Plan Verification for LLM-Based Embodied Task Completion Agents

Ananth Hariharan, Vardhan Dongre, Dilek Hakkani-Tür, Gokhan Tur

University of Illinois Urbana-Champaign

NEURAL INFORMATION PROCESSING SYSTEMS

Motivation & Problem Setting

Human demonstrations and LLM-generated plans for embodied tasks often contain:

- redundant or repeated actions
- irrelevant steps and object picks
- contradictions in state
- missing actions required to complete goals

These errors reduce data quality for imitation learning and RL. At the same time, human trajectories include valuable error-recovery patterns that should be preserved. We aim to clean demonstrations efficiently while retaining such structure.

Challenges in LLM-Based Planning

- LLMs generate plausible but frequently non-executable plans
- Steps may not correspond to available actions or states
- Missing prerequisites lead to incomplete tasks
- Rule-based verification is brittle and domain-specific
- Need a scalable, model-agnostic method to detect and fix errors

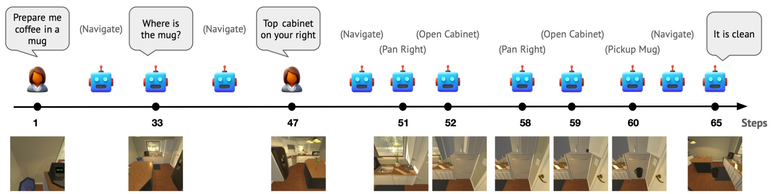

Figure 1: Diagram of Sample Workflow in TEACh Dataset

Contributions

- **General Verification Framework:** A two-agent loop where a Judge LLM critiques actions and a Planner LLM applies revisions.
- **Natural-Language Criteria:** No heuristics or ground-truth simulators; uses zero-shot reasoning.
- **Broad Generalization:** Handles irrelevant, redundant, contradictory, and missing actions.
- **Fast Convergence:** 96.5% of plans fixed within ≤ 3 iterations.

Formal Summary

We represent a plan π as a sequence:

$$\pi = (a_1, \ldots, aT)$$

Error set E :

$$E(\pi) = \{i | ai \text{ is redundant, contradictory, or missing}\}$$

Goal: find the shortest valid plan achieving task goal g :

$$\pi^* = \arg\min_{\tilde{\pi}} |\tilde{\pi}| \quad \text{s.t. } \tilde{\pi} \text{ achieves } g$$

Judge LLM J outputs critiques i :

$$J(g, \pi) \to (i, \text{type}, \text{reason})$$

Planner P applies corrections:

$$P(\pi, C) \to \pi'$$

Verification operator V :

$$V = P \circ J$$

Convergence assumption:

$$\mathbb{E}\left[E\left(\pi^{(k+1)}\right)\right] \leq (1-\delta), E\left[E\left(\pi^{(k)}\right)\right]$$

Method Overview

Our framework operates as an iterative critique-and-rewrite dialog:

1. **Planner** proposes a candidate action sequence.
2. **Judge** reviews each step, flagging `REMOVE` and `MISSING` actions with natural-language explanations.
3. **Planner** revises the sequence accordingly.
4. **Loop** stops when no further issues appear (max five rounds).

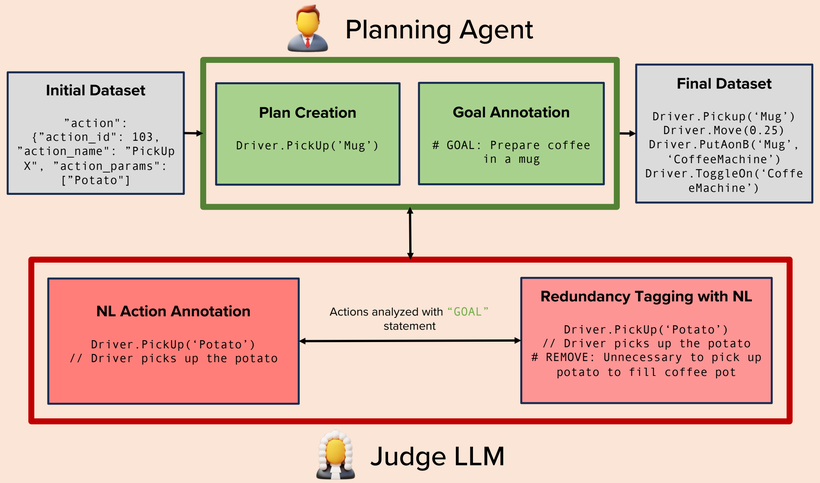

Figure 2: Diagram of Planning Agent and Judge LLM Interaction Process for Plan Verification

Experiments & Findings

On 1,408 manually annotated TEACh actions:

Judge LLM	Recall	Precision
GPT o4-mini	**80%**	93%
DeepSeek-R1	68%	**100%**
Gemini 2.5	74%	90%
LLaMA4 Scout	74%	85%
Rule-based	22%	71%

Iterative critique-and-rewrite consistently improves results, shown below:

Judge LLM	Planner LLM – Recall (%) / Precision (%) / F-score			
	GPT o4-mini	DeepSeek-R1	Gemini 2.5	LLaMa 4 Scout
GPT o4-mini	88 / 90 / 89.0	**90 / 80 / 84.7**	85 / 91 / 87.8	89 / 87 / 87.9
DeepSeek-R1	65 / 99 / 78.5	**68 / 100 / 80.9**	62 / 100 / 76.5	66 / 98 / 78.9
Gemini 2.5	84 / 98 / 90.7	86 / 97 / 91.2	**89 / 99 / 93.9**	89 / 96 / 92.2
LLaMa 4 Scout	76 / 92 / 83.5	81 / 90 / 85.3	79 / 93 / 85.9	75 / 89 / 81.6

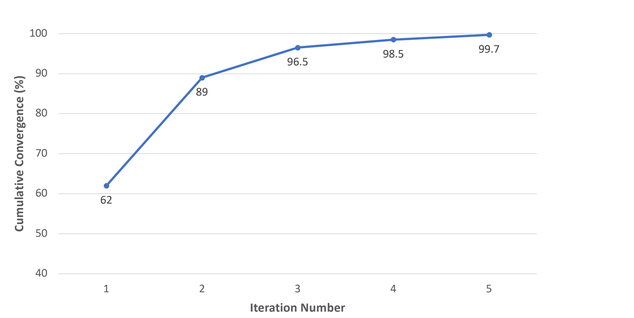

Figure 3: Cumulative convergence of action sequences across iterations.

Qualitative Findings

Successful Corrections

- Removing early or irrelevant pickups
- Eliminating contradictory toggles
- Inserting missing goal-critical steps

Recall Failures

- Long-range dependencies (e.g., picking up an object far before use)
- Multi-action context requiring deeper reasoning

Precision Failures

- Multi-step preparations mis-labeled as redundant
- Valid reuse of objects incorrectly flagged
- These highlight strengths in surface-level logic and weaknesses in long-horizon reasoning.