NEURAL INFORMATION PROCESSING SYSTEMS

Learning And Mining from DatA

Re^2 Agent: Reflection and Re-execution Agent for Embodied Decision Making

Yang Chen[1,2] Hong-Jie You[1,3] Jie-Jing Shao[1] Xiao-Wen Yang[1,3] Ming Yang[1,3]
Yu-Feng Li[1,3] Lan-Zhe Guo[1,2,*]
(nju-lamda12)

[1] State Key Laboratory of Novel Software Technology, Nanjing University
[2] School of Intelligence Science and Technology, Nanjing University
[3] School of Artificial Intelligence, Nanjing University
[*] Corresponding author (guolz@lamda.nju.edu.cn)

Embodied Agent Interface Challenge @ NeurIPS 2025

Background

LAMDA Learning And Mining from DatA

Embodied Decision Making

Perception
State
Environment Dynamics

↓

Decision making

↓

Goals & Value
Action Policy
Learning & Memory

Embodied Benchmark

Embodied task

↓

Action 1 -> Action 2 -> … -> Action N

↓

Sim.A or Sim.B

↓

Succ. √ or Fail ×

Background

Current Benchmark

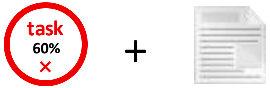
Embodied task

A lack of standardization in three areas:
1. Embodied decision-making tasks
2. Modules that an LLM can interface with or be implemented for
3. Fine-grained evaluation metrics beyond a single success rate

EAI Benchmark

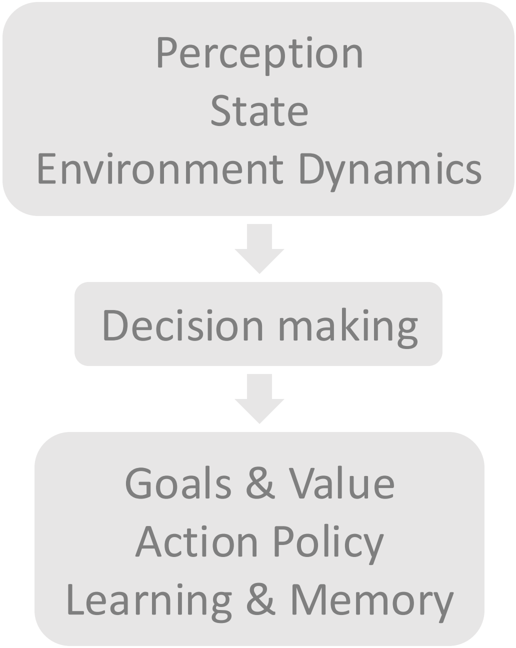
Embodied task

1. Standardization of goal specifications
2. Standardization of modules and interfaces
3. Standardization of fine-grained evaluation metrics with broad coverage

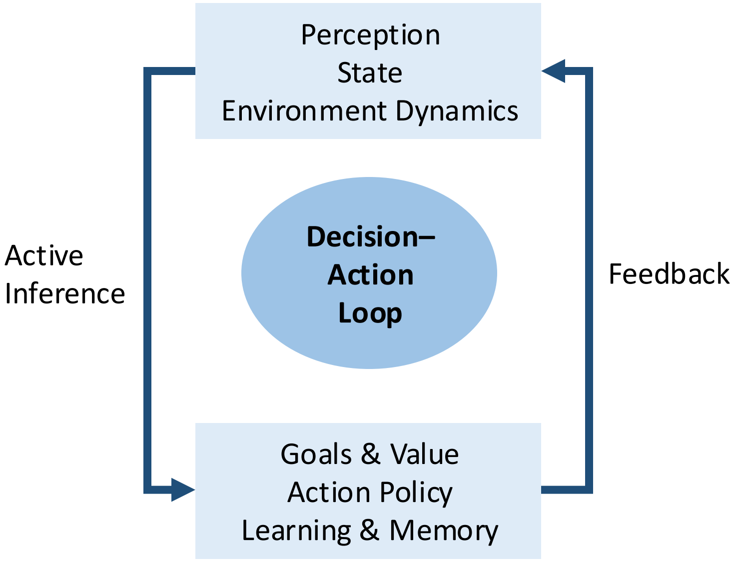


Motivation

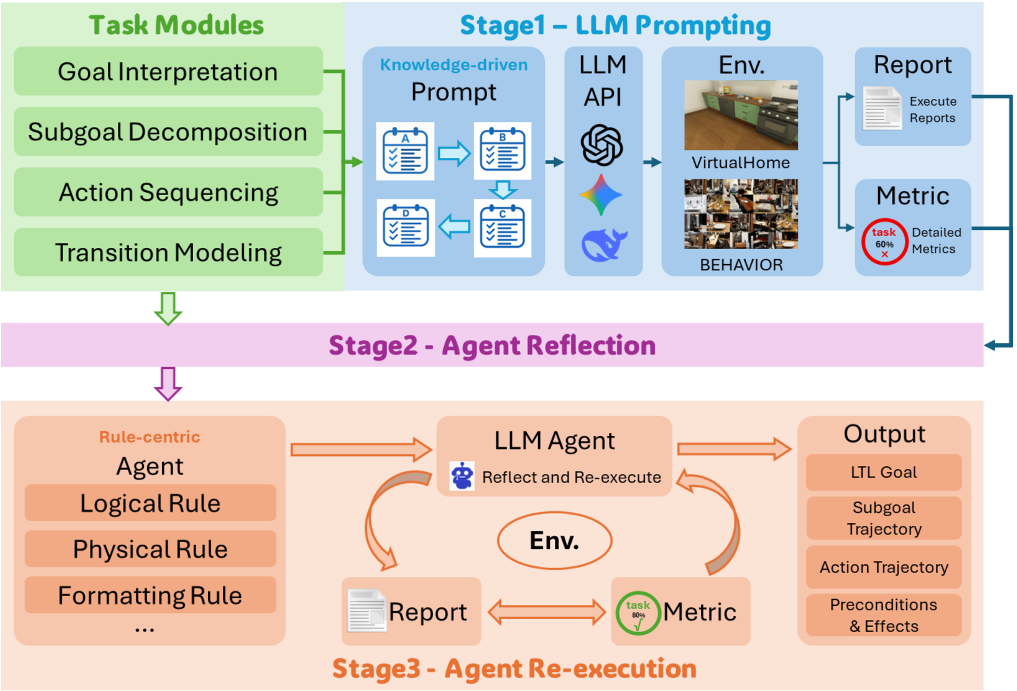
Current Embodied Decision Making



Decision-Action Loop



Method



Overview of the Re[2] Agent framework. The system operates in three stages: (1) knowledge-driven prompt design; (2) reflective of execution reports and metrics to derive task rules; and (3) re-execution of the task through environmental interaction to produce the final output.

Method

Module 1: Goal Interpretation

Knowledge-Driven Prompt Design:

- Goal Interpretation: Formalized as minimal end-state synthesis under hard constraints.
- Structured Output: Deterministic JSON with three lists: node goals, edge goals, and action goals.
- Relation-First Modeling: Prioritizes containment and placement tasks.
- State-Action Separation: Prefers state goals for device control; uses action goals only when necessary.
- Minimization Principle: Removes intermediate states and contradictions to improve executability.
- Outcome: Compact, simulator-aligned specifications that reduce error propagation and stabilize downstream evaluation.

Method

Rule-Centric Agent Design:

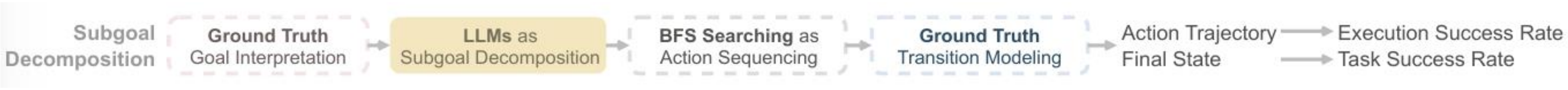
- Generator + Corrector: Combines initial prediction with reflective feedback and temporal logic.
- Correction Process: Uses original goal, previous output, and reference examples to produce corrected JSON and LTL-like formulas.
- Rule Induction: Driven by execution reports and success metrics for consistency with environment constraints.
- Closed-Loop Refinement: Agent re-executes tasks with injected rules, analyzes outcomes, and iterates.
- Result: Higher success rates, cleaner specifications, and improved semantic alignment.

Method

Task2: Subgoal Decomposition

Knowledge-Driven Prompt Design:

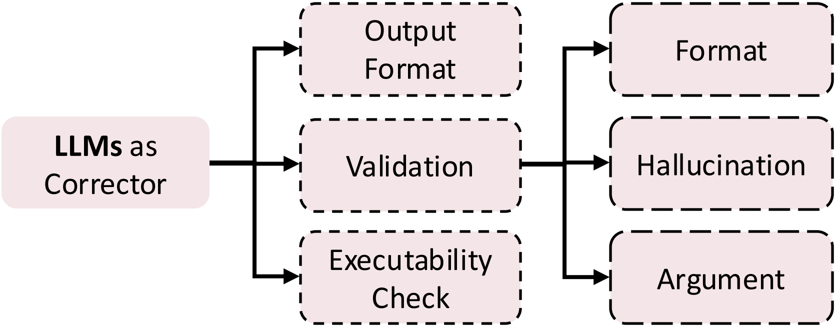
- Goal: Generate an executable action sequence transforming initial state s_0 to target state g.
- Strategy: One-shot autoregressive generation using LLMs; full sequence produced in a single forward pass.
- Inputs: s_0 , g, and interactable objects O.
- Constraints:
  - Physical: Hand occupancy, free manipulators.
  - Temporal: Correct ordering (e.g., open container before placing objects).
  - Logical: Action-specific preconditions.



Method

Rule-Centric Agent Design:

- Output Format: Python-evaluable list of dictionaries; no explanatory text or markdown.
- Validation Pipeline:
  - Format Validation: Structural correctness.
  - Hallucination Detection: Ensure actions $\in A$, objects $\in O$.
  - Argument Validation: Parameter consistency.
- Executability Check: Sequential execution via transition model M; detects missing steps, extra steps, incorrect order, infeasible actions.



Comparison of BEHAVIOR and VirtualHome

- **Common Strengths:** High grammatical correctness, strong state goal achievement.
- **Differences:**
  - VirtualHome excels in planning coherence & subgoal decomposition, but struggles with semantic interpretation & transition modeling.
  - BEHAVIOR shows robust semantic reasoning & transition modeling, but weaker in subgoal decomposition.
- **Recommendation:**
  - Develop environment-adaptive strategies for navigation, semantic abstraction, and action-space complexity.
  - Incorporate rule-based validation and explicit navigation modules to reduce hallucinations and ordering errors.

Experiment

Table 1: Results (%) overview. *V*: VirtualHome, *B*: BEHAVIOR..

Model	Overall Perf.	Average Perf. *Module SR*		Goal Interpretation F_1		Action Sequencing *Task SR*		Action Sequencing *Execution SR*		Subgoal Decomposition *Task SR*		Subgoal Decomposition *Execution SR*		Transition Modeling F_1		Transition Modeling *Planner SR*	
		V	*B*	*V*	*B*	*V*	*B*	*V*	*B*	*V*	*B*	*V*	*B*	*V*	*B*	*V*	*B*
GPT-4o	52.25	49.41	55.09	24.9	77.4	68.3	40.0	83.6	50.0	68.1	45.0	85.1	51.0	43.1	62.9	29.6	53.0
GPT-o3	64.43	61.75	67.10	35.3	79.1	68.9	64.6	83.2	70.8	74.3	52.0	87.1	57.3	44.6	52.4	92.4	93.0
GPT-5	68.32	68.09	68.55	36.3	79.2	72.3	71.0	84.9	74.0	73.2	54.0	87.2	60.0	36.4	43.0	86.3	97.0
Re[1] Prompt	74.54	71.58	77.50	49.0	84.9	71.2	73.0	84.9	78.0	74.7	80.0	**88.0**	86.6	54.8	43.0	91.4	97.0
Re[2] Agent	**81.36**	**76.36**	**86.35**	**57.9**	**85.0**	**73.5**	**81.0**	**84.9**	**91.0**	**74.7**	**80.0**	86.6	**88.0**	**98.8**	**99.8**	**99.9**	**99.0**

We evaluated GPT-series models (GPT-4o, GPT-o3, GPT-5) using closed-source API calls, with **GPT-5** as the primary planner. Tasks were long-horizon, so the context length was set to **8192** tokens, and each task allowed up to **5** **retries** with **reflective** summaries to improve subsequent attempts.

Re[1] means using only one round of prompt optimization.

Conclusion

- Our work establishes that iterative reflection and rule abstraction are critical for robust embodied decision making. Evaluated on the Embodied Agent Interface benchmark, our Re^2 agent achieves a combined score of 81.36 (86.35 on BEHAVIOR, 76.36 on VirtualHome), significantly outperforming non-iterative baselines and **demonstrating effective closing of the gap between language understanding and grounded action execution**.

- Looking ahead, we aim to build upon the Re^2 agent framework by collecting more environmental interaction data, thereby truly integrating **the learning capabilities of LLMs with dynamic feedback from embodied scenarios to achieve closed-loop responsiveness in open-world settings**.

Thanks for your listening!

Presented by **Yang Chen**

2025-12-07