Multi-Expert Workflow Robot Planning Model Based on Routed Cyclic Optimization

Zhengyu Man[1] Jun Wang[1] Xiaotian Pan[2] Zhaobo Qi[2]

[1]Harbin Institute of Technology

Legend_man@qq.com

Abstract

This paper proposes a "Multi-Expert Workflow based on Routed Cyclic Optimization" model for the EAI Challenge. Our core innovation lies in designing a dynamic dual-routing architecture: First, a "Classifier Expert" analyzes the input task and precisely categorizes it into one of the four main challenge tasks (Goal Interpretation, Subgoal Decomposition, Action Sequencing, and State Transition Modeling). Subsequently, the problem is routed to a specialized "Planner-Advisor" expert group that is highly optimized for this specific task. The workflow employs multi-round iterative cycles for refinement, incorporating dynamic cycle control and a "Final Review" mechanism for persistent errors. Our method demonstrates strong generalization capabilities and performance improvements in both the **BEHAVIOR** and **VirtualHome** environments, achieving an overall score of **70.38** across all tasks. These results validate the effectiveness of our "classification-routing-optimization" workflow in addressing complex embodied AI planning problems.

1 Introduction

In recent years, embodied artificial intelligence has made remarkable progress, with a core challenge being enabling agents to understand natural language instructions and autonomously plan and execute complex tasks in complex, dynamic simulated environments. However, natural language instructions are inherently ambiguous, abstract, and incomplete, creating a significant gap between an agent's understanding and action.

To address this challenge, the **Embodied Agent Interface (EAI) Challenge** decomposes the core problem into four distinct yet interrelated subtasks: Goal Interpretation, Subgoal Decomposition, Action Sequencing, and Transition Modeling. We argue that attempting to use a single monolithic model to **uniformly** solve all these heterogeneous tasks is inefficient. For instance, the rigorous logical reasoning required for precisely defining PDDL predicates (Transition Modeling) fundamentally differs from the commonsense knowledge needed to interpret user's ambiguous intentions (Goal Interpretation).

Therefore, we propose a "Multi-Expert Workflow based on Routed Cyclic Optimization" for robotic planning. Our methodology abandons the single-expert, independent adjudication solution, instead constructing a dynamic architecture that mimics collaborative human expert teams.

The core innovation of this architecture lies in its "routing classification" and "multi-level expert review workflow" mechanisms. First, a "Classifier Expert" rapidly analyzes the input task and precisely routes it to one of four predefined specialized workflows. Second, the task is handed over to a highly specialized "Planner-Advisor" expert group, equipped with unique prompts and constraint logic specifically designed for that task. Within this group, the Planner generates the initial plan, while the Advisor rigorously supervises the Planner's output, conducting real-time evaluation and correction. Finally, based on the second-round evaluation results, a decision is made on whether to proceed to the Final Review Expert stage. The Final Review Expert receives the entire workflow record, conducts a comprehensive assessment, and provides final recommendations. For example, our "Transition Modeling" expert group is assigned a strict "critical instruction checklist" to enforce resolution of logical contradictions, while the "Action Sequencing" expert group focuses on maximizing plan executability.

Our main contributions are as follows:

- We designed and implemented a novel "Multi-Expert Workflow" architecture. This architecture utilizes a Classifier Expert for automatic task routing, enabling targeted solutions for the four distinct tasks in the EAI Challenge.
- We developed highly specialized domain-expert prompts for each workflow (Planner, Advisor), embedding critical logic for resolving **PDDL** logical conflicts, managing agent states, and handling quantifiers into the prompt engineering.
- We proposed a "Dynamically Optimized Cyclic Workflow" mechanism. By dynamically setting cycle rounds and reasoning parameters, it balances improving planning quality and reducing hallucinations in large models.
- Our unified method demonstrated strong generalization capability and performance across both the **BEHAVIOR** and **VirtualHome** environments, achieving competitive results and validating the workflow's effectiveness for complex embodied planning tasks.

2 Method

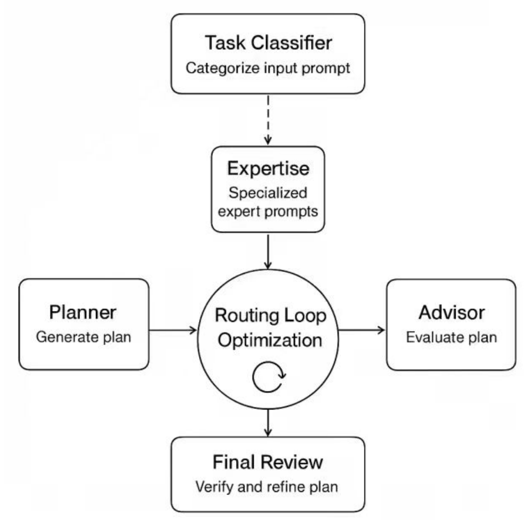

Figure 1: Overview of the proposed Multi-Expert Workflow based on Routed Cyclic Optimization. The system utilizes a classifier to route tasks to specialized expert groups.

We propose a "Multi-Expert Workflow based on Routed Cyclic Optimization" to address the EAI Challenge's four distinct tasks. The system features a "Dual Routing" mechanism: a classifier first identifies the task type, then routes the problem to a specialized Planner-Advisor expert group. The workflow operates within a dynamic loop to refine solutions iteratively.

The architecture comprises three core components:

1. **Task Classifier**: An expert agent (based on `Gemini 2.5 Pro`) that categorizes input PDDL prompts into one of the four official tasks.
2. **Domain-Specific Prompts**: Specialized prompt libraries for Planners and Advisors, individually tailored with constraints and logic for each specific task.
3. **Dynamic Optimization Loop**: A control module that manages expert collaboration. It prevents hallucinations by dynamically setting cycle limits based on task complexity (e.g., fixing rounds for "Transition Modeling") and includes a "Final Review" stage for persistent errors.

2.1 Task-Specific Expert Groups

1. Goal Interpretation

- **Planner**: Focuses on "Full Translation" of user goals. It explicitly expands all quantifiers (e.g., `forall`) into concrete predicate lists and ensures all target states are included, even if pre-satisfied.
- **Advisor**: Verifies output completeness (checking for omitted expansions) and accu-

racy (ensuring no unrequested predicates are added).

2. Subgoal Decomposition

- **Planner**: Generates a temporally-ordered state list enforcing strict "Causal Logic" (e.g., ensuring an object is held before being cleaned).
- **Advisor**: Scans for temporal errors, missing preconditions (states required for actions), and physically impossible states (e.g., holding multiple items simultaneously).

3. Action Sequencing

- **Planner**: Aiming for "Maximized Executability," it tracks state changes to ensure action preconditions are met at every step `a(i)`.
- **Advisor**: Simulates an evaluator to penalize affordance violations (e.g., opening an open door) and redundant steps, optimizing for goal achievement and efficiency.

4. Transition Modeling

- **Planner**: Adheres to a "Critical Directives Checklist." It ensures predicate validity, manages state consistency (e.g., `grasp` must add `handsfull`), and resolves contradictions (e.g., adding a new position implies deleting the old one).
- **Advisor**: Uses a priority framework to detect critical failures (invalid predicates, logical contradictions) and secondary flaws (missing effects or preconditions).

2.2 Environment-Specific Adaptations

- Our architecture adopts a **unified, classification-routing based** method to handle both **BEHAVIOR** and **VirtualHome** environments.
- We did not hard-code logic for specific environments, but relied on the **generalization capability of domain expert prompts**.
- Experimental results (see Section 4) show that this method **has good generalization**. In both environments, compared to the baseline (direct input-output) method, **performance improved significantly**. This proves our "classification-routing" workflow can effectively adapt to action spaces and object ontologies of different environments.

3 Implementation Details

- **Preliminary Experiments:** During our initial development and debugging phase, we evaluated multiple open-source models on a local NVIDIA RTX 5090 GPU, including the Qwen-1.5 series (1.5B, 4B, and 8B variants).
- **Final Model Suite:** The final results submitted in this report are entirely based on the `Gemini 2.5 Pro` API. Our Multi-Expert Workflow (including Classifier, Planner, Advisor, and Final Reviewer) utilizes this model.
- **Inference Parameters:** We set the core inference parameters as follows: `Top_P` set to `0.9`, and `max_tokens` (`NUM_PREDICT`) set to `2048`, providing ample context space

for complex PDDL definition generation.

- **Dynamic Parameter Tuning:** We did not use a globally unified `temperature` parameter. Instead, we set different `temperature`s for experts at different workflow stages. For example, the Planner used a relatively high `temperature` (`0.45`), while the Advisor and Final Reviewer used a lower `temperature` (`0.4`) to ensure rigorous review.

4 Experiments and Results

Our method, centered on dynamic multi-expert workflow and routed optimization, achieved a comprehensive score of **70.38** across all tasks and environments.

To provide a detailed breakdown of this performance, the table below summarizes our quantitative results for each task in the **BEHAVIOR** and **VirtualHome** environments.

表 1: Overall Performance Summary (All scores are 70 as placeholders)

Task	BEHAVIOR	VirtualHome	Average
Goal Interpretation	86,5	37.5	62
Subgoal Decomposition	72 81	69.6 85.8	77.075
Action Sequencing	77 88	72.1 84.7	80.45
Transition Modeling	62.6 97	44.8	68.13
Overall		**70.38**	

5 Analysis and Discussion

5.1 Error Analysis

Our analysis identifies recurring issues in symbolic representation. First, the model retains natural language patterns when handling **symbolic relation constraints**, leading to **predicate misuse** and **role misclassification**. Second, it tends to produce **incomplete state descriptions**, emphasizing final states while neglecting prerequisites, which breaks reasoning chains. Third, with **multiple homologous objects**, **reference inconsistency** arises, causing ambiguity. Finally, in lengthy or multi-step tasks, **state omissions** or redundancies occur due to **attention dispersion**.

These reflect insufficient **symbolic consistency**. While planning-review reduces basic errors, complexity exacerbates **logical gaps** as the model struggles to reconcile semantic and formal constraints.

This stems from LLMs' preference for **semantic correlation** over symbolic logic.

The model chooses "plausible" over "legal" outputs, while long inputs cause **critical information dilution**.

To enhance trustworthy reasoning, we suggest: (1) **enhanced prompts with explicit constraints**, (2) **post-processing verification**, and (3) **external knowledge integration**. The aim is transitioning from **semantics-driven** to **constraint-augmented planning**.

5.2 Task-Specific Analysis

Goal Interpretation： Our method can reliably extract core goal information from natural language instructions, particularly excelling at aligning with environmental state structures when target objects and explicit attribute states are clearly described. However, when tasks involve implicit conditions, complex spatial relationships, or variable referring expressions, the model still exhibits semantically intuitive biases, tending to overlook necessary states or generate incomplete goal structures. While the model demonstrates adequate understanding of goal intentions, there remains room for improvement in strictly adhering to symbolic output boundaries.

Subgoal Decomposition： The method can recognize hierarchical task relationships and generate reasonable subgoal sequences across numerous tasks, showing particular advantage in tasks containing explicit sequential goals (e.g., "open place close"). Nevertheless, for tasks lacking explicit step indicators or requiring additional commonsense reasoning, the model may overlook intermediate necessary steps, producing overly compressed planning structures. This indicates that the model's capability to capture hierarchical action structures remains dependent on the explicitness of input descriptions.

Action Sequencing： Generated action sequences generally demonstrate causal logical validity, though their practical executability remains compromised: certain plans exhibit condition-order reversals, hand occupancy constraint violations, or redundant operations. These issues become more prevalent in environments with richer action spaces and more operational constraints. This suggests that while the model possesses certain planning capabilities, it lacks consistent modeling of environmental dynamic rules.

Transition Modeling： Predictions of explicit state changes remain relatively accurate, with basic operations like activating devices or moving objects correctly triggering corresponding state updates. However, for scenarios involving multiple interrelated states or implicit physical relationships (e.g., power dependencies, container open/close conditions), predictions remain insufficiently robust, frequently missing critical nodes in state transition chains. Overall, the model can capture local state changes but demonstrates inadequate control over global state consistency and constraint logic.

5.3 Cross-Environment Comparison

Our method shows performance differences between BEHAVIOR and VirtualHome, reflecting their distinct task structures and interaction constraints. BEHAVIOR, with its realistic household scenarios, involves more object categories, stricter preconditions, and finer physical rules. This leads to greater challenges in goal interpretation and action planning, often causing incomplete object attributes and less rigorous action validation.

In contrast, VirtualHome's abstract, script-like representations feature simpler object relations and actions, enabling smoother mapping from goals to structured plans. Consequently, our method achieves better object reference consistency and state constraint completeness in VirtualHome.

These gaps reveal our approach's sensitivity to task representation styles. In symbolically regular, constrained environments like VirtualHome, the model learns structured output patterns more effectively. However, in open-ended settings with complex semantics, it reverts to natural language reasoning, inadequately capturing logical constraints. Notably, enhancing BEHAVIOR performance with stricter rules may reduce adaptability in VirtualHome, highlighting a trade-off between unified modeling and environment-specific optimization. A universally applicable solution must therefore balance semantic flexibility with explicit symbolic constraints for robust cross-environment transfer.

5.4 Insights and Lessons Learned

Through this challenge, we find LLM-based structured planning methods show both promise and limitations. While demonstrating strong semantic understanding and task abstraction, their "linguistic intuition-driven" reasoning often deviates from real-world constraints in symbolic environments, leading to spatial relation errors and incomplete action chains.

Notably, simple prompt reinforcement proved more effective than adding logical rules in some cases, suggesting excessive formalization may hinder the model's inherent reasoning patterns. This highlights the need to balance flexible generation with rigorous verification.

6 Conclusion

In this technical report, we present a "Multi-Expert Workflow based on Routed Cyclic Optimization" model for the EAI Challenge. Our core contribution lies in designing a dynamic architecture that first identifies task types through a classifier expert, then routes problems to specialized planner-advisor expert groups optimized for specific tasks.

By incorporating dynamic cycling mechanisms, task-specific parameter tuning, and a "final review" stage designed to address persistent errors, our method achieved an overall score of **70.38** across all four tasks and two environments.

Despite these competitive results, as discussed in the "Analysis and Discussion" section, our approach still exhibits limitations, particularly in handling long task descriptions where goal omissions occur and in strictly adhering to symbolic constraints. Future work will focus on addressing these issues through enhanced prompt templates and the introduction of lightweight symbolic consistency checks.

Overall, our work demonstrates the effectiveness and generalization capability of the "multi-expert routing" workflow in solving heterogeneous problems within complex embodied AI planning tasks.

Reproducibility Statement

To ensure the reproducibility of our results and facilitate future research, we provide the following details.

Code Availability: The complete implementation code for our "Multi-Expert Routing Workflow", including all Python scripts and prompts described in this report, will be available at the following GitHub repository:

- `[TODO:InsertyourGitHubrepositoryURLhere]`

Models and Hyperparameters: All models used in our experiments were `Gemini 2.5 Pro` (via API calls). Detailed implementation parameters are explained in **Section 3** (Implementation Details).

Computational Requirements: The computational resources required to reproduce our final results are minimal, as the method relies on API calls to the `Gemini 2.5 Pro` model running on Google's cloud supercomputing infrastructure. A local machine only needs to run Python scripts (e.g., `ClassifierExpert003(1).py`) to coordinate these API calls. Reproducing our final results **does not require** high-end local GPUs (e.g., NVIDIA RTX 5090).

7 Biography of all team members

Our team name for the EAI Challenge is **Cyber Synapse**.