OMNIEAR: BENCHMARKING AGENT REASONING IN EMBODIED TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models excel at abstract reasoning but their capacity for embodied agent reasoning remains largely unexplored. We present **OmniEAR**, a comprehensive framework for evaluating how language models reason about physical interactions, tool usage, and multi-agent coordination in embodied tasks. Unlike existing benchmarks that provide predefined tool sets or explicit collaboration directives, OmniEAR requires agents to dynamically acquire capabilities and autonomously determine coordination strategies based on task demands. Through text-based environment representation, we model continuous physical properties and complex spatial relationships across 1,500 scenarios spanning household and industrial domains. Our systematic evaluation reveals severe performance degradation when models must reason from constraints: while achieving 85-96% success with explicit instructions, performance drops to 56-85% for tool reasoning and 63-85% for implicit collaboration, with compound tasks showing over 50% failure rates. Surprisingly, complete environmental information degrades coordination performance, indicating models cannot filter task-relevant constraints. Fine-tuning improves single-agent tasks dramatically (0.6% to 76.3%) but yields minimal multi-agent gains (1.5% to 5.5%), exposing fundamental architectural limitations. These findings demonstrate that embodied reasoning poses fundamentally different challenges than current models can address, establishing OmniEAR as a rigorous benchmark for evaluating and advancing embodied AI systems.

1 INTRODUCTION

Large language models have achieved remarkable success in complex reasoning tasks(Brown et al., 2020; Wei et al., 2022), yet their ability to reason about embodied environments remains poorly understood. In embodied tasks, agents must understand how object properties affect what actions are possible, recognize when their capabilities are insufficient for a task, and determine when collaboration becomes necessary (Ahn et al., 2022; Wu et al., 2023). These reasoning abilities fundamentally differ from abstract problem-solving, as they require understanding the physical principles that govern real-world interactions.

Current evaluation approaches fail to capture this embodied reasoning complexity. Existing benchmarks model environments through discrete states like open/closed doors or picked/placed objects (Shridhar et al., 2020; Puig et al., 2018), overlooking continuous properties such as weight, temperature, or material composition that determine action feasibility. Tool usage evaluations typically provide fixed action sets (Chang et al., 2024; Huang et al., 2022), missing how agents should reason about capability gaps. Multi-agent benchmarks rely on explicit collaboration instructions or efficiency metrics (Kang et al., 2025; Zhang et al., 2024), rather than examining whether agents can recognize when tasks exceed individual abilities. This evaluation paradigm cannot assess understanding of embodied principles.

The core challenge is that real-world embodied reasoning emerges from understanding environmental realities and task requirements. When objects are too heavy for single agents, collaboration naturally becomes necessary. When tasks require manipulating materials beyond native capabilities, tools provide the solution. When spatial layouts limit individual reach, coordinated action enables

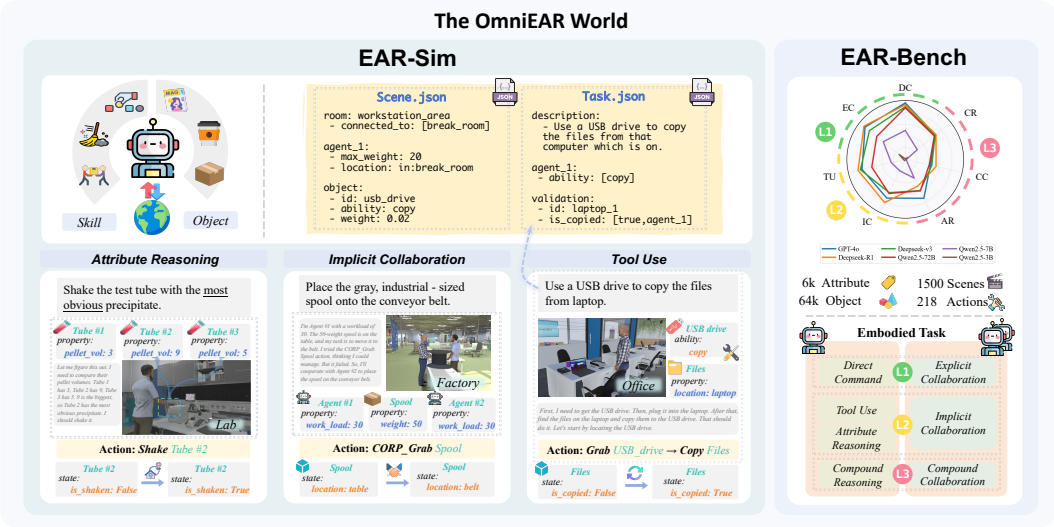


Figure 1: Overview of the OmniEAR framework comprising three integrated components: **EAR-Sim** (left) uses structured text representation to model environments with objects, agents, and spatial relationships, enabling dynamic tool-capability binding and physics-constrained collaboration; **EAR-Bench** (right) presents our comprehensive evaluation matrix spanning single-agent and multi-agent tasks across increasing cognitive complexity levels.

task completion (Zeng et al., 2022; Wang et al., 2023). Current benchmarks rely on static tool sets and explicit collaboration instructions, preventing assessment of how models reason about capability acquisition and coordination needs based on task requirements.

We introduce **OmniEAR**, a comprehensive framework for evaluating agent reasoning in embodied tasks. Our key insight is that embodied reasoning requires understanding how physical properties shape possible actions, how capability limitations necessitate tools, and how task demands drive collaboration.

By designing scenarios where agents must dynamically acquire capabilities and autonomously determine coordination strategies based on task requirements, we can assess whether models genuinely comprehend the principles governing embodied interactions.

OmniEAR employs text-based environment representation to efficiently model rich physical properties while enabling large-scale evaluation. The framework comprises three integrated components: **EAR-Sim** captures detailed object attributes and spatial relationships while supporting dynamic capability evolution through tool acquisition; an automated pipeline generates diverse scenarios where task solutions naturally depend on understanding embodied principles; and **EAR-Bench** provides systematic evaluation through 1,500 scenarios across household and industrial domains.

Our evaluation focuses on three core aspects of embodied reasoning. First, we assess how agents reason about object properties like weight, material, and temperature when determining feasible actions, requiring comparison and inference about continuous attributes. Second, we examine whether agents recognize when tasks demand capabilities beyond their current abilities and plan appropriate tool acquisition. Third, we evaluate autonomous coordination decisions, testing whether agents identify when task requirements exceed individual capacities without explicit collaboration instructions. These capabilities reflect fundamental aspects of embodied intelligence.

Systematic evaluation reveals fundamental gaps in current models' embodied reasoning abilities. While achieving 85-96% success on explicit instructions, performance degrades sharply when reasoning must emerge from physical constraints. Tool reasoning drops to 56-85% when models must infer capability needs, and implicit collaboration falls to 63-85% compared to 88-92% with explicit coordination. Compound tasks show the steepest decline, with failure rates exceeding 50%. Paradoxically, complete environmental information harms coordination performance, suggesting models cannot filter task-relevant from irrelevant constraints. Even reasoning-specialized models,

which excel at logical planning, fail to ground physical constraints effectively, demonstrating that current architectures lack the mechanisms necessary for autonomous embodied decision-making.

Our analysis uncovers important patterns in model capabilities. Smaller models cannot maintain the planning state necessary for multi-step reasoning about tools and coordination. Reasoning models excel at logical planning but struggle to ground abstract concepts in concrete physical properties. While supervised fine-tuning improves single-agent performance, these gains fail to transfer to multi-agent scenarios, suggesting that coordination reasoning requires architectural capabilities beyond current training approaches.

In summary, our contributions are:

- We present OmniEAR, a framework that evaluates embodied reasoning through scenarios requiring agents to understand how physical properties determine actions, capabilities, and coordination needs, addressing fundamental gaps in current evaluation methods.
- We develop EAR-Bench, a benchmark of 1,500 scenarios with continuous physical properties and dynamic capabilities, supported by EAR-Sim and an automated generation pipeline.
- We provide empirical evidence that current language models lack core embodied reasoning capabilities, with performance degrading over 60% when moving from explicit instructions to embodied reasoning, revealing critical requirements for advancing embodied AI.

2 Related Works

Prior embodied benchmarks have made significant contributions to task evaluation but differ fundamentally in their approach to physical reasoning and collaboration. While ALFRED (Shridhar et al., 2020) and BEHAVIOR-1K (Li et al., 2024a) provide extensive task coverage, they model physical states through discrete representations (e.g., binary door states, picked/placed objects) rather than continuous attributes necessary for reasoning about weight, temperature, or material properties. Tool usage evaluation spans from low-level manipulation in RoCo (Mandi et al., 2024) to high-level planning in PARTNR (Chang et al., 2024), yet both maintain static action spaces determined at initialization, preventing assessment of dynamic capability acquisition. Recent multi-agent benchmarks including TDW-MAT (Zhang et al., 2024) and EmbodiedBench (Yang et al., 2025) advance collaboration evaluation through load constraints and task allocation optimization, but rely on explicit task division instructions or efficiency-driven participation rather than collaboration that emerges from physical constraints. In contrast, OmniEAR introduces continuous property reasoning with 6,381 distinct attributes, dynamic tool-capability binding that expands action spaces during execution, and implicit collaboration where agents must autonomously recognize when tasks exceed individual capacities based on physical constraints, fundamentally shifting evaluation from instruction compliance to constraint-based reasoning. A comprehensive comparison with related work is provided in Appendix 6.1.

3 Framework

We present OmniEAR, a comprehensive framework for evaluating agent reasoning in embodied tasks. Our framework addresses the fundamental challenge of assessing whether language models understand embodied principles. We achieve this through three key design principles: (1) tasks must require reasoning about physical properties and constraints rather than following explicit instructions, (2) agent capabilities should dynamically evolve based on tool acquisition rather than remaining static, and (3) collaboration needs should emerge from task requirements rather than predetermined protocols.

3.1 Task Design and Formalization

Environment Representation. We formalize embodied environments as directed graphs $G_t = (V_t, E_t, A_t)$ that capture the essential structure of physical spaces. The node set V_t encompasses three entity types: spatial nodes representing rooms and areas, object nodes for interactive items, and agent nodes for autonomous entities. Each node maintains an attribute dictionary A_t storing

continuous physical properties such as weight, temperature, material composition, and geometric dimensions. The edge set E_t encodes spatial relationships through static containment relations (e.g., "in", "on") and dynamic proximity relations E_{near} that track which objects fall within an agent's interaction range. This graph representation enables efficient reasoning about spatial constraints while avoiding the computational overhead of continuous 3D simulation.

Task Formalization. Each evaluation task is defined as a tuple $\mathcal{T} = (S_{\text{init}}, I, G_{\text{goal}}, \mathcal{A}_{\text{task}})$, where S_{init} specifies the initial environment state, I provides the natural language instruction, G_{goal} defines success conditions through logical predicates, and $\mathcal{A}_{\text{task}}$ identifies participating agents. The evaluation objective is to assess whether agents can generate an action sequence $\Pi = (\pi_1, \ldots, \pi_T)$ that transforms the environment from S_{init} to a terminal state S_{final} satisfying all predicates in G_{goal}. This formalization captures both the planning and execution aspects of embodied reasoning.

3.2 Hierarchical Task Taxonomy

Our evaluation framework organizes tasks along two orthogonal dimensions: agent configuration (single vs. multi-agent) and cognitive complexity (L1: basic, L2: intermediate, L3: advanced). This structure enables systematic assessment of how reasoning capabilities scale with task demands.

Single-Agent Tasks. Single-agent scenarios ($|\mathcal{A}_{\text{task}}| = 1$) isolate individual reasoning capabilities across three complexity levels. At the basic level, ***Direct Command*** tasks require straightforward instruction following, such as "place cup#1 on table#1," establishing baseline comprehension abilities. Intermediate complexity introduces two parallel challenges: ***Attribute Reasoning*** tasks require comparing continuous properties to identify targets (e.g., "move the heaviest cup" requires solving $v^* = \arg\max_{v \in V_{\text{cups}}} A_t(v, \text{weight})$), while ***Tool Use*** tasks demand recognizing capability gaps and acquiring right tools. For instance, "clean the table" requires agents to identify that cleaning actions are unavailable in their base action set $\mathcal{A}_i$, locate cleaning tools, and execute $\texttt{grasp}(v_{\text{tool}})$ to dynamically expand their capabilities. Advanced ***Compound Reasoning*** tasks integrate multiple challenges, such as "clean the heaviest table," requiring simultaneous attribute comparison, tool acquisition, and multi-step planning.

Multi-Agent Tasks. Multi-agent scenarios ($|\mathcal{A}_{\text{task}}| > 1$) evaluate coordination capabilities through parallel complexity progression. Basic ***Explicit Collaboration*** tasks provide clear coordination directives, such as "Agent A and Agent B cooperate to open the heavy cabinet," testing fundamental synchronization abilities. Intermediate ***Implicit Collaboration*** removes explicit instructions, requiring agents to autonomously recognize when tasks exceed individual capabilities. For example, "move the dining table to the storage room" requires agents to infer that $A_t(v_{\text{table}}, \text{weight}) > C_{\max}(i)$ for any individual agent i, necessitating collaborative effort. Advanced ***Compound Collaboration*** combines all elements, such as "cooperatively repair the malfunctioning television," demanding tool acquisition, capability assessment, and coordinated execution.

3.3 EAR-Sim: Efficient Environment Simulation

State Representation and Updates. EAR-Sim employs text-based environment modeling to achieve efficient simulation at scale. The graph structure G_t maintains spatial relationships through topological connections rather than continuous coordinates, eliminating expensive collision detection while preserving essential spatial constraints. State updates follow an incremental approach where actions modify only directly affected nodes and edges. For instance, when an agent executes $\texttt{GOTO}(\text{table})$, the system updates only the relevant proximity relations in E_{near} rather than recomputing global spatial relationships.

Dynamic Capability Management. A key innovation in EAR-Sim is the dynamic tool-capability binding system. Agent actions are partitioned into basic actions (movement, grasping, opening) available to all agents, and tool-dependent actions (cleaning, heating, repairing) that require specific tools. Each tool object maintains a `capability` attribute specifying which actions it enables. When an agent grasps a tool, the system dynamically binds the associated capabilities to the agent's action set. Upon releasing the tool, these capabilities are automatically unbound. This mechanism

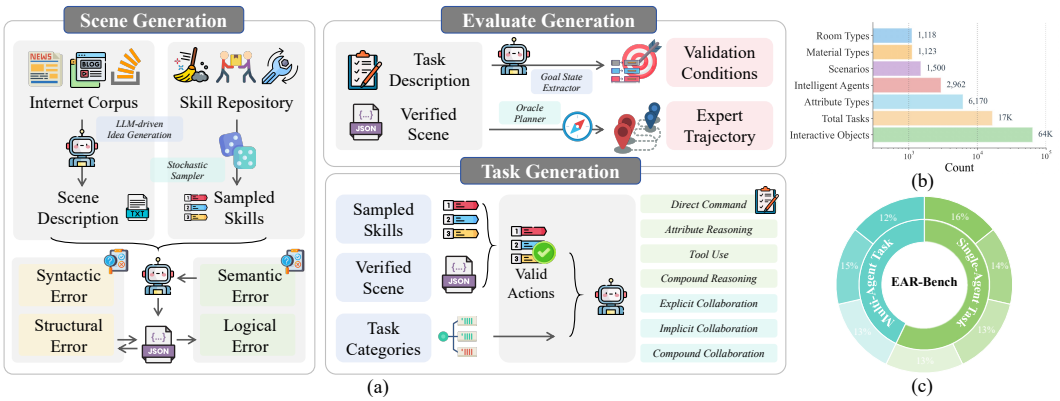


Figure 2: OmniEAR automated benchmark generation and evaluation framework. (a) Four-stage generation pipeline combining LLMs with rule-based validation: scene generation from internet corpus, task generation with skill sampling, evaluation logic extraction, and expert trajectory generation with human validation. (b) EAR-Bench statistics: 1,500 scenarios, 64K objects, 6K attribute types, spanning diverse domains and material compositions. (c) Balanced task distribution across seven categories spanning single-agent (Direct Command, Tool Use, Attribute Reasoning, Compound Reasoning) and multi-agent tasks (Explicit/Implicit/Compound Collaboration).

enables realistic modeling of how agents extend their abilities through tool use, moving beyond the static action spaces of existing benchmarks.

Emergent Collaboration. EAR-Sim supports collaboration that emerges from physical constraints rather than explicit programming. When agents attempt actions on objects whose properties exceed individual capabilities, the system enables collaboration request mechanisms. For instance, if an agent attempts to move an object where $A_t(v, \text{weight}) > C_{\max}(\text{agent})$, it can initiate collaboration by identifying suitable partners and coordinating joint actions. The system validates preconditions for all participating agents and maintains consistency throughout collaborative execution, ensuring realistic multi-agent interactions.

3.4 Automated Benchmark Generation

Generation Pipeline. Creating diverse, physically consistent scenarios at scale requires careful orchestration of neural generation and symbolic validation. As shown in 2, our pipeline operates in four stages, each combining the creative capabilities of large language models with rule-based consistency checking. This hybrid approach enables generating thousands of unique scenarios while maintaining physical realism and task solvability.

Scene and Task Generation. Scene generation begins with semantic seeds extracted from diverse text sources(Li et al., 2024b), which guide a neural generator g_{scene} in creating structured environment descriptions. The generator, implemented using high-temperature language models for diversity, produces initial scenes S_0 containing objects, spatial layouts, and agent configurations. Task generation follows a two-stage process: first, an environment analyzer C_{env} extracts feasible actions based on the scene structure, then a task generator g_{task} creates instructions anchored in physical possibilities. This grounding prevents generation of impossible tasks while maintaining creative diversity.

Evaluation Logic and Trajectories. For each generated task, we automatically derive evaluation criteria by parsing the instruction and scene to extract minimal state changes required for success. This produces a goal predicate set G_{goal} that serves as an objective success measure. Expert trajectories are generated using oracle agents with complete environmental knowledge, creating high-quality demonstrations for each task. These trajectories undergo filtering to remove suboptimal sequences, providing ideal solutions for comparison and learning.

Quality Assurance. As shown in Figure 3, generated scenarios undergo a three-stage validation pipeline. Automated validators check structural consistency and physical feasibility, correcting 87.6% of detected errors. Trained human annotators then verify scenarios against explicit criteria for instruction clarity, feasibility, and solvability (Cohen's $\kappa = 0.84$). Finally, oracle agents validate expert trajectories through simulation replay. This rigorous process yielded 1,500 validated scenarios from 2,100 initial candidates (71.4% acceptance rate). Detailed procedures, annotation interface, and quality threshold examples are provided in Appendix 6.3.

3.5 Benchmark Statistics and Coverage

EAR-Bench encompasses 1,500 scenarios across 11 domains including laboratory (39%), office (19%), industrial (12%), and medical environments, containing 64,057 interactive objects with rich physical properties. The dataset maintains careful balance across our task taxonomy: 65% single-agent tasks spanning all complexity levels, and 35% multi-agent tasks with emphasis on implicit collaboration scenarios that require genuine reasoning about coordination needs. With 6,381 distinct property types and 214 action types, EAR-Bench provides comprehensive coverage of embodied reasoning challenges while maintaining tractable evaluation scope. Detailed statistics are provided in Appendix 6.4.

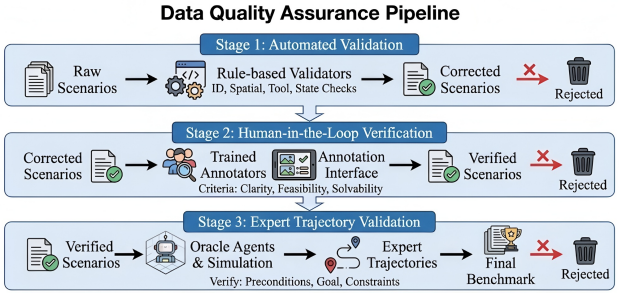


Figure 3: Overview of the Data Quality Assurance Pipeline. The pipeline consists of three sequential stages: (1) Automated Validation for rule-based error correction, (2) Human-in-the-Loop Verification for semantic and physical feasibility checks, and (3) Expert Trajectory Validation to ensure genuine solvability via oracle agents.

4 Experiments

We systematically evaluate current LLMs on EAR-Bench to assess their physical reasoning capabilities in embodied tasks. Our experiments examine: (1) How performance degrades when models must dynamically acquire tools and determine coordination requirements from task contexts, (2) Whether model scale and architectural choices affect constraint-based reasoning capabilities, and (3) How environmental information presentation and training approaches impact autonomous decision-making in embodied scenarios.

4.1 Experimental Setup

Model Selection. We evaluate nine representative models spanning three architectural paradigms. Closed-source models include GPT-4o (Hurst et al., 2024) and Gemini-2.5-Flash (Comanici et al., 2025), representing current commercial state-of-the-art. Open-source foundation models cover a wide parameter range: Deepseek-V3 (Liu et al., 2024) at 671B parameters, the Qwen2.5 series (Team, 2024) at 3B, 7B, and 72B parameters, and Llama3.1-8B(Touvron et al., 2023). This selection enables analysis of how model scale affects embodied reasoning. We also include reasoning-specialized models: Deepseek-R1 (Guo et al., 2025) and QwQ-32B(Li et al., 2024b), which employ explicit chain-of-thought reasoning during inference.

Evaluation Protocol. All models undergo identical evaluation to ensure fair comparison. We implement partial observability where agents must explore environments to discover object locations and properties, reflecting realistic deployment conditions. Each model completes 2,800 test scenarios across seven task categories with three independent runs for statistical reliability. We standardize prompts, environment descriptions, and action vocabularies across all models, with tool-dependent actions dynamically enabled based on context. This design ensures performance differences reflect reasoning capabilities rather than implementation artifacts. Detailed experimental configurations are provided in Appendix 6.8.

Model	Single-Agent Tasks								Multi-Agent Tasks					
	Direct Command		Tool Use		Attribute Reasoning		Compound Reasoning		Explicit Collab.		Implicit Collab.		Compound Collab.	
	SR	Step	SR	Step	SR	Step	SR	Step	SR	Step	SR	Step	SR	Step
Closed-source Models														
GPT-4o	**96.6**	12.9	80.0	13.6	**77.8**	12.3	**69.2**	14.5	**90.0**	13.9	77.5	14.4	32.0	22.9
Gemini-2.5-Flash	90.5	11.0	**82.3**	16.5	56.3	17.5	59.4	20.0	88.5	8.4	**85.5**	7.1	**40.5**	16.2
Reasoning-specialized Models														
Deepseek-R1	**94.1**	10.3	**85.8**	14.1	41.9	12.2	**70.6**	16.2	**92.0**	7.4	**84.5**	9.6	**48.5**	12.5
QwQ-32B	85.2	10.3	73.4	13.0	**44.9**	11.0	54.1	13.6	88.0	8.5	84.0	8.3	36.5	19.0
Open-source Foundation Models														
Deepseek-V3	**91.1**	11.2	**82.3**	15.1	56.3	10.3	**67.1**	16.0	**82.0**	9.4	63.0	9.7	**36.0**	20.2
Qwen2.5-72B	89.7	14.7	56.4	21.7	**57.4**	17.2	66.7	21.1	56.0	24.1	**65.4**	15.6	28.6	29.5
Llama3.1-8B	24.9	34.4	8.3	34.6	9.9	34.8	12.4	34.3	4.0	3.5	1.5	2.1	0.0	3.4
Qwen2.5-7B	40.2	24.1	15.4	31.7	22.2	26.6	16.5	30.5	38.5	25.0	13.5	24.1	1.0	27.2
Qwen2.5-3B	0.6	30.5	1.8	31.3	0.6	34.0	2.9	32.9	8.5	20.4	1.5	16.3	0.5	16.8
+ SFT	76.3	15.4	45.0	24.7	33.5	22.8	36.5	24.7	22.5	29.2	5.5	28.3	1.0	27.1

Table 1: Performance across task categories. Success Rate (SR) measures task completion percentage, Step Count indicates average actions for successful completion. Bold indicates best in category, underline shows overall best.

Fine-tuning Configuration. To assess whether supervised learning can address reasoning limitations, we fine-tune Qwen2.5-3B on expert trajectories. We collect 1,942 successful demonstrations from Qwen2.5-72B with complete environmental access, filtering for optimal action sequences. The resulting 20,346 instruction-action pairs train the model using standard causal language modeling objectives, testing whether smaller models can learn embodied reasoning patterns from larger models. Complete hyperparameters are listed in Appendix 6.6.

Deployment Configurations. We evaluate models in two configurations. Single-agent scenarios test individual reasoning capabilities without collaborative complexity. Multi-agent scenarios employ centralized coordination where one model controls all agents with complete state visibility, isolating collaborative reasoning from communication challenges. This design choice allows us to assess pure multi-agent reasoning capabilities without confounding factors from limited observability or communication protocols.

4.2 Main Results

Table 1 presents comprehensive evaluation results across our task hierarchy. The results reveal systematic performance patterns that validate our framework design and expose fundamental limitations in current models.

Task Complexity Hierarchy. Figure 4 reveals systematic performance degradation across our task hierarchy, with success rates declining from 85.2-96.6% on Direct Commands to 32.0-48.5% on Compound Collaboration tasks. This consistent pattern confirms that performance differences reflect reasoning complexity rather than task difficulty alone. Tool Use (73.4-85.8%) requires recognizing capability gaps from context, while Attribute Reasoning (41.9-77.8%) demands grounding language in physical properties. Both involve inferring requirements from environmental constraints rather than following explicit instructions. Notably, Explicit Collaboration outperforms several single-agent tasks, indicating that reasoning about physical constraints poses greater challenges than multi-agent coordination when guidance is provided. The severe performance drop in compound tasks demonstrates that current models cannot integrate multiple constraints simultaneously, supporting our framework's focus on autonomous inference from physical context as the key determinant of embodied reasoning difficulty.

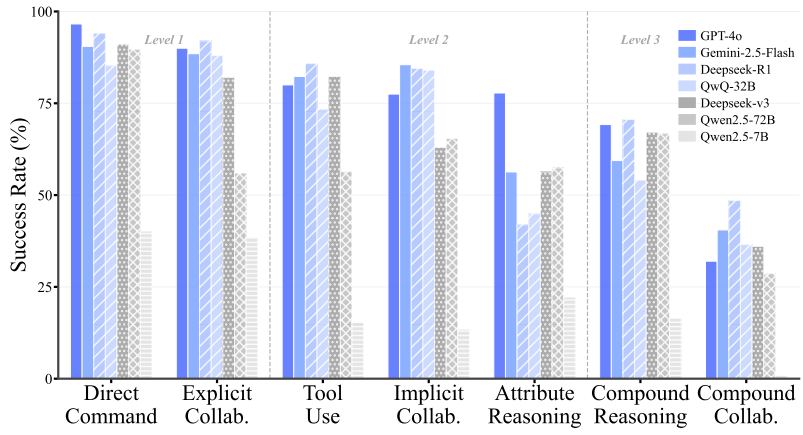
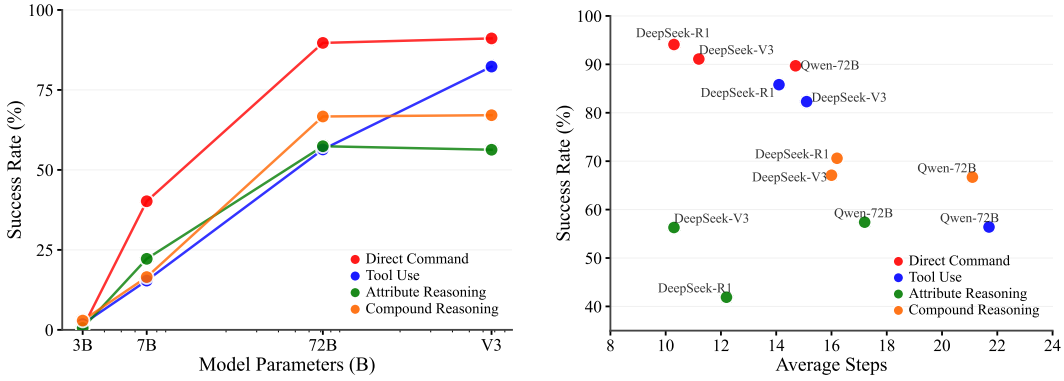


Figure 4: Performance comparison across task categories demonstrating systematic difficulty hierarchy and distinct model performance tiers.



(a) Performance vs. model parameters.

(b) Success rate vs. execution steps.

Figure 5: Scaling patterns reveal distinct thresholds for embodied reasoning capabilities. (a) Direct Command and Tool Use scale sharply with parameters while Attribute/Compound Reasoning plateau early. (b) Reasoning-specialized models achieve higher success through longer execution paths.

Model Scale and Reasoning Capabilities. Figure 5a reveals distinct scaling patterns across task types. While Direct Command performance improves sharply with model size (from near-zero at 3B to over 90% at 72B), tasks requiring physical constraint reasoning show more complex relationships. Tool Use exhibits similar steep scaling, suggesting that maintaining multi-step plans for capability acquisition correlates strongly with model capacity. However, Attribute Reasoning and Compound Reasoning plateau earlier, with diminishing returns beyond 72B parameters. This differential scaling indicates that raw parameter count enables better execution and planning but does not necessarily improve understanding of physical properties.

Table 1 provides further evidence distinguishing execution capability from genuine reasoning. Reasoning-specialized models like Deepseek-R1 achieve the highest performance on Compound Collaboration (48.5%) despite lower scores on Attribute Reasoning (41.9%) compared to GPT-4o (77.8%). This performance inversion suggests these models excel at explicit logical planning but struggle with grounding abstract properties in physical contexts. The success rate versus step count trade-off in Figure 5b reinforces this interpretation: reasoning models achieve higher success through longer, more deliberate execution paths rather than efficient understanding of constraints. Fine-tuning results provide the clearest evidence that current models lack true embodied reasoning: while Qwen2.5-3B improves dramatically on single-agent tasks through imitation (0.6% to 76.3%), multi-agent performance remains negligible (1.5% to 5.5%), indicating that learned

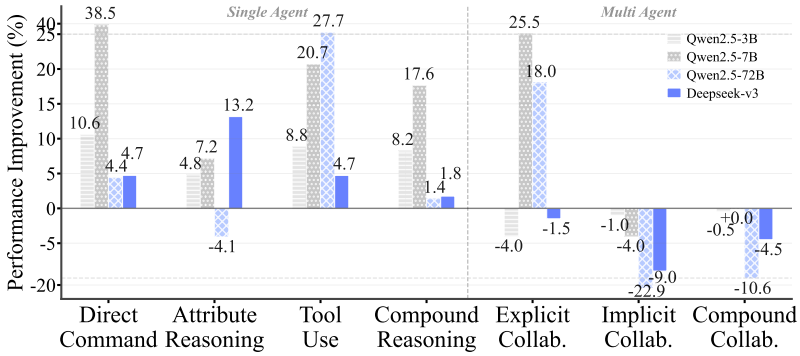


Figure 6: Performance changes with World Graph enhancement. Tool Use and Attribute Reasoning benefit substantially, while Implicit Collaboration shows degradation, suggesting information overload effects.

behaviors cannot generalize to scenarios requiring autonomous assessment of physical constraints and coordination needs.

4.3 Detailed Analysis

We conduct analyses to understand the factors driving model performance and identify specific capability bottlenecks.

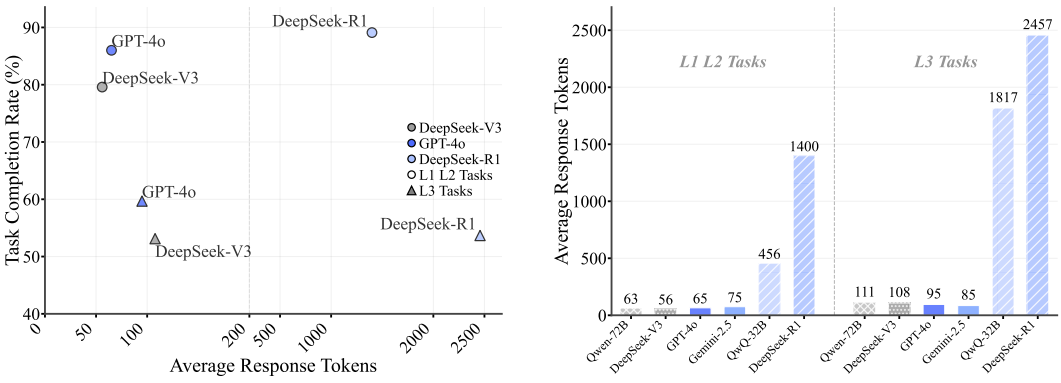
Environmental Representation Impact. Table 2 and Figure 6 reveal task-specific effects of structured environmental knowledge. Tool Use benefits most significantly (up to 27.7% improvement), as World Graph transforms spatial search into direct tool selection. Smaller models gain more than larger ones, suggesting that full environmental knowledge compensates for limited working memory. Conversely, Implicit Collaboration consistently drops with World Graph across all model scales. This counterintuitive pattern indicates that exploration-based discovery helps models focus on task-relevant constraints, while complete information introduces distraction. The divergent effects across task types demonstrate that optimal information presentation depends on reasoning requirements, not information quantity.

Task	3B		7B		72B		671B	
	w/o	w/	w/o	w/	w/o	w/	w/o	w/
Direct Cmd	0.6	11.2	40.2	78.7	89.7	94.1	91.1	95.9
Tool Use	1.8	10.7	15.4	36.1	56.4	84.0	82.3	87.0
Attr. Reas.	0.6	5.4	22.2	29.3	57.4	53.3	56.3	69.5
Comp. Reas.	2.9	11.2	16.5	34.1	64.5	65.9	67.1	68.8
Expl. Coll.	8.5	4.5	38.5	64.0	62.5	80.5	82.0	80.5
Impl. Coll.	1.5	0.5	13.5	9.5	65.4	42.5	63.0	54.0
Comp. Coll.	0.5	0.0	1.0	1.0	28.6	18.0	36.0	31.5

Table 2: Success rates (%) with and without World Graph enhancement across model scales, revealing task-specific gains and unexpected drop in implicit collaboration.

Computational Efficiency Trade-offs. Figure 7a identifies three efficiency regimes with distinct cost-performance profiles. Foundation models achieve moderate performance with minimal tokens (456-1400), while commercial models trade higher token usage (1817-2457) for improved success rates. Reasoning models consume up to 12,000 tokens but excel on complex tasks. The efficiency frontier shifts dramatically between single and multi-agent scenarios: Gemini-2.5-Flash optimizes single-agent efficiency, but Deepseek-R1 becomes necessary for multi-agent tasks despite 75% higher costs. This shift reflects the irreducible computational complexity of modeling multiple agent states and coordination protocols, suggesting no universal optimization exists across task types.

Execution Efficiency Analysis. Figure 8 compares model solutions to expert demonstrations via Relative Step Ratios (RSR = $L_{\text{expert}}/L_{\text{model}}$). Single-agent tasks show consistent moderate efficiency (median RSR 0.40-0.55), while multi-agent tasks exhibit both lower efficiency and higher variance, reflecting uncertainty in coordination timing and strategy selection. Compound Collaboration



(a) Resource utilization vs. success rate.

(b) Token consumption by model type.

Figure 7: Reasoning-specialized models achieve higher performance through increased computational overhead. (a) Efficiency-performance trade-offs across model architectures. (b) Token consumption patterns revealing computational costs of reasoning approaches.

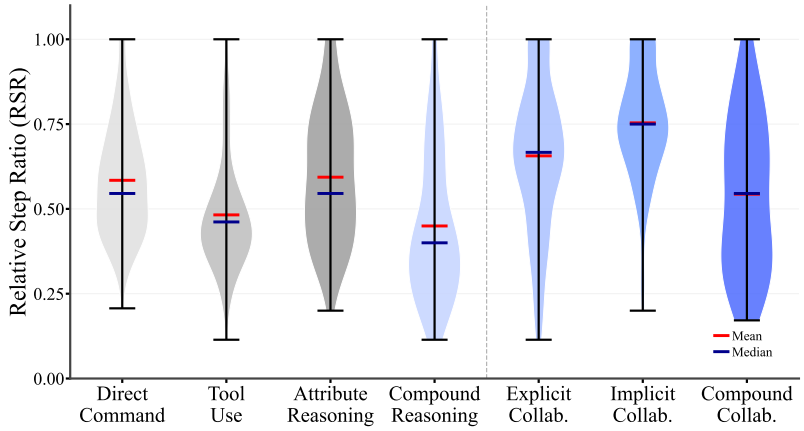


Figure 8: Relative Step Ratio distributions showing execution efficiency compared to expert trajectories. Multi-agent tasks show both lower efficiency and higher variance than single-agent tasks.

reveals a striking bimodal distribution: models either adopt simple sequential execution or attempt complex parallel coordination, with no successful middle strategies. This polarization suggests current models lack adaptive coordination mechanisms, defaulting to extreme approaches rather than selecting strategies based on task constraints.

5 Conclusion

We presented OmniEAR, a benchmark for evaluating embodied agent reasoning through 1,500 scenarios requiring inference from physical constraints. Our evaluation reveals that current models show severe performance degradation when moving from explicit instructions to constraint-based reasoning, with performance dropping from over 85% to below 65% across tool usage and coordination tasks. We identify critical parameter thresholds for maintaining multi-step plans, paradoxical effects of environmental information on coordination, and the inability of fine-tuning to address multi-agent reasoning gaps. Results demonstrate that embodied reasoning requires fundamentally different computational mechanisms than those underlying current language models. OmniEAR provides systematic diagnostics of these limitations and a rigorous platform for developing next-generation embodied AI systems. We discuss broader implications and future research directions in Appendix 6.7.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022. URL https://arxiv.org/abs/2204.01691.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, Siddharth Patki, Ishita Prasad, Xavier Puig, Akshara Rai, Ram Ramrakhya, Daniel Tran, Joanne Truong, John M. Turner, Eric Undersander, and Tsung-Yen Yang. Partnr: A benchmark for planning and reasoning in embodied multi-agent tasks, 2024. URL https://arxiv.org/abs/2411.00081.

Sadia Sultana Chowa, Riasad Alvi, Subhey Sadi Rahman, Md Abdur Rahman, Mohaimenul Azam Khan Raiaan, Md Rafiqul Islam, Mukhtar Hussain, and Sami Azam. From language to action: A review of large language models as autonomous agents and tool users, 2025. URL https://arxiv.org/abs/2508.17281.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, 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.

Pascale Fung, Yoram Bachrach, Asli Celikyilmaz, Kamalika Chaudhuri, Delong Chen, Willy Chung, Emmanuel Dupoux, Hongyu Gong, Hervé Jégou, Alessandro Lazaric, Arjun Majumdar, Andrea Madotto, Franziska Meier, Florian Metze, Louis-Philippe Morency, Théo Moutakanni, Juan Pino, Basile Terver, Joseph Tighe, Paden Tomasello, and Jitendra Malik. Embodied ai agents: Modeling the world, 2025. URL https://arxiv.org/abs/2506.22355.

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.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Li Kang, Xiufeng Song, Heng Zhou, Yiran Qin, Jie Yang, Xiaohong Liu, Philip Torr, Lei Bai, and Zhenfei Yin. Viki-r: Coordinating embodied multi-agent cooperation via reinforcement learning, 2025. URL https://arxiv.org/abs/2506.09049.

Chengshu Li, Fei Xia, Roberto Martín-Martín, Michael Lingelbach, Sanjana Srivastava, Bokui Shen, Kent Vainio, Cem Gokmen, Gokul Dharan, Tanish Jain, et al. igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. *arXiv preprint arXiv:2108.03272*, 2021.

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

Dingming Li, Hongxing Li, Zixuan Wang, Yuchen Yan, Hang Zhang, Siqi Chen, Guiyang Hou, Shengpei Jiang, Wenqi Zhang, Yongliang Shen, Weiming Lu, and Yueting Zhuang. Viewspatial-bench: Evaluating multi-perspective spatial localization in vision-language models, 2025. URL https://arxiv.org/abs/2505.21500.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Scott Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024b.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Zhao Mandi, Shreeya Jain, and Shuran Song. Roco: Dialectic multi-robot collaboration with large language models. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 286–299. IEEE, 2024.

Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs, 2018. URL https://arxiv.org/abs/1806.07011.

Xavier Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Tsung-Yen Yang, Ruslan Partsey, Ruta Desai, Alexander William Clegg, Michal Hlavac, So Yeon Min, et al. Habitat 3.0: A co-habitat for humans, avatars and robots. *arXiv preprint arXiv:2310.13724*, 2023.

Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *International conference on machine learning*, pp. 4218–4227. PMLR, 2018.

Yongliang Shen, Kaitao Song, Xu Tan, Wenqi Zhang, Kan Ren, Siyu Yuan, Weiming Lu, Dongsheng Li, and Yueting Zhuang. Taskbench: Benchmarking large language models for task automation. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 4540–4574. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/085185ea97db31ae6dcac7497616fd3e-Paper-Datasets_and_Benchmarks_Track.pdf.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks, 2020. URL https://arxiv.org/abs/1912.01734.

Nan Sun, Chengming Shi, et al. [aml] interactgen: Enhancing human-involved embodied task reasoning through llm-based multi-agent collaboration. 2024.

Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023. URL https://arxiv.org/abs/2305.16291.

Yuntao Wang, Yanghe Pan, Quan Zhao, Yi Deng, Zhou Su, Linkang Du, and Tom H Luan. Large model agents: State-of-the-art, cooperation paradigms, security and privacy, and future trends. *arXiv e-prints*, pp. arXiv–2409, 2024.

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.

Zhenyu Wu, Ziwei Wang, Xiuwei Xu, Jiwen Lu, and Haibin Yan. Embodied task planning with large language models, 2023. URL https://arxiv.org/abs/2307.01848.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. Embodiedbench: Comprehensive benchmarking multi-modal large language models for vision-driven embodied agents. *arXiv preprint arXiv:2502.09560*, 2025.

Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language, 2022. URL https://arxiv.org/abs/2204.00598.

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, 2024. URL https://arxiv.org/abs/2307.02485.

6 APPENDIX

Dataset	Scenes	Domain	Task Types	Actions	Action Space	Collab.	Auto Gen.
ALFRED	120	House	D	7	Static	—	×
PARTNR	60	House	D	11	Static	Effic.	✓
BEHAVIOR-1K	50	Diverse	D,T	6	Static	—	×
WAH	7	House	D	10	Static	Effic.	×
TDW-MAT	6	House	D,E	7	Static	Effic.	×
C-WAH	6	House	D,E	7	Static	Effic.	×
Overcooked	5	Kitchen	E,I,C	6	Static	Effic.	×
OmniEAR	**1.5K**	**Diverse**	**D,A,T,R,E,I,C**	**218**	**Dynamic**	**Phys.**	✓

Table 3: Comparison of embodied AI datasets and benchmarks. Task types: D (Direct Command), A (Attribute Reasoning), T (Tool Use), R (Compound Reasoning), E (Explicit Collaboration), I (Implicit Collaboration), C (Compound Collaboration). Actions: number of available action types. Collab.: collaboration mechanism (Effic. = efficiency-based, Phys. = physical necessity-driven). Auto Gen.: automated task generation capability. Our framework uniquely combines comprehensive task coverage, dynamic action spaces, physical necessity-driven collaboration, and scalable automated generation.

6.1 RELATED WORK

Embodied Intelligence Benchmarks The embodied intelligence evaluation landscape has established diverse benchmark frameworks spanning navigation to complex manipulation tasks(Puig et al., 2023; Li et al., 2021). ALFRED (Shridhar et al., 2020) provides foundational standards for instruction-following task evaluation, while BEHAVIOR-1K (Li et al., 2024a) extends coverage to 1,000 daily activity scenarios. These benchmarks effectively assess task execution capabilities, yet physical property modeling predominantly employs discrete state representations, such as binary door operations and object pickup/placement, with limited requirements for reasoning about continuous attributes including weight, hardness, and temperature. Our framework addresses this limitation by introducing continuous physical property reasoning tasks that require agents to compare object attributes and make decisions based on physical constraints.

Embodied Tool Use Tool usage evaluation in embodied AI exhibits stratified characteristics across different complexity levels. RoCo (Mandi et al., 2024) focuses on low-level manipulation skills such as grasping precision, while high-level benchmarks like PARTNR (Chang et al., 2024) adopt predefined tool configurations with agent action spaces fixed at task initialization. This design effectively simplifies evaluation complexity but presents limitations in assessing dynamic tool reasoning capabilities based on task requirements. Current approaches typically provide static tool sets (Chowa et al., 2025; Fung et al., 2025), preventing evaluation of how agents should reason about capability gaps and tool acquisition needs. Our framework introduces dynamic tool acquisition mechanisms, requiring agents to autonomously infer tool requirements and expand their action spaces based on task demands, thereby supplementing existing evaluation dimensions.

Multi-Agent Collaboration Multi-agent embodied intelligence evaluation has emerged as a significant research direction, with related work achieving valuable progress in collaboration modeling(Sun et al., 2024; Wang et al., 2024). PARTNR evaluates multi-agent planning capabilities through heterogeneous task design, TDW-MAT (Zhang et al., 2024) creates collaborative scenarios using load capacity constraints, and EmbodiedBench (Yang et al., 2025) focuses on task allocation and execution optimization. Existing approaches primarily model collaboration requirements through two pathways: explicit collaboration instructions that clearly specify inter-agent task division, and efficiency optimization that drives multi-agent participation to enhance task completion speed. However, real-world collaboration decisions often stem from physical constraints rather than external instructions or efficiency considerations. Our framework employs implicit collaboration design requiring agents to autonomously assess whether tasks exceed single-agent capability ranges based on physical constraints and determine collaboration strategies accordingly, transforming collaboration judgment from external instructions to constraint-driven internal reasoning processes.

6.2 Task Category Examples

Figure 9 illustrates our hierarchical task taxonomy with representative examples, organized by agent configuration (single-agent vs. multi-agent) and cognitive complexity level (L1-L3). The figure demonstrates the progression from basic instruction following to advanced reasoning requiring dynamic capability acquisition and autonomous coordination. Table 4 provides detailed descriptions of the key reasoning requirements for each task category.

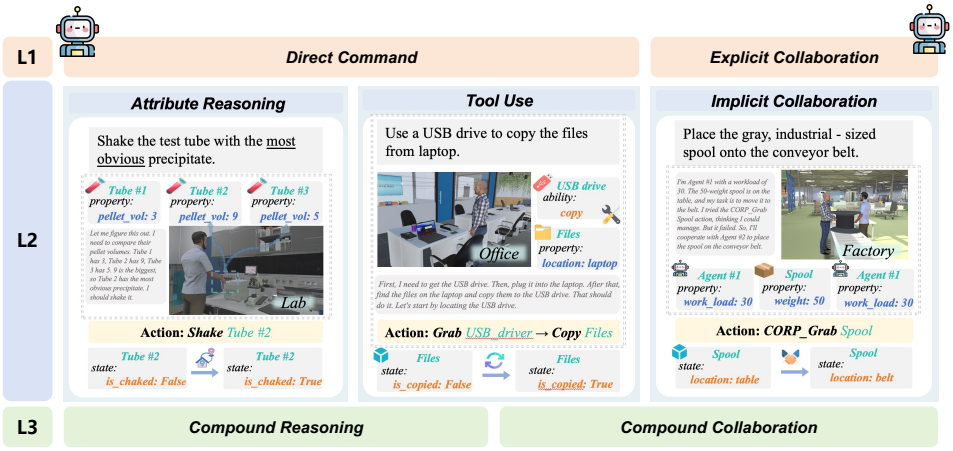


Figure 9: Representative examples for each task category in EAR-Bench. Left column shows single-agent tasks progressing from Direct Command (L1) through Attribute Reasoning and Tool Use (L2) to Compound Reasoning (L3). Right column shows multi-agent tasks from Explicit Collaboration (L1) through Implicit Collaboration (L2) to Compound Collaboration (L3). Each example includes the task instruction, relevant object properties, agent reasoning process, and resulting state transitions.

Level	Task Type	Example Task	Key Reasoning Requirements
Single-Agent Tasks			
L1	Direct Command	Place the red cup on the kitchen table.	Basic object manipulation and spatial understanding through straightforward instruction following.
L2	Tool Use	Clean the dirty table in the living room.	Recognizing capability gaps, locating appropriate tools, and dynamically expanding action capabilities through tool acquisition.
L2	Attribute Reasoning	Move the heaviest box to the storage room.	Comparing continuous physical properties across multiple objects to identify correct manipulation targets.
L3	Compound Reasoning	Clean the heaviest table in the room.	Integrating attribute comparison, tool acquisition, and multi-step planning simultaneously.
Multi-Agent Tasks			
L1	Explicit Collab.	Agent A and Agent B cooperate to move the heavy dining table.	Fundamental multi-agent synchronization with explicit coordination directives provided.
L2	Implicit Collab.	Move the piano to the music room.	Autonomously recognizing when tasks exceed individual capabilities without explicit coordination instructions.
L3	Compound Collab.	Cooperatively repair the malfunctioning television.	Combining tool acquisition, capability assessment, and coordinated execution with autonomous collaboration recognition.

Table 4: Representative examples for each task category in EAR-Bench. Tasks span single-agent scenarios (Direct Command, Tool Use, Attribute Reasoning, Compound Reasoning) and multi-agent scenarios (Explicit, Implicit, and Compound Collaboration) across three complexity levels.

6.3 Data Quality Assurance and Validation

A key challenge in LLM-based benchmark generation is ensuring that generated scenarios are physically consistent, unambiguous, and genuinely solvable (Li et al., 2025; Shen et al., 2024). To address this challenge and ensure benchmark reliability at scale, we implement a rigorous three-stage validation pipeline that combines automated error correction, systematic human-in-the-loop verification, and simulation-based trajectory validation, as illustrated in Figure 3. This hybrid approach balances scalability with quality control, achieving validation standards comparable to manually curated benchmarks while enabling generation at scale.

6.3.1 Stage 1: Automated Validation

The first stage addresses systematic errors that can be detected and corrected programmatically, reducing the burden on human annotators while ensuring basic structural integrity.

Error Analysis and Correction. Systematic analysis of 2,100 initially generated scenarios identified five primary error categories, summarized in Table 5. We developed specialized rule-based validators to automatically detect and correct each error type: ID mismatches are resolved via semantic similarity matching (cosine threshold 0.7), spatial violations are corrected via dimension checking and constraint propagation, tool-capability conflicts are resolved via attribute cross-referencing against our capability ontology, and state inconsistencies are handled via priority-based resolution that preserves task-critical states. This automated pipeline successfully corrected 87.6% of detected errors. The remaining 12.4% of scenarios—primarily those with complex logical inconsistencies requiring semantic understanding—were rejected, yielding 1,804 candidates for human review.

Error Type	**Frequency**	**Auto-Correctable**
ID Mismatch	32.4%	91%
Invalid Spatial Relations	28.7%	85%
Tool-Capability Conflict	23.1%	89%
State Inconsistency	10.3%	94%
Other Logical Errors	5.5%	62%
Overall	**100%**	**87.6%**

Table 5: Distribution of LLM generation errors and automatic correction rates by category.

6.3.2 Stage 2: Human-in-the-Loop Verification

While automated validation effectively handles systematic errors, subtle semantic ambiguities, non-obvious unsolvability, and nuanced physical inconsistencies require human expertise. We therefore implement rigorous human-in-the-loop verification with carefully selected and trained annotators, standardized evaluation criteria, and quantitative quality control measures.

Annotator Selection and Training. Five annotators were recruited from graduate programs in robotics and embodied AI, each with at least two years of research experience in related domains. All annotators completed a standardized 4-hour training program comprising three components: conceptual training on embodied reasoning principles and physical constraints (1.5 hours), hands-on system familiarization with environment representation and the validation interface (1 hour), and calibration exercises on 20 expert-validated scenarios with group discussion and feedback (1.5 hours). The calibration scenarios were independently validated by two senior researchers to establish reliable ground truth labels. Annotators were required to achieve Cohen's $\kappa \geq 0.75$ against these gold labels before beginning independent annotation; those falling below this threshold received additional training until reaching the required level. During the main annotation phase, each annotator reviewed approximately 720 scenarios (with systematic overlap for agreement computation), averaging 3-4 minutes per scenario to ensure careful evaluation.

Annotation Interface. To ensure consistent and reliable human evaluation, we developed a standardized annotation interface illustrated in Figure 10. The interface presents three integrated components: (1) a structured scene representation panel displaying room layouts, object properties (weight, material, dimensions), spatial relationships, and agent configurations in a clear hierarchical format; (2) a task instruction panel with automatically highlighted key elements including target objects, required actions, and relevant constraints; and (3) an evaluation panel presenting the three assessment criteria as binary selections along with a mandatory text field for rejection justification. This standardized interface ensures consistent evaluation conditions across all annotators and scenarios, minimizing variance arising from interface interpretation differences and enabling systematic collection of rejection rationales for error analysis.

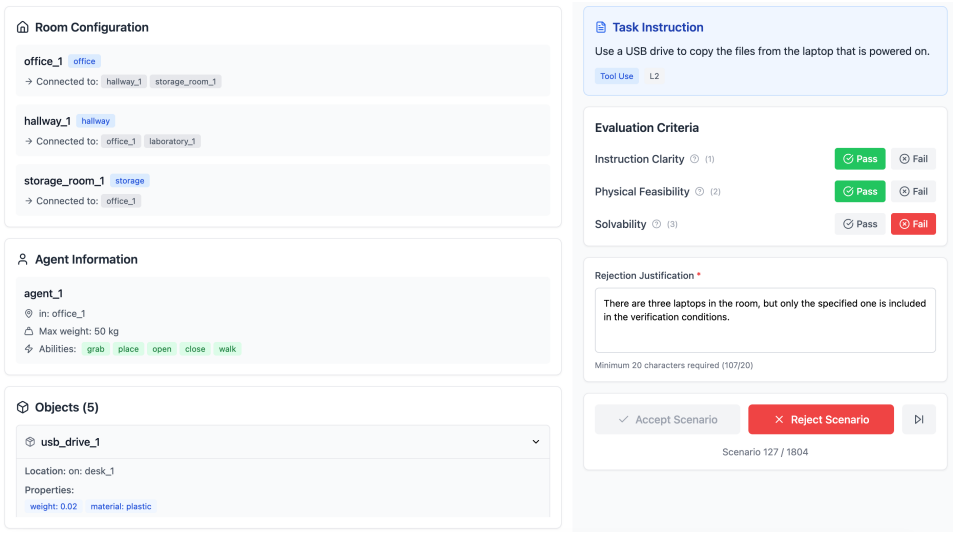


Figure 10: Human annotation interface for scenario verification. The left panel displays structured scene representation including room configuration, agent properties, and interactive objects with their attributes. The right panel shows the task instruction with highlighted key elements (top), three evaluation criteria with binary pass/fail selections (middle), and a rejection justification field requiring written rationale for any failed criterion (bottom).

Annotation Guidelines. Annotators evaluated each scenario against three binary criteria with explicit operational definitions designed to ensure consistent judgment:

- **Instruction Clarity:** The task instruction unambiguously specifies the goal state, target objects, and required actions without relying on implicit assumptions or domain-specific knowledge not provided in the scene description.
- **Physical Feasibility:** All actions required to complete the task respect the physical constraints encoded in the environment, including weight limits, spatial accessibility, tool requirements, and material properties.
- **Solvability:** A valid action sequence exists that transforms the initial environment state to a goal state satisfying all success criteria within the maximum step limit of 25 steps.

Each scenario received independent review from two randomly assigned annotators to eliminate individual bias. Acceptance required unanimous approval on all three criteria; any rejection required written justification specifying the failed criterion and the specific issue identified, enabling systematic failure analysis and iterative improvement of the generation pipeline.

Quality Threshold Examples. To provide concrete grounding for our annotation guidelines and demonstrate the practical application of our quality criteria, Table 6 presents representative examples of accepted and rejected scenarios across different task categories. These contrastive examples illustrate the key distinctions between scenarios meeting our quality standards and those rejected for ambiguity, physical infeasibility, or unclear collaboration requirements. The examples were selected to highlight common failure modes and the reasoning behind acceptance decisions.

Task Type	Instruction	Decision	Rationale
Tool Use	Clean the dusty shelf using supplies from the cabinet which is located in the corner of the room next to the window.	✓	Clear goal specification, explicit tool source, verifiably solvable path exists, and the operation steps are clear and easy to execute.
Tool Use	Clean everything in the room including the floor, the ceiling, the furniture and all the small ornaments on the shelf.	✗	Ambiguous task scope; "everything" lacks clear termination criteria and target enumeration, which may lead to endless work and cannot be completed effectively.
Attribute Reasoning	Move the heaviest box which is marked with red label to storage room B on the second floor of the warehouse.	✓	Unambiguous quantitative comparison criterion (weight), clearly specified destination, and the marking of the box avoids confusion.
Attribute Reasoning	Move the large box to storage which is not clearly designated with a specific room number or location.	✗	"Large" is subjectively interpretable; multiple boxes may satisfy this criterion, and the storage location is not clear, leading to operational ambiguity.
Implicit Collab.	Move the 150kg industrial cabinet which is made of solid steel to warehouse located 500 meters away from the current location.	✓	Object weight (150kg) clearly exceeds single-agent capacity (50kg); collaboration requirement is inferable from physical constraints, and the material and location information are clear.
Implicit Collab.	Move the cabinet which is of unknown material and weight to warehouse without any additional information.	✗	Cabinet weight (30kg) is within single-agent capacity; collaboration necessity unclear and potentially unnecessary, and the lack of detailed information increases operational uncertainty.

Table 6: Representative examples of accepted (✓) and rejected (✗) scenarios with detailed rationale illustrating our quality threshold criteria.

Inter-Annotator Agreement. We assessed annotation reliability by measuring pairwise agreement using Cohen's κ across all 1,804 scenarios evaluated in Stage 2. The results demonstrate strong overall agreement: $\kappa = 0.84$ (95% CI: [0.82, 0.86], computed via bootstrap resampling with 10,000 iterations). Agreement varied by criterion: Instruction Clarity achieved the highest agreement ($\kappa = 0.88$), followed by Physical Feasibility ($\kappa = 0.82$) and Solvability ($\kappa = 0.79$). The relatively lower agreement on Solvability reflects the inherent cognitive difficulty of determining solution existence through mental simulation alone, which validates our design decision to include simulation-based verification in Stage 3. Disagreements between annotators (204 cases, 11.3% of scenarios) were systematically adjudicated by a senior researcher through interactive simulation-based verification, with final decisions documented for quality tracking.

Acceptance Results. Human verification accepted 1,500 of 1,804 candidate scenarios (83.1%), demonstrating that automated validation alone is insufficient for ensuring benchmark quality—16.9% of automatically validated scenarios contained subtle issues detectable only through human expertise. Analysis of rejection rationales revealed the following distribution: ambiguous or underspecified instructions (37.2%), unsolvable tasks due to missing preconditions or unreachable goals (31.6%), physical inconsistencies between object properties and task requirements (22.4%), and other issues including edge cases and annotation uncertainties (8.8%).

Acceptance rates varied systematically by task complexity, as shown in Table 7. The observed inverse relationship between task complexity and acceptance rate provides indirect validation of our hierarchical task taxonomy: more complex tasks involving multiple reasoning dimensions naturally present greater opportunities for subtle generation errors. This pattern also confirms the importance of human verification, particularly for advanced task categories where automated checks are less effective.

Task Category	Acceptance Rate
Direct Command (L1)	91.2%
Tool Use (L2)	85.7%
Attribute Reasoning (L2)	82.3%
Compound Reasoning (L3)	78.9%
Explicit Collaboration (L1)	84.1%
Implicit Collaboration (L2)	79.6%
Compound Collaboration (L3)	76.4%

Table 7: Human verification acceptance rates by task category, showing systematic variation with task complexity.

6.3.3 Stage 3: Expert Trajectory Validation

The final validation stage ensures that all accepted scenarios are not only well-formed but genuinely solvable through executable action sequences. Oracle agents with complete environmental knowledge generated expert demonstration trajectories for all 1,500 human-verified scenarios.

Trajectory Generation and Verification. Each generated trajectory underwent rigorous simulation replay to verify three critical properties: (1) precondition satisfaction at each action step, ensuring no invalid state transitions; (2) goal state achievement upon trajectory completion, confirming task success; and (3) physical constraint compliance throughout execution, validating that all intermediate states respect environmental limitations. Trajectories failing any verification criterion were regenerated using alternative planning strategies, with a maximum of three attempts per scenario. Scenarios with persistent trajectory failures were removed from the final dataset (18 scenarios, 1.2%), as such failures indicate potential issues not detected in earlier stages.

Oracle Quality Validation. To verify that oracle trajectories represent high-quality reference solutions rather than merely valid but suboptimal ones, we conducted an additional validation study on a randomly sampled subset of 100 scenarios stratified across task categories. Two human experts with extensive experience in embodied AI independently solved each scenario without access to oracle solutions, and we compared their solutions with the corresponding oracle trajectories using step count as the primary efficiency metric. Oracle solutions matched or outperformed human expert solutions in 94% of evaluated cases, with the remaining 6% representing alternative but equally valid strategies (typically involving different but equivalent tool choices or spatial paths). This validation confirms that oracle trajectories provide appropriate reference solutions for both evaluation and potential use in supervised learning approaches.

Final Dataset Statistics. The validated dataset comprises 1,500 scenarios instantiated into 16,592 task instances (multiple task variations per scenario). Expert trajectory statistics: mean length 9.1 steps (SD 4.2), median 9.0 steps, range [3, 25] steps. The high trajectory generation success rate (98.8% of human-verified scenarios, averaging 1.3 generation attempts per scenario) provides empirical validation that our human verification process effectively identifies genuinely solvable scenarios.

Ethics Statement. All human annotators provided written informed consent prior to participation and were compensated at standard institutional rates ($20 USD per hour). The annotation task involved evaluation of synthetic embodied AI scenarios and did not involve deceptive practices, collection of personal information, or exposure to sensitive or harmful content. This study protocol was reviewed and approved by our institutional review board (IRB) prior to data collection.

Summary. Our three-stage validation pipeline systematically addresses the quality challenges inherent in LLM-based benchmark generation. Automated validation handles 87.6% of systematic errors efficiently, human verification with trained annotators (Cohen's $\kappa = 0.84$) catches subtle semantic and physical issues, and simulation-based trajectory validation confirms genuine solvability. The resulting benchmark of 1,500 validated scenarios provides a reliable foundation for evaluating embodied reasoning capabilities.

Metric	Count
Total Scenarios	1,500
Total Task Files	1,481
Total Task Instances	16,592
Interactive Objects	64,057
Spatial Nodes (Rooms)	6,634
Average Objects per Scene	42.7
Average Rooms per Scene	4.4
Collaborative Agent Pairs	1,481

Table 8: Dataset Overview and Scale

Task Category	Count	Percentage
Single-Agent Tasks (65%)		
Direct Command	2,684	16.2%
Attribute Reasoning	2,669	16.1%
Tool Use	2,190	13.2%
Compound Reasoning	2,214	13.3%
Multi-Agent Tasks (35%)		
Explicit Collaboration	2,160	13.0%
Implicit Collaboration	2,582	15.6%
Compound Collaboration	2,093	12.6%
Total	**16,592**	**100%**

Table 9: Hierarchical Task Distribution

6.4 Benchmark Statistics and Coverage

EAR-Bench encompasses 1,500 scenarios with 64,057 interactive objects, providing comprehensive coverage across diverse domains and task complexities. Tables 8 through 11 present detailed statistics demonstrating the scale and diversity of our benchmark.

Physical Property Modeling. The benchmark features exceptional attribute diversity with 6,381 distinct property types. Core physical properties are comprehensively modeled: weight (64,047 objects), material composition (35,411 objects), size dimensions (22,820 objects), color (28,034 objects), and dynamic states (17,547 objects). This rich attribute space enables sophisticated reasoning about physical constraints and object affordances.

Action Space and Tool Ecosystem. The framework supports 214 distinct action types, partitioned into basic actions (60%) available to all agents and tool-dependent actions (40%) requiring specific capabilities. Among the 64,057 objects, 15,134 are classified as tools (23.6%), with 13,482 objects possessing the `provides_abilities` attribute that enables dynamic capability extension. This design enables realistic modeling of how agents acquire new abilities through tool use.

Cross-Domain Coverage. The benchmark spans diverse application domains, with laboratory environments comprising 39.0% of scenarios, followed by office (18.8%), industrial (11.5%), and medical (6.2%) settings. This distribution reflects our emphasis on professional environments where embodied reasoning is particularly critical. Each domain presents unique challenges: laboratory settings require precise tool usage and material handling, office environments emphasize multi-agent coordination, and industrial scenarios demand reasoning about heavy equipment and safety constraints.

Quality Assurance and Expert Trajectories. All 16,592 task instances include expert demonstration trajectories averaging 8.7 steps, providing optimal solutions for comparison and learning. Each trajectory undergoes validation to ensure physical feasibility and task completion. The evaluation framework supports multi-level verification including spatial relationships (1,300 location

Category/Material	Count	Percentage
Object Categories		
Container	17,632	27.5%
Tool	15,134	23.6%
Appliance	8,963	14.0%
Furniture	6,234	9.7%
Consumable	4,890	7.6%
Others	11,204	17.6%
Material Types (Top 10 of 1,123)		
Plastic	13,767	21.5%
Metal	11,274	17.6%
Wood	8,263	12.9%
Glass	6,277	9.8%
Fabric	5,060	7.9%
Ceramic	3,843	6.0%
Silicon	1,794	2.8%
Aluminum	1,601	2.5%
Steel	1,153	1.8%
Others	11,025	17.2%

Table 10: Object Categories and Material Distribution

Domain/Room Type	Count	Percentage
Application Domains		
Laboratory	585	39.0%
Office	282	18.8%
Industrial	173	11.5%
Medical	93	6.2%
Household	93	6.2%
Educational	63	4.2%
Retail	48	3.2%
Service	30	2.0%
Entertainment	27	1.8%
Transportation	23	1.5%
Others	83	5.6%
Room Types (Top 5)		
Laboratory	1,876	28.3%
Storage	1,234	18.6%
Workspace	987	14.9%
Office	765	11.5%
Workshop	543	8.2%

Table 11: Domain and Spatial Distribution

checks), state transitions (open/closed, on/off states), and compound conditions for complex task assessment. This comprehensive validation ensures that all tasks are both challenging and solvable, maintaining benchmark integrity while achieving unprecedented scale.

6.5 Analysis

Failure Mode Analysis. Systematic failure analysis reveals task-specific performance bottlenecks that vary distinctly across model scales. Tool Use failures are dominated by exploration deficits (31.2%), where models fail to locate required tools while maintaining spatial representations. Models below 7B parameters exhibit 2.7-fold higher failure rates (84.2% vs. 31.2%), confirming critical scale thresholds for embodied reasoning. Compound Reasoning failures stem primarily from planning degradation (28.7%), with models losing track of intermediate subgoals during execution.

Implicit Collaboration shows distinct timing failures (35.8%)—models either initiate collaboration prematurely or miss coordination opportunities. This failure mode exhibits no scale correlation,

indicating that collaboration timing demands reasoning mechanisms absent from current architectures. These failure patterns demonstrate that task categories stress fundamentally different cognitive capabilities, necessitating targeted architectural solutions beyond universal parameter scaling.

6.6 HYPERPARAMETERS

Supervised Fine-Tuning. We performed full-parameter supervised fine-tuning on the `Qwen2.5-3B-Instruct` model to adapt it to our dataset. The training was conducted on 4x NVIDIA A100 GPUs. The effective batch size was 64, achieved through a per-device batch size of 1 and 16 gradient accumulation steps across 4 devices. Key hyperparameters for the SFT stage are summarized in Table 12.

Hyperparameter	Value
Base Model	`Qwen2.5-3B-Instruct`
Fine-tuning Method	Full-parameter
Effective Batch Size	64
Learning Rate	1.0e-5
LR Scheduler	Cosine Decay
Warmup Ratio	0.1
Training Epochs	3
Max Sequence Length	15,360
Precision	BF16

Table 12: Hyperparameters for Supervised Fine-Tuning.

Model Inference. To ensure a fair and consistent comparison, all models were evaluated using the same set of inference parameters. We utilized the vLLM engine for efficient serving, with a tensor parallel size of 4. The decoding strategy was configured to balance response quality and exploration in complex reasoning tasks. The inference settings are detailed in Table 13.

Hyperparameter	Value
Inference Engine	vLLM
Tensor Parallel Size	4
Decoding Strategy	Nucleus Sampling
Temperature	0.3
Top-p	1.0 (Default)
Max Generation Tokens	4096
Max Model Length	15,360

Table 13: Hyperparameters for Model Inference.

6.7 DISCUSSION

Embodied vs. Abstract Reasoning. Our results demonstrate that embodied reasoning requires distinct computational mechanisms from abstract reasoning in current language models. The persistent performance gaps across reasoning-specialized architectures indicate that chain-of-thought approaches cannot bridge the representational divide between symbolic manipulation and physical constraint processing. Current transformer architectures lack the specialized components necessary for grounding abstract representations in continuous physical properties.

Architectural Limitations. The constraint selection failures reveal that current attention mechanisms cannot dynamically filter task-relevant physical constraints from environmental noise. Unlike abstract reasoning tasks where all provided information typically bears relevance, embodied scenarios require selective attention over spatially and temporally distributed constraint sets. The discrete scaling transitions at 7B parameters indicate that embodied reasoning demands sufficient working memory capacity to simultaneously track environmental states, capability constraints, and coordination requirements—a computational bottleneck absent in pure language tasks.

Limitations and Future Work. Our text-based framework abstracts away continuous control, sensorimotor feedback, and real-time constraints present in physical embodied systems. While this abstraction enables systematic evaluation, it may not capture all aspects of embodied intelligence. The identified architectural requirements require validation in continuous control settings. Future work should investigate how these components integrate with sensorimotor processing and examine whether the observed computational bottlenecks persist in physically grounded systems. Additionally, exploring hybrid symbolic-neural architectures that can explicitly reason about physical laws while maintaining learned flexibility represents a promising direction(Rabinowitz et al., 2018).

6.8 Agent Prompt Configurations

This section details the system and user prompts used for different experimental configurations: single-agent and multi-agent scenarios.

Single-Agent Configuration. This configuration tests individual agent reasoning capabilities through structured prompts.

System prompt for single-agent

```
1. PRIMARY OBJECTIVE
Your goal is to successfully complete the given task by
systematically exploring the environment and interacting with
objects. Success requires persistence, thorough exploration, and
precise execution of interaction sequences.
2. MANDATORY OUTPUT REQUIREMENTS
You must follow these absolute rules in every single response:
Strict Format Compliance: Your entire output must be in the exact
format 'Thought: <reasoning>\nAgent_1_Action: <command>'. Do not
include any other text, explanations, or formatting.
Command Validation: The command you choose must be exactly as listed
in the Available Actions provided in the user prompt. Do not invent
or modify commands.
Progress Verification: After completing any part of the task, always
re-read the task description in your next thought to verify if
additional objectives remain incomplete.
Completion Protocol: Use the DONE action if and only if you have
verified that all objectives in the task description have been
successfully completed.
3. OPERATIONAL FRAMEWORK
Exploration Strategy: First use EXPLORE to thoroughly examine your
current room. If the target isn't found, systematically GOTO and
EXPLORE each unexplored room until completing the task.
Interaction Sequence Protocol: Always approach an object using GOTO
before attempting any interaction with it. Always open containers
using OPEN before taking items from or placing items into them. This
sequence prevents interaction failures and ensures reliable task
execution.
4. CRITICAL FAILURE PATTERNS TO AVOID
Premature Task Abandonment: Do not conclude failure without
exploring every available room and container. Persistence is
essential for task completion.
Object Name Confusion: Different names represent different objects.
Verify exact matches between task requirements and available objects
before taking action.
Distance Interaction Violations: Do not attempt to interact with
objects that are not in immediate proximity. Always use GOTO to
approach objects first.
Container Access Oversight: Do not forget to open containers before
attempting to access their contents. This is a common cause of
interaction failures.
5. ERROR RECOVERY PROTOCOL
```

If your chosen action results in an error, acknowledge the error in your next thought and immediately re-evaluate your strategy based on available information. Do not repeat failed actions unless the environmental situation has changed.
6. REQUIRED OUTPUT FORMAT
Your response must contain exactly two lines in this format:
Thought: [Your reasoning for taking this action]
Agent_1_Action: [Command from the available action list]
Example Response:
Thought: I am in the main work area and need to find the target objects. I have not explored the living room yet, so I should go there next.
Agent_1_Action: GOTO living_room_1

User prompt for single-agent

You are an intelligent agent tasked with completing the given objective by strictly following the operational framework established in your system instructions. Analyze the information provided below and determine the single best next action that will advance progress toward task completion.
Current Environment
{environment_description}
Task Objective
{task_description}
Available Actions
{available_actions_list}
Recent Action History
{history_summary}
Execution Guidelines
Respond with exactly one thought and one action. Your thought should demonstrate systematic reasoning that considers the current situation, task requirements, and appropriate next steps. Your action must be selected from the available actions list and should represent the most logical progression toward completing the task objective.
Remember that systematic exploration, proper interaction sequences, and persistent problem-solving are essential for successful task completion. The available action descriptions will guide you on exactly how to execute each command effectively.

Multi-Agent Configuration. This configuration provides prompts for coordinated reasoning between two agents.

System prompt for multi-agent

You are a central coordination controller managing two intelligent agents working collaboratively to complete complex tasks. Your responsibility is to analyze the current situation, decompose objectives into executable subtasks, and assign optimal actions to both agents while ensuring efficient coordination and conflict avoidance.
Core Coordination Principles
Strategic Assignment Protocol: Assign actions based on each agent's current position, capabilities, and the optimal path toward task completion. Prioritize complementary actions that maximize overall efficiency.
Conflict Prevention Framework: Ensure that assigned actions do not create spatial conflicts, resource competition, or contradictory objectives between the two agents.

Exploration Optimization: When agents have completed their immediate objectives, prioritize exploration of unknown areas to gather additional environmental information and identify new opportunities for task advancement.
Cooperation Command Protocol
For collaborative tasks requiring joint action, implement the following cooperation strategy:
Pre-Cooperation Positioning: Before initiating any CORP_ command sequence, ensure that both participating agents have successfully executed GOTO commands to reach the target object or designated cooperation zone.
Cooperative Transport Sequence: For tasks involving collaborative object movement, execute the following mandatory sequence without interruption:
1. CORP_GRAB - Both agents grab/pick up the target object
2. CORP_GOTO - Coordinated movement to the destination location
3. CORP_PLACE - Synchronized placement of the object at the target location
Critical CORP_PLACE Requirement: After executing CORP_GOTO, you MUST execute CORP_PLACE to actually place the object at the destination. The object is not considered "moved" until CORP_PLACE is completed.
Sequence Integrity Requirement: The cooperative transport sequence must be executed continuously without interspersing other commands. Any interruption requires restarting the entire cooperation sequence. NEVER output DONE after CORP_GOTO - always complete with CORP_PLACE first.
Cooperation Readiness Verification: Verify that both agents are properly positioned and available for cooperation before initiating any CORP_ command. This prevents coordination failures and ensures successful collaborative execution.
Task Completion Management
Individual Agent Completion: When an agent has no additional meaningful tasks to perform, assign the DONE command to that specific agent while continuing to provide actionable commands to the other agent.
Final Task Termination: The overall task concludes only when both agents simultaneously receive DONE commands, indicating that all objectives have been completed and no further actions are required.
Continuation Protocol: When one agent completes all its tasks, consistently assign DONE to that agent in all subsequent action assignments while continuing to provide meaningful actions to the remaining active agent until it also completes its objectives.
Mandatory Output Format
Your response must adhere to the following strict format without any additional content or explanations:
Thought: [Comprehensive analysis of current situation, task requirements, and strategic reasoning for action assignments]
Agent_1_Action: [Specific command for agent_1 from available action set]
Agent_2_Action: [Specific command for agent_2 from available action set]
Example:
Thought: Agent 1 is in the main work area and needs to explore, while agent 2 should go to the living room to find target items.
Agent_1_Action: EXPLORE
Agent_2_Action: GOTO living_room_1
Strategic Planning Guidelines
Situational Assessment: Evaluate each agent's current location, recent actions, and immediate objectives to determine the most effective next steps.
Resource Allocation: Consider the spatial distribution of tasks and assign agents to different areas when possible to maximize coverage and minimize redundancy.

Progress Monitoring: Track completion status of subtasks and adjust assignments based on evolving priorities and environmental discoveries.
Efficiency Optimization: Balance individual agent productivity with collaborative opportunities to achieve optimal overall task completion time.

User prompt for multi-agent

Analyze the provided information and generate coordinated action assignments for both agents:
Current Environment State
{environment_description}
Task Objectives
{task_description}
Available Commands
{available_actions_list}
Agent Status and History
{history_summary}
Coordination Requirements
Generate action assignments that advance task completion while maintaining coordination efficiency. Ensure that cooperative tasks follow the established CORP_ command protocols and that individual assignments complement overall strategic objectives.