CtrlAct: Grounding LLMs to Bridge the Gap between Embodied Instruction and Action

Qingyang Xiao[1] * Bo Su[1] * Ling Sun[2,3] Zhu Zhu[3] Thai Le[1] †

[1]Department of Computer Science, Indiana University
[2]Cognitive Science Program, Indiana University
[3]Department of Linguistics, Indiana University
{xiaoq,subo,ls44,zz79,tle}@iu.edu

Abstract

Large Language Models (LLMs) show strong natural language understanding but often fail in embodied AI settings that demand physical validity and causal reasoning. We evaluate open-source models across four tasks: Goal Interpretation, Subgoal Decomposition, Action Sequencing, and Transition Modeling. Our analysis shows that LLMs often generate full action sequences without considering intermediate environmental feedback, which leads to runtime failures. To address this issue, we encode physical constraints as declarative rules in system prompts and applies Supervised Fine-Tuning (SFT) to align the model with domain dynamics. These interventions improve physical validity, but their effectiveness varies by task. This study clarifies how prompt engineering and SFT affect embodied performance, revealing both the capabilities and the persistent constraints of current open-source models.

1 Introduction

Large Language Models (LLMs) show strong reasoning performance in standard natural language processing tasks. However, using these models to control embodied agents remains difficult, because the agent must convert high-level human instructions into physically valid actions. This requires the model to interpret linguistic intent while respecting action preconditions and environmental constraints (for example, an item cannot be retrieved from a closed fridge without opening it first). Although many studies use commercial models such as GPT, Gemini, and Claude, it remains important to evaluate open-source models to support reproducibility and accessible research.

In this study, we evaluate open-source LLMs on two embodied benchmarks: VirtualHome and BEHAVIOR. These environments cover different levels of complexity, from the structured and predictable household scenes in VirtualHome to the physically realistic and high-diversity object interactions in BEHAVIOR. Beyond baseline evaluation, we study three alignment strategies intended to reduce the gap between linguistic instructions and executable actions: (1) guided reasoning using linguistic rule-based prompts and LLM-generated prompts; (2) domain alignment through Supervised Fine-Tuning (SFT); and (3) inference control using Reinforcement Learning (RL) and activation steering. We also apply a Best-of-N inference strategy to assess the models' reasoning potential.

Our analysis show a clear separation between improving knowledge of environmental dynamics and improving planning behavior. SFT helps models learn transition patterns that follow environment physics, but this gain does not reliably extend to long-horizon planning (subgoal decomposition). In addition, methods that are commonly reported to help in standard NLP tasks, such as RL and activation steering, did not lead to better sequential reasoning in realistic embodied settings. These findings suggest that although open-source models can perform well in structured scenarios when

[*] Both authors contributed equally to this research.

[†] Corresponding author: tle@iu.edu

given suitable guidance, realistic embodied planning demands methods that go beyond simple reward tuning or straightforward fine-tuning.

2 Related Work

Embodied AI studies agents that act and learn in interactive environments and is increasingly important both in practical applications such as domestic assistance [13], industrial automation [30], and interactive agents [14, 15], and in scientific inquiry, where it serves as a setting to examine the differences between human cognition and machine cognition [19, 24]. In particular, embodied tasks highlight the open question of how far complex behavior can be solved through explicit world models [5, 16], and to what extent successful performance instead depends on embodied adaptation, interaction driven learning or evolutionary style processes [3, 16]. In this sense, embodied AI is increasingly framed as a test of whether modern foundation models can realize generalist agents capable of abstract reasoning and grounded action.

The current embodied AI landscape is often organized into three methodological families. First, vision language action models learn policies that map multimodal observations and language instructions directly to actions [4, 7, 31]. Recent surveys [10, 17] show that these models offer strong perceptual grounding and learn rich affordance priors from large scale data, but remain difficult to interpret, lack explicit mechanisms for high level reasoning, and require substantial data and compute to achieve robust real world performance.

Second, LLM planner approaches use object centric, logical or program based abstractions, with the LLM reasoning over this structured interface while classical planners or simulators handle dynamics [2, 12]. Surveys of such robotics [1, 8, 28] highlight that this symbolic modular design offers interpretable intermediate representations, but also suffers from brittle grounding, cascading module errors and distribution mismatch between LLM pretraining data and formal planning interfaces.

Third, reinforcement learning and self evolving agent methods integrate language models with trial and error interaction, self play and reflection driven updates in open ended environments [22, 27]. Recent overviews [6] note that these approaches enable continual adaptation, emergent skills and a form of intelligence grounded in ongoing agent–environment coupling that aligns with 4E perspectives on cognition [26], but they also face persistent challenges in sample efficiency, stability, safety and scaling to long horizon, structurally complex tasks reflective of real human activities.

Across existing embodied AI paradigms, evaluations typically report only final task success, providing little insight into which sub-abilities fail during decision making, as highlighted by Li et al. [11]. This creates a gap in understanding how errors arise within modular components such as linguistic goal grounding, or environment transition consistency. A second gap is the absence of a systematic distinction between *representational domain alignment*, namely how well a model learns the task ontology and symbolic representation during training, and *online reasoning*, namely how those representations are used at inference time. As these capacities are rarely disentangled, prior work lacks a clear mapping from observed error types to their underlying causes. Thus, addressing these gaps can directly help tackle the identified challenges, including brittle grounding, cascading module errors and distribution mismatch between LLM pretraining data and formal planning interfaces [1, 8, 28]. By isolating whether such failures arise from misaligned *representations* or flawed *inference*, we gain clearer targets for improving symbolic grounding, planning reliability and interface robustness. In addition, reinforcement learning and self evolving agent methods depend on effective error analysis driven adaptation; resolving these gaps enables more reliable detection of failure modes and more principled mechanisms for self correction.

3 Benchmark

We use the Embodied Agent Interface (EAI) benchmark dataset [11], which aggregates 338 tasks from VirtualHome [20] and 100 tasks from BEHAVIOR [23], covering a broad range of household activities. Each task is provided with natural-language prompts and task description, LTL goals, symbolic trajectories, and PDDL-style transition models. The benchmark decomposes every task into four embodied-reasoning subtasks: Goal Interpretation, Subgoal Decomposition, Action Sequencing, and Transition Modeling.

Goal Interpretation. Goal Interpretation maps natural-language commands to grounded symbolic objectives. This stage primarily stresses an LLM's semantic understanding: the model must resolve which objects, actions, and spatial or relational constraints are referenced in the instruction, even when phrasing is brief, elliptical, or ambiguous. Beyond identifying the linguistic content, the model must also map surface expressions to their grounded meanings in the simulated world, producing symbolic predicates that accurately represent the intended goal state. Since this is the only stage that directly translates natural language into the task ontology, errors at this level typically arise from representational mismatch or semantic misalignment rather than procedural reasoning.

Subgoal Decomposition. Subgoal Decomposition requires substantially stronger agentic and procedural reasoning than Goal Interpretation. Here, the model must infer the intermediate states that enable progress toward the final objective, despite such steps not being explicitly listed in the instruction. Effective decomposition requires understanding task structure and causal preconditions—for instance, recognizing that the fridge must be opened before placing an item inside. This reasoning is not derivable from language alone; it requires the agent to infer latent affordances, temporal dependencies, and constraints in the environment.

Action Sequencing. Action Sequencing extends procedural reasoning to the level of concrete, low-level operations. The model must determine the precise ordering of actions such that all preconditions are satisfied and the resulting trajectory is feasible. Decisions such as when to approach, grasp, open, or manipulate an object depend on understanding causal and temporal structure in the environment, rather than linguistic cues. Failures at this stage typically stem from missing preconditions, incorrect ordering, or an incomplete understanding of how actions interact over time.

Transition Modeling. Transition Modeling demands accurate prediction of how actions alter the environment's state. This involves reasoning over preconditions, effects, and object relations, and is especially challenging for non-spatial interactions such as grasping or containment. Prior evaluations show that even advanced LLMs achieve low F1 scores on non-spatial relations [11], indicating a persistent gap in modeling fine-grained physical dynamics. This subcomponent is closest to causal modeling: the agent must anticipate how its actions propagate through the environment's state space.

4 Method

We evaluate and improve the inference and embodied representation capacity of LLM models through three interventions, as shown in Figure 1: model sampling evaluation, guided reasoning, and domain alignment. Model sampling evaluation (section 4.1) studies how architectural choices, inference-time parameters, and controlled oversampling influence task performance across environments. Guided reasoning (section 4.2) introduces structured inductive biases through linguistic rule–guided prompts and case-based prompt restructuring to test whether targeted reasoning cues can correct recurring inference failures. Domain alignment methods (section 4.3) aim to adapt the model's understanding of embodied tasks through steering, supervised finetuning, and reinforcement learning. Together, these interventions allow us to study both representation-level and policy-level limits in embodied AI tasks and analyze how each form of intervention reshapes the agent's decision-making patterns in BEHAVIOR and VIRTUALHOME.

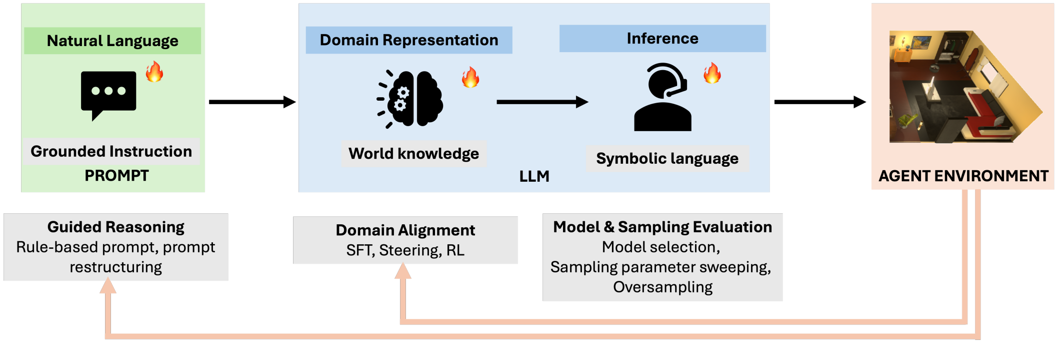

Figure 1: Methodology diagram illustrating how natural-language prompts, LLM internal representations, and inference modules interact to generate symbolic actions in the embodied environment. Guided reasoning, domain alignment, and model-sampling strategies target distinct parts of the pipeline to assess and improve LLM embodied inference.

4.1 Model & Sampling Evaluation

4.1.1 Native Behavior

We first evaluate the native behavior of two open-source foundation models, Qwen3-Next and GPT-OSS. These models have moderate scale, which allows testing on our server. According to their technical reports [25, 18], Qwen3-Next achieves 77.2 on GPQA [21], while GPT-OSS-120B reaches 80.1. Since GPQA is a standard benchmark for advanced reasoning, these results demonstrate the models' fundamental capabilities. This suggests that open-source models possess the physical reasoning patterns required for embodied tasks.

We test several sampling strategies in vLLM, including temperature, top_k, top_p, and repetition_penalty. We observe unstable outputs from GPT-OSS-120B when quantization is enabled, so we run offline inference on eight NVIDIA L40S GPUs without quantization. We allocate sufficient context length and output tokens to avoid restricting model behavior (see implementation details in Section 5).

4.1.2 Oversampling (Best-of-N)

After determining the standard sampling parameters (temperature, top_k, top_p, and repetition_penalty), we further examine the LLM's capabilities through oversampling (specifically, a Best-of-N strategy). Following the methodology of Yue et al. [29], we generate multiple candidate outputs and filter them using our evaluation framework. This approach is designed to probe the reasoning potential available across multiple samples rather than to measure sampling efficiency. We apply this method exclusively to the BEHAVIOR benchmark for subgoal decomposition and action sequencing. We gradually increase the sample size (N) from 10 to 500, which leads to progressive improvements in success rates. Notably, performance on BEHAVIOR subgoal decomposition rises from a baseline of 67 to 96, whereas action sequencing improves from 79 to 85. These results indicate that the impact of oversampling varies depending on the specific task structure.

4.2 Guided Reasoning

Natural language functions in two distinct roles within the embodied-task pipeline. First, it encodes the human-facing task description that must be converted into the simulator's symbolic task specification. Within the pipeline, goal interpretation is the primary stage that directly parses natural-language instructions and maps them to grounded predicates. This makes it the main source of errors caused by mismatches between linguistic form and the task ontology. For this reason, we apply rule-based prompting exclusively to goal interpretation, where explicit linguistic constraints can correct systematic grounding errors (Section 4.2.1). Second, natural language frames the LLM's reasoning process itself. This prompt-level use operates on the model rather than the environment. Therefore, we use LLM-generated structural guidance across all tasks to provide a consistent reasoning interface that does not depend on the specific task description (Section 4.2.2).

4.2.1 Rule-Based Linguistic Guidance

Errors in linguistic grounding often arise at the interface between natural-language task descriptions and the structured task ontology used for embodied inference, particularly during goal interpretation (GI). To reduce these errors, we test whether rule-based linguistic guidance can steer the model toward outputs that align better with the ontology.

We performed a linguistic error analysis by comparing natural-language task descriptions, model-predicted GI outputs, and ground-truth symbolic goals. Using predictions from the VirtualHome training split, we manually identified frequent mismatches and condensed them into a compact set of linguistic generalization rules. One class of rules distinguishes between atelic and telic verbs, where atelic verbs do not imply a completed end state (for instance, *wash clothes* does not entail *clothes are clean*; *drink water* does not entail *an empty cup*). Other rules capture structural cues, such as interpreting verb–object expressions involving animate entities (e.g., *pet cat*) as action goals rather than state changes, mapping prepositional-phrase goal names (e.g., *on sofa*) to spatial relations, and identifying tool-mediated expressions (e.g., *with dishwasher*) that require specific grounded predicates.

We incorporated these rules into the system prompt as inference-time guidance, specifying how surface linguistic forms should map to the task ontology. An example prompt is provided in Appendix A.1. This compiled prompt is applied unchanged to the VirtualHome validation split, enabling us to evaluate whether rules derived from training set patterns improve goal interpretation on unseen tasks.

4.2.2 LLM-Generated Structural Guidance

We also test whether the system prompt can be optimized for each task using the LLM itself. We provide the original raw prompts to the LLM, sample 5–10 input examples (without ground truth), and instruct the model to construct a refined system prompt in Markdown format. The meta-prompt used for this generation is:

```
You are an expert in embodied agent tasks.
You need to prepare a system prompt to help LLM understand prompt
structure, guide reasoning steps, and response in JSON format.
```

The resulting LLM-generated system prompts provide detailed instructions on prompt structure, reasoning rules, representative output formats, and common failure examples. These prompts act as strong structural guidance, allowing us to assess the model's reasoning capabilities under a controlled and consistently enforced interface.

4.3 Domain Alignment

Supervised Fine-Tuning (SFT) Supervised fine-tuning proves highly effective for Transition Modeling across both benchmarks. Even when using only 20% of the ground-truth data, the model achieves a 95% success rate, suggesting that the data requirement could be reduced further. However, given the limited data scale (a few hundred examples per task), transition modeling is prone to rapid overfitting. We attribute this strong performance to the nature of the task: PDDL planning tasks usually follow structured logic that may already be present in the model's pretraining corpus. We hypothesize that the language model has likely encountered PDDL-related data during pretraining, explaining the substantial performance boost observed with minimal fine-tuning.

Activation Engineering (Steering) We attempted activation engineering via contrastive learning by extracting steering vectors from both ground-truth and incorrect outputs across base models (including GPT-OSS and Qwen3-Next). However, this approach yielded negligible improvement. We experimented with the Difference of Means (DoM) vector strategy, varying pair sampling sizes from 5 to 20, and comparing model-targeted versus layer-targeted interventions. None of these configurations led to better performance; in some instances, accuracy actually declined compared to the baseline.

Reinforcement Learning (RL) As demonstrated in [11], LLMs perform significantly worse on Subgoal Decomposition in BEHAVIOR than in VirtualHome or other BEHAVIOR subtasks. This performance gap makes BEHAVIOR subgoal decomposition the logical target for RL intervention. We define the RL reward function as follows: trajectories that satisfy all task goals receive a reward of 1.0; partially correct outputs receive a reward proportional to their success ratio (0.5 × ratio); structurally valid but functionally incorrect sequences receive -0.3; and unparsable outputs receive -1.0. We evaluate RL effectiveness by comparing the RL-tuned model against the base model on the BEHAVIOR subgoal task, specifically measuring subgoal accuracy, transition feasibility, and sequence validity. This setup directly tests whether RL improves temporal coherence in the specific domain where the base model struggles most.

5 Implementation Details

In our experiments, we primarily utilized two large language models: Qwen3-Next-80B (Qwen3-Next) [25] and GPT-OSS-120B (GPT-OSS) [18]. Both models were evaluated under various configurations, including default settings, the application of system prompts, and supervised fine-tuning (SFT). Table 1 summarized the implementation details per task for our rendered final results as in section 6. We will release our code in `https://github.com/omics-ai/ctrlact`.

For inference, we set up the offline inference engine vLLM 0.11.0 with Python 3.12.11, cuda 12.8, PyTorch 2.8.0, Transformers 4.57.1, and Tokenizers 0.22.1. We ran both LLM models on 8 NVIDIA L40S GPUs. Qwen3-Next was accessed from `https://huggingface.co/Qwen/Qwen3-Next-80B-A3B-Thinking` and GPT-OSS from `https://huggingface.co/openai/gpt-oss-120b`.

Table 1: Model and inference settings for tasks in **B** (BEHAVIOR) and **V**(VirtualHome). Each column is a model setting and each row is a different environment-task combination. Shorthands: **GI** = Goal Interpretation, **SD** = Subgoal Decomposition, **AS** = Action Sequencing, **TM** = Transition Modeling. **GPU** indicates the number of L40S GPUs used per setting. **Temp** = temperature. Model names: **GPT-OSS** = GPT-OSS-120B, **Qwen3** = Qwen3-Next-80B.

Env	Task	GPU	Model	Max model len	Max tokens	Temp	top-k	top-p
B	**GI**	8	GPT-OSS	131,072	16,384	1.0	20	0.90
	SD	4	Qwen3	131,072	16,384	0.4	20	0.85
	AS	4	Qwen3	131,072	16,384	0.4	20	0.80
	TM	8	GPT-OSS	131,072	16,384	0.3	20	0.95
VH	**GI**	8	GPT-OSS	131,072	16,384	1.0	20	0.95
	SD	8	GPT-OSS	131,072	16,384	0.8	20	0.90
	AS	8	GPT-OSS	131,072	16,384	1.0	20	0.85
	TM	8	GPT-OSS	131,072	16,384	0.8	20	0.90

6 Final Results

Table 2 presents the performance of two LLM models across four experimental configurations: (1) Native: the baseline model without specialized system prompts or SFT; (2) the model with optimized system prompts; (3) the model utilizing the oversampling strategy; (4) the supervised fine-tuned (SFT) model. While our methodology also explored activation engineering (steering) and reinforcement learning, we exclude these methods from the primary results table as they yielded negligible improvement over the baseline in the evaluation phase.

Table 2: Final results (% accuracy) for each task and environment, grouped by base model: Qwen3-Next-80B (Qwen3-Next) was used for Subgoal Decomposition (SD) and Action Sequencing (AS) in **Behavior**, and GPT-OSS-120B (GPT-OSS) was used for Goal Interpretation (GI), Transition Modeling (TM) in **Behavior**, and for all tasks in **VirtualHome**. Each cell shows test accuracy. "Sys. Prompt & SFT" indicates system prompt and supervised fine-tuning were both applied.

	Behavior				**VirtualHome**			
	GI	**SD**	**AS**	**TM**	**GI**	**SD**	**AS**	**TM**
Native	83.2*	67†	79†	80.9*	36.9*	66.2*	71.6*	64.4*
Sys. Prompt	-	-	-	82.2*	42.2*	73.4*	72.1*	70.8*
Oversampling	-	96†	85†	-	-	-	-	-
SFT	-	-	-	98.85*	48.2*	-	-	95.6*
Final	83.2*	96†	85†	98.85*	48.2*	73.4*	72.1*	95.6*

*: GPT-OSS results; †: Qwen3-Next results. For GI and TM on Behavior, and all columns on VirtualHome, base model is GPT-OSS. For SD and AS on Behavior, base model is Qwen3-Next.

7 Case Study and Lessons Learned

7.1 Native Behaviors Are Competent but Not Universally Reliable

Table 2 shows that the baseline LLM achieves scores around 80 on three tasks (Goal Interpretation, Action Sequencing, and Transition Modeling) within the Behavior benchmark, yet yields significantly lower performance on the VirtualHome benchmark.

During the development phase, we tested Qwen3-30B (Qwen3-30B-A3B-Thinking-2507), Qwen3-Next (Qwen3-Next-80B-A3B-Thinking), GPT-OSS-20B (High reasoning), and GPT-OSS-120B (High reasoning). We observed superior performance with the larger-scale models (Qwen3-Next and GPT-OSS-120B). We also tested multiple sampling parameters on Qwen3-Next and GPT-OSS-120B to investigate model capacity. For Qwen3-Next, lower temperatures resulted in better performance, whereas higher temperatures favored GPT-OSS-120B (see Table 1).

7.2 Guided Reasoning Excels in General Environments but Not Realistic Ones

Guided reasoning methods demonstrate strong benefits in VirtualHome but do not transfer effectively to the more realistic and complex Behavior environment. In VirtualHome, both the rule-based system prompt for the goal interpretation task and the LLM-generated restructuring prompts improve model performance across all major evaluation measures. These interventions help the model align surface language with the task ontology and correct systematic errors inherent to this structured environment.

For GI in VirtualHome, the rule-based prompt raises the baseline F1 score from 0.407 to 0.548 in the development phase, and from 0.369 to 0.422 in the evaluation phase. This confirms that handcrafted rules effectively address the recurrent patterns identified during our analysis. The LLM-generated restructuring prompts, applied across all VirtualHome tasks, yield similar improvements, indicating that guided reasoning provides consistent gains in this general and semantically regular setting.

In Behavior, however, these same LLM-generated restructuring prompts do not improve planning. They strengthen only transition modeling, with no gains in other tasks. This contrast highlights a consistent pattern: guided reasoning enhances domain-level knowledge representation in structured environments but does not scale to realistic embodied settings where long-horizon planning and action decomposition dominate task difficulty.

7.3 Domain Alignment Improves Knowledge, Not Planning

Across all evaluated methods (SFT, steering, RL), only SFT produced consistent gains, and these were confined to transition modeling in both VirtualHome and Behavior. In contrast, methods aimed at improving planning (steering and RL) failed to enhance sequential reasoning and, in most cases, degraded performance.

SFT reliably strengthened domain-specific knowledge representations, improving the model's ability to infer feasible state transitions and object–relation structures. However, these improvements did not translate into better long-horizon action sequencing or subgoal decomposition, even in VirtualHome where overall task complexity is lower.

Steering and RL exhibited the opposite trend: they decreased overall subgoal accuracy and increased temporal and precondition-related errors. Steering destabilized ordering decisions without improving grounding. Similarly, despite structured rewards, RL underperformed the base model regarding execution success, goal accuracy, and sequence validity. The training signals for RL were likely too noisy or insufficiently shaped for the complexity of the planning space, causing the policy to diverge from the pretrained structure rather than refine it.

Taken together, these results illustrate a clear distinction: methods that provide domain alignment (SFT) enhance knowledge-level transition modeling, whereas methods intended to improve planning (steering, RL) often prove detrimental. This suggests that planning in realistic embodied environments requires richer intermediate supervision or hierarchical modeling rather than simple reward-shaping or lightweight control signals.

8 Limitation

One limitation of our current approach is the reliance on on-policy evaluations, which limits the efficiency of the planning process. A promising future direction is to train an offline reward model that can guide inference without requiring simulator rollouts, thereby utilizing model capacity more effectively. Another potential attempt is the multi-modal integration by combining vision and language models to strengthen embodied learning. This would allow us to investigate whether such models can maintain reasoning capabilities even under degraded inputs, mirroring the human ability to infer physical structure without direct perception (e.g., estimating the number of windows in a distant building).

9 Conclusion

In this work, we evaluated the capabilities of open-source Large Language Models on embodied tasks using BEHAVIOR and VirtualHome benchmarks. Our results demonstrate that while current models possess competence in structured environments, they struggle to generalize to realistic, physically constrained settings. We found that guided reasoning (including linguistic rule-based prompting and structural prompt optimization) works effectively in the highly regularized VirtualHome domain but fails to scale to the complex planning challenges in BEHAVIOR. Similarly, while Supervised Fine-Tuning successfully improved domain-specific transition modeling, methods explicitly targeted at planning, such as reinforcement learning and activation steering, failed to enhance sequential reasoning and often degraded performance. These findings suggest that simple reward shaping is insufficient for realistic embodied planning. Instead, future progress will likely require richer domain supervision to bridge the gap between static knowledge representation and dynamic action execution.

Reproducibility Statement

We did offline inference using vLLM 0.11.0 with cuda 12.8, PyTorch 2.8.0, Transformers 4.57.1, and Tokenizers 0.22.1 on 8 NVIDIA L40S GPUs. We used Qwen3-Next-80B from `https://huggingface.co/Qwen/Qwen3-Next-80B-A3B-Thinking` and GPT-OSS-120B from `https://huggingface.co/openai/gpt-oss-120b`. Detailed hyperparameters are provided in Table 1. We performed supervised fine-tuning on Tinker [9] based on Tinker cookbook.

References

[1] Mohamed Aghzal, Erion Plaku, Gregory J Stein, and Ziyu Yao. A survey on large language models for automated planning. *arXiv preprint arXiv:2502.12435*, 2025.

[2] 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.

[3] Zhulin An, Xinqiang Yu, Chu Wang, Yinlong Zhang, and Chunhe Song. Embodied intelligence: Recent advances and future perspectives. *The Innovation Informatics*, 1(1):100008–1, 2025.

[4] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: an embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, pages 8469–8488, 2023.

[5] Pascale Fung, Yoram Bachrach, Asli Celikyilmaz, Kamalika Chaudhuri, Delong Chen, Willy Chung, Emmanuel Dupoux, Hongyu Gong, Hervé Jégou, Alessandro Lazaric, et al. Embodied ai agents: Modeling the world. *arXiv preprint arXiv:2506.22355*, 2025.

[6] Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, et al. A survey of self-evolving agents: On path to artificial super intelligence. *arXiv preprint arXiv:2507.21046*, 2025.

[7] 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.

[8] Yeseung Kim, Dohyun Kim, Jieun Choi, Jisang Park, Nayoung Oh, and Daehyung Park. A survey on integration of large language models with intelligent robots. *Intelligent Service Robotics*, 17(5):1091–1107, 2024.

[9] Thinking Machines Lab. Tinker, 2025.

[10] Haoran Li, Yuhui Chen, Wenbo Cui, Weiheng Liu, Kai Liu, Mingcai Zhou, Zhengtao Zhang, and Dongbin Zhao. Survey of vision-language-action models for embodied manipulation. *arXiv preprint arXiv:2508.15201*, 2025.

[11] Manling Li, Shiyu Zhao, Qineng Wang, Kangrui Wang, Yu Zhou, Sanjana Srivastava, Cem Gokmen, Tony Lee, Li Erran Li, Ruohan Zhang, Weiyu Liu, Percy Liang, Li Fei-Fei, Jiayuan Mao, and Jiajun Wu. Embodied agent interface: Benchmarking llms for embodied decision making, 2025.

[12] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Peter Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *Advances in Neural Information Processing Systems*, volume 35, pages 29218–29230, 2022.

[13] Matthew Lisondra, Beno Benhabib, and Goldie Nejat. Embodied ai with foundation models for mobile service robots: A systematic review, 2025.

[14] Yang Liu, Xinshuai Song, Kaixuan Jiang, Weixing Chen, Jingzhou Luo, Guanbin Li, and Liang Lin. Meia: Multimodal embodied perception and interaction in unknown environments, 2024.

[15] Yang Liu, Xinshuai Song, Kaixuan Jiang, Weixing Chen, Jingzhou Luo, Guanbin Li, and Liang Lin. Multimodal embodied interactive agent for cafe scene. *CoRR*, 2024.

[16] Xiaoxiao Long, Qingrui Zhao, Kaiwen Zhang, Zihao Zhang, Dingrui Wang, Yumeng Liu, Zhengjie Shu, Yi Lu, Shouzheng Wang, Xinzhe Wei, Wei Li, Wei Yin, Yao Yao, Jia Pan, Qiu Shen, Ruigang Yang, Xun Cao, and Qionghai Dai. A survey: Learning embodied intelligence from physical simulators and world models, 2025.

[17] Yueen Ma, Zixing Song, Yuzheng Zhuang, Jianye Hao, and Irwin King. A survey on vision-language-action models for embodied ai, 2025.

[18] OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

[19] Giuseppe Paolo, Jonas Gonzalez-Billandon, and Balázs Kégl. Position: a call for embodied ai. In *Forty-first International Conference on Machine Learning*, 2024.

[20] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs, 2018.

[21] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark, 2023.

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

[23] Sanjana Srivastava, Chengshu Li, Michael Lingelbach, Roberto Martín-Martín, Fei Xia, Kent Vainio, Zheng Lian, Cem Gokmen, Shyamal Buch, C. Karen Liu, Silvio Savarese, Hyowon Gweon, Jiajun Wu, and Li Fei-Fei. Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments, 2021.

[24] Fuchun Sun, Runqi Chen, Tao Ji, et al. A comprehensive survey on embodied intelligence: Advancements, challenges, and future perspectives. *CAAI Artificial Intelligence Research*, 3:9150042, 2024.

[25] Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[26] Francisco J Varela, Evan Thompson, and Eleanor Rosch. *The embodied mind, revised edition: Cognitive science and human experience*. MIT press, 2017.

[27] 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. *arXiv preprint arXiv:2305.16291*, 2023.

[28] Jiaqi Wang, Enze Shi, Huawen Hu, Chong Ma, Yiheng Liu, Xuhui Wang, Yincheng Yao, Xuan Liu, Bao Ge, and Shu Zhang. Large language models for robotics: Opportunities, challenges, and perspectives. *Journal of Automation and Intelligence*, 4(1):52–64, 2025.

[29] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *2nd AI for Math Workshop @ ICML 2025*, 2025.

[30] Chaoran Zhang, Chenhao Zhang, Zhaobo Xu, Qinghongbing Xie, Jinliang Hou, Pingfa Feng, and Long Zeng. Embodied intelligent industrial robotics: Concepts and techniques, 2025.

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

A Appendix

A.1 Rule-Based System Prompts Used across Tasks

```
 You MUST follow these rules exactly:
0) Output strict JSON with keys exactly:  ’node goals’, ’edge goals’,
’action goals’.
If the goal description is a sequence of actions, ONLY focus on the
final state, not goal at each step.
1) Do NOT invent relations for any object, objects/states/actions.
All NEED to be strictly as defined in the prompt.
Use exact ontology spellings like ’toothpaste’, ’drinkingglass’,
’HOLDSLH’, ’HOLDSRH’.
Interpret ’Go to the toilet’ as ’Use the toilet’ for all prompts.
Use the Goal name to determine node/edge/action goals.  Use the goal
description along with relevant objects ONLY to fill variable values
(e.g., fill in egg for grocery)!!
2) ONLY use within prompt domain linguistic knowledge, BUT APPLY
domain general world knowledge:  if everything can only be ON
a machine, then use ON instead of INSIDE to represent physical
insideness.
3) Prioritize ’node goals’, ’edge goals’ OVER ’action goals’.
4) If the ’Goal Name’ is of a verb-object structure, and the object
is an or part of an animate being, PRIORITIZE ’action goals’ and
’edge goals’ over ’node goals’.
5) Pay attention to the TELICITY of the VERB in the task goal:  if
the action goal does NOT entail the endpoint of the action, then DO
NOT use node/edge goals that describe the endpoint.
6) Do NOT use ’action goals’ unless absolutely necessary.  AVOID
’action goals’ not entailed by the task goal.
7) Before committing to each specific goal, ask and reflect if the
goal is strictly entailed by the task goal.
8) DO NOT use ’node goals’ and ’edge goals’ if they do NOT
necessarily differ from the initial state.
The prompts forgot to define an action SLEEP, add it in when
considering action goals.
9) Prefer EDGE goals.  If edges alone capture the goal, leave node
and action lists EMPTY.
10) Handedness matters.  Never substitute LH and RH.
11) Actions are permitted only if the goal CANNOT be represented by
’node goals’ and ’edge goals’ (alone); otherwise leave ’action goals’
empty.
12) If uncertain, match prior example spelling exactly.
13) For telic verbs (those with a clear completion), represent the
endpoint using node/edge goals unless the ontology requires the
action itself to capture the meaning.
14) For atelic verbs or dynamic human-controlled manipulations (e.g.,
LOOKAT, TOUCH, WATCH, GRAB, RINSE, TYPE), include the action goal
only when the verb is explicitly present in the Goal name.
Return only the JSON. No explanations.
```