trajectories. Subsequent post-training on our base model using this dataset demonstrates significant performance improvement, achieving an average task (packing three objects as one task) success rate over 80% in the target industrial packing application. This result validates both the robot's capability in handling dynamic industrial manipulation tasks and the effectiveness of our data-driven training approach for complex robotic operations. In addition to teleoperation data, we

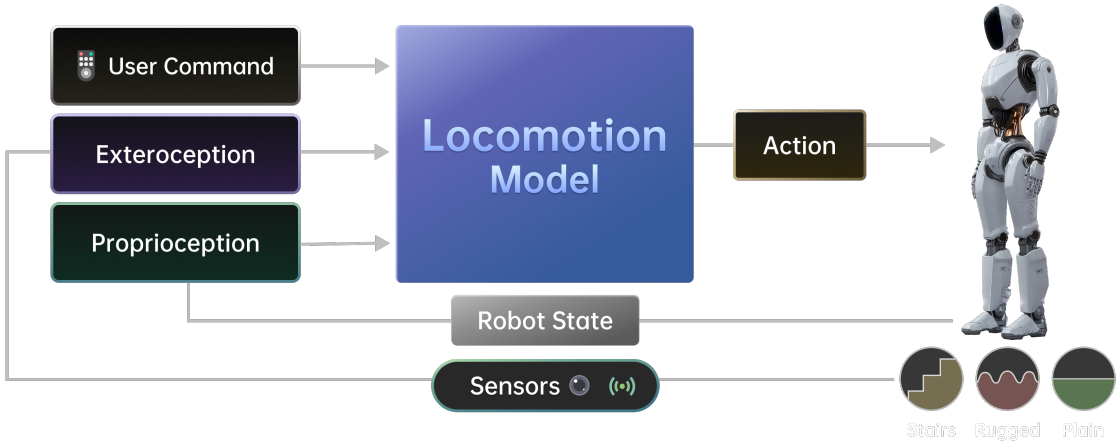

Figure 8: An illustration of the locomotion model.

also leverage data collected using hand-held grippers for model finetuning.This approach enables the acquisition of high-quality, dexterous manipulation trajectories that are otherwise challenging to obtain through teleoperation alone. We deploy and evaluate the fine-tuned model on the JAKA-K1 robot, which is equipped with the same type of gripper used during data collection (TEK CTAG2F90-C). Owing to the increased dexterity afforded by hand-held data collection, we are able to extend the previous packing task by introducing a bimanual handover step, in which bottles must be precisely transferred from one gripper to the other before insertion. We collect 500 demonstrations, enabling the fine-tuned model to achieve over 80% success rate. These results show that our hand-held gripper data enables fine-grained skill learning and successful transfer to real robots.

5.2. RL for Locomotion

The locomotion model is trained through simulation-based RL before being deployed to physical robots via a sim2real approach. When discrepancies emerge between simulated and real-world performance, we employ a systematic methodology involving real-robot data collection combined with techniques such as ASAP [50] and actuator modeling [61] to quantify and bridge the reality gap. These discrepancy models are subsequently incorporated back into the simulation training loop to refine the locomotion policy. We have developed a generalized training framework with an integrated real-robot data feedback pipeline, designed to maintain adaptability across diverse robotic morphologies while minimizing hardware-specific parameter tuning. This architecture has demonstrated robust performance across multiple commercial platforms including Unitree G1, Leju Kuavo, Pudu D9, Turling RX-V3, and Lexiang M001 robots. Please refer to our official website for demonstration videos.

6. Conclusion

We present TAIROS, an integrated platform comprising three core modules: a multi-modal perception module, a long-horizon planning module, and a unified perception-action module.

These components are designed to operate independently through standardized APIs/SDKs and collectively as a complete agent, providing robots with comprehensive end-to-end task execution capabilities. TAIROS is specifically engineered to address practical industrial demands, supporting diverse robotic applications through its flexible architecture. The platform enables robot manufacturers to offer embodied intelligence services via standardized interfaces, significantly lowering the development barrier for third-party integration. Additionally, TAIROS incorporates cloud-based simulation capabilities that allow instant deployment of virtual environments for planning and perception model validation, complete with pre-configured robotic agents, scenarios, and tasks to accelerate capability demonstration. Looking forward, the platform will leverage Cloud inference clusters to deliver a fully integrated development ecosystem encompassing data collection/annotation, algorithm training, model validation, and OTA deployment to physical robots - creating a closed-loop workflow that enhances research efficiency and industrial adoption in embodied intelligence. Currently, TAIROS has demonstrated compatibility with diverse robotic morphologies, including bipedal/wheeled humanoids, quadrupeds, and robotic arms, supporting various end-effectors from grippers to dexterous hands. The platform has been successfully deployed in collaboration with multiple robotics companies in industries including manufacturing, automotive, home appliances, and exhibition services, validating its practical applicability across sectors.

FULL AUTHOR LIST

All author names in alphabetical order of last names:
Ling Chen, Liuzhu Chen, Peihao Chen, Weiheng Chi, Qinghui Dai, Yuchun Guo, Bo Han, Lei Han, Shuliang He, Wanjia He, Yufei Huang, Tiande Jiang, Yiyang Jing, Jiaming Li, Jie Li, Jingchen Li, Tingguang Li, Xiong Li, Yangyang Li, Haitao Lin, Yonggen Ling, Tianliang Liu, Yuandong Liu, Yuzhen Liu, Zhiqing Liu, Minglei Lu, Zisheng Lu, Bohan Ma, Siyi Qian, Jiyuan Ren, Jiapeng Sheng, Yunrui Shi, Sichang Su, Zeran Su, Manxi Sun, Xiao Teng, Ye Tian, Rui Wang, Shuai Wang, Yuquan Wang, Lingzhu Xiang, Xuantang Xiong, Youda Xiong, Jiawei Xu, Bin Yang, Xin Yang, Zihao Yu, Lijun Yue, Dongsheng Zhang, He Zhang, Jingbo Zhang, Shenghao Zhang, Yizheng Zhang, Yufeng Zhang, Zhengyou Zhang, Zibo Zhang, Rui Zhao, Hongyan Zhi, Cheng Zhou, Cheng Zhou, Weijie Zhou, Chengwei Zhu, Xiaomeng Zhu, Yajuan Zhu, Yonghui Zhu.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[3] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[4] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[6] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, et al. Palm-e: An embodied multimodal language model. 2023.

[7] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[8] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, et al. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[9] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer, 2024.

[10] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. Yolo-world: Real-time open-vocabulary object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16901–16911, 2024.

[11] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[12] Bangguo Yu, Yuzhen Liu, Lei Han, Hamidreza Kasaei, Tingguang Li, and Ming Cao. Vln-game: Vision-language equilibrium search for zero-shot semantic navigation. *arXiv preprint arXiv:2411.11609*, 2024.

[13] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al.

Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5021–5028. IEEE, 2024.

[14] Abdelrhman Werby, Chenguang Huang, Martin Büchner, Abhinav Valada, and Wolfram Burgard. Hierarchical open-vocabulary 3d scene graphs for language-grounded robot navigation. In *First Workshop on Vision-Language Models for Navigation and Manipulation at ICRA 2024*, 2024.

[15] Wonje Choi, Woo Kyung Kim, Minjong Yoo, and Honguk Woo. Embodied cot distillation from llm to off-the-shelf agents. *arXiv preprint arXiv:2412.11499*, 2024.

[16] Qi Zhao, Haotian Fu, Chen Sun, and George Konidaris. Epo: Hierarchical llm agents with environment preference optimization. *arXiv preprint arXiv:2408.16090*, 2024.

[17] Hanwen Wan, Yifei Chen, Zeyu Wei, Dongrui Li, Zexin Lin, Donghao Wu, Jiu Cheng, Yuxiang Zhang, and Xiaoqiang Ji. Embodiedagent: A scalable hierarchical approach to overcome practical challenge in multi-robot control. *arXiv preprint arXiv:2504.10030*, 2025.

[18] Dejie Yang, Zijing Zhao, and Yang Liu. Planllm: Video procedure planning with refinable large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 9166–9174, 2025.

[19] Jun Cen, Chenfei Wu, Xiao Liu, Shengming Yin, Yixuan Pei, Jinglong Yang, Qifeng Chen, Nan Duan, and Jianguo Zhang. Using left and right brains together: Towards vision and language planning. *arXiv preprint arXiv:2402.10534*, 2024.

[20] Zhenyu Wu, Ziwei Wang, Xiuwei Xu, Jiwen Lu, and Haibin Yan. Embodied task planning with large language models. *arXiv preprint arXiv:2307.01848*, 2023.

[21] Kaiyuan Chen, Shuangyu Xie, Zehan Ma, Pannag R Sanketi, and Ken Goldberg. Robo2vlm: Visual question answering from large-scale in-the-wild robot manipulation datasets. *arXiv preprint arXiv:2505.15517*, 2025.

[22] Taewoong Kim, Byeonghwi Kim, and Jonghyun Choi. Multi-modal grounded planning and efficient replanning for learning embodied agents with a few examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 4329–4337, 2025.

[23] Xiatoian Liu, Hector Palacios, and Christian Muise. Egocentric planning for scalable embodied task achievement. *Advances in Neural Information Processing Systems*, 36:54586–54613, 2023.

[24] Muzhi Han, Yifeng Zhu, Song-Chun Zhu, Ying Nian Wu, and Yuke Zhu. Interpret: Interactive predicate learning from language feedback for generalizable task planning. *arXiv preprint arXiv:2405.19758*, 2024.

[25] Shyam Sundar Kannan, Vishnunandan LN Venkatesh, and Byung-Cheol Min. Smart-llm: Smart multi-agent robot task planning using large language models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12140–12147. IEEE, 2024.

[26] Weimin Xiong, Yifan Song, Qingxiu Dong, Bingchan Zhao, Feifan Song, Xun Wang, and Sujian Li. Mpo: Boosting llm agents with meta plan optimization. *arXiv preprint arXiv:2503.02682*, 2025.

[27] 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*, 2025.

[28] Jinyeon Kim, Cheolhong Min, Byeonghwi Kim, and Jonghyun Choi. Pre-emptive action revision by environmental feedback for embodied instruction following agents. In *8th Annual Conference on Robot Learning*, 2024.

[29] 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, 2024.

[30] 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.

[31] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[32] Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. *arXiv preprint arXiv:2311.01977*, 2023.

[33] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[34] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[35] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

[36] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024.

[37] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[38] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. *arXiv preprint arXiv:2410.07864*, 2024.

[39] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. *pi_0*: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[40] Karl Pertsch, Kyle Stachowicz, Brian Ichter, Danny Driess, Suraj Nair, Quan Vuong, Oier Mees, Chelsea Finn, and Sergey Levine. Fast: Efficient action tokenization for vision-language-action models. *arXiv preprint arXiv:2501.09747*, 2025.

[41] Figure AI. Helix: The next step in ai, 2025.

[42] Johan Bjorck, Fernando Castañeda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, et al. Gr00t n1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[43] Lei Han, Qingxu Zhu, Jiapeng Sheng, Chong Zhang, Tingguang Li, Yizheng Zhang, He Zhang, Yuzhen Liu, Cheng Zhou, Rui Zhao, et al. Lifelike agility and play in quadrupedal robots using reinforcement learning and generative pre-trained models. *Nature Machine Intelligence*, 6(7):787–798, 2024.

[44] Tairan He, Zhengyi Luo, Xialin He, Wenli Xiao, Chong Zhang, Weinan Zhang, Kris Kitani, Changliu Liu, and Guanya Shi. Omnih2o: Universal and dexterous human-to-humanoid whole-body teleoperation and learning. *arXiv preprint arXiv:2406.08858*, 2024.

[45] Huayi Wang, Zirui Wang, Junli Ren, Qingwei Ben, Tao Huang, Weinan Zhang, and Jiangmiao Pang. Beamdojo: Learning agile humanoid locomotion on sparse footholds. *arXiv preprint arXiv:2502.10363*, 2025.

[46] Mazeyu Ji, Xuanbin Peng, Fangchen Liu, Jialong Li, Ge Yang, Xuxin Cheng, and Xiaolong Wang. Exbody2: Advanced expressive humanoid whole-body control. *arXiv preprint arXiv:2412.13196*, 2024.

[47] Tao Huang, Junli Ren, Huayi Wang, Zirui Wang, Qingwei Ben, Muning Wen, Xiao Chen, Jianan Li, and Jiangmiao Pang. Learning humanoid standing-up control across diverse postures. *arXiv preprint arXiv:2502.08378*, 2025.

[48] Zixuan Chen, Mazeyu Ji, Xuxin Cheng, Xuanbin Peng, Xue Bin Peng, and Xiaolong Wang. Gmt: General motion tracking for humanoid whole-body control. *arXiv preprint arXiv:2506.14770*, 2025.

[49] Tairan He, Wenli Xiao, Toru Lin, Zhengyi Luo, Zhenjia Xu, Zhenyu Jiang, Jan Kautz, Changliu Liu, Guanya Shi, Xiaolong Wang, et al. Hover: Versatile neural whole-body controller for humanoid robots. *arXiv preprint arXiv:2410.21229*, 2024.

[50] Tairan He, Jiawei Gao, Wenli Xiao, Yuanhang Zhang, Zi Wang, Jiashun Wang, Zhengyi Luo, Guanqi He, Nikhil Sobanbab, Chaoyi Pan, et al. Asap: Aligning simulation and real-world physics for learning agile humanoid whole-body skills. *arXiv preprint arXiv:2502.01143*, 2025.

[51] Daniel Kahneman. *Thinking, fast and slow*. macmillan, 2011.

[52] Ye Tian, Baolin Peng, Linfeng Song, Lifeng Jin, Dian Yu, Lei Han, Haitao Mi, and Dong Yu. Toward self-improvement of llms via imagination, searching, and criticizing. *Advances in Neural Information Processing Systems*, 37:52723–52748, 2024.

[53] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[54] Harrison Chase. LangChain, October 2022.

[55] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In *Conference on Robot Learning*, pages 80–93. PMLR, 2023.

[56] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[57] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

[58] Marion Lepert, Jiaying Fang, and Jeannette Bohg. Phantom: Training robots without robots using only human videos. *arXiv preprint arXiv:2503.00779*, 2025.

[59] Shangchen Zhou, Chongyi Li, Kelvin CK Chan, and Chen Change Loy. Propainter: Improving propagation and transformer for video inpainting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10477–10486, 2023.

[60] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025.

[61] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.