COCORELI: Enforcing Execution Preconditions for Reliable Collaborative Instruction Following

Swarnadeep Bhar[1] Omar Naim[1] Eleni Metheniti[2] Bastien Navarri[2] Morteza Ezzabady[1] Loïc Cabannes[3]
Nicholas Asher[1,2,4]

[1] IRIT, Université de Toulouse [2] ANITI [3] ENS Paris-Saclay [4] CNRS

Motivation

Current collaborative LLM agents may detect missing information, yet still continue execution.

Reliable collaborative agents should couple missing-information detection with execution control.

Existing LLM Agents

Instruction
↓
Missing Information Detected or Missed
↓
Continue Execution
↓
Hallucinated Action

COCORELI

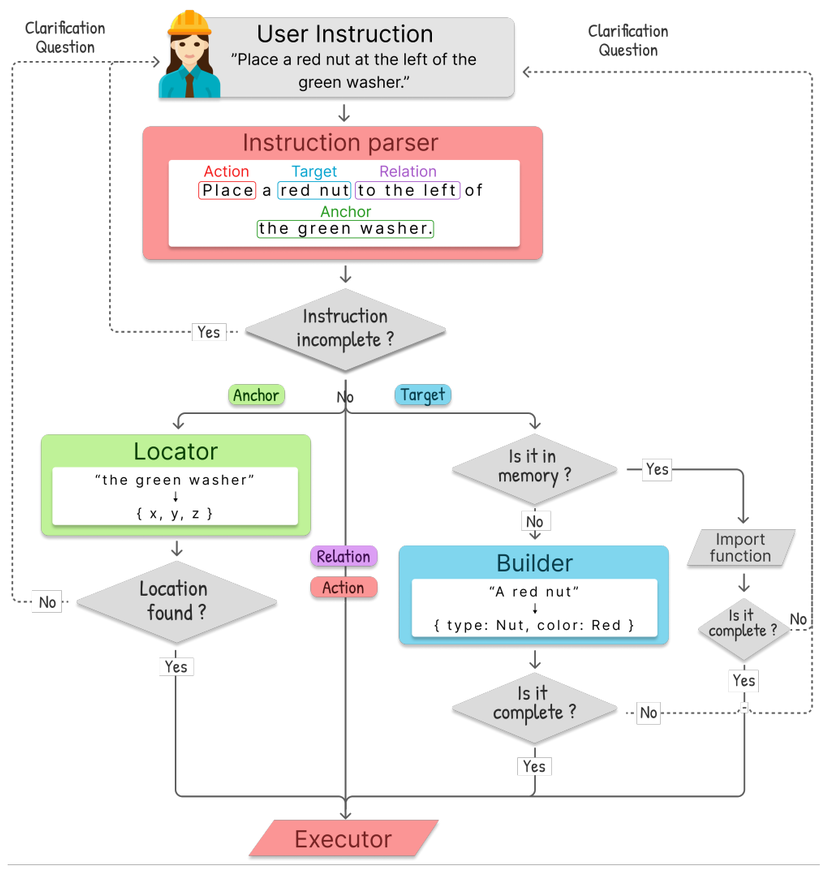
Instruction
↓
Detect Missing Information
↓
Block Execution
↓
Ask Clarification
↓
Execute

"Place a bolt at the top left corner."
What is the color of the bolt?

Detection is not enough.
Reliable collaboration requires coupling missing-information detection with execution preconditions.

COCORELI Framework

COCORELI combines specialized LLM modules with explicit execution preconditions for reliable collaborative instruction following.



Why a Modular Architecture?

Collaborative instruction following requires solving several distinct reasoning problems before execution. COCORELI decomposes these into specialized modules with explicit interfaces.

Reasoning Responsibility	Module
Identify target part	Instruction Parser
Resolve spatial location	Locator
Resolve part attributes	Builder
Manage collaborative dialogue	Discourse Module
Store & retrieve abstractions	Memory
Enforce execution preconditions	Executor

Rather than relying on a single generation step, COCORELI separates parsing, grounding, verification, memory retrieval, and execution into explicit reasoning stages.

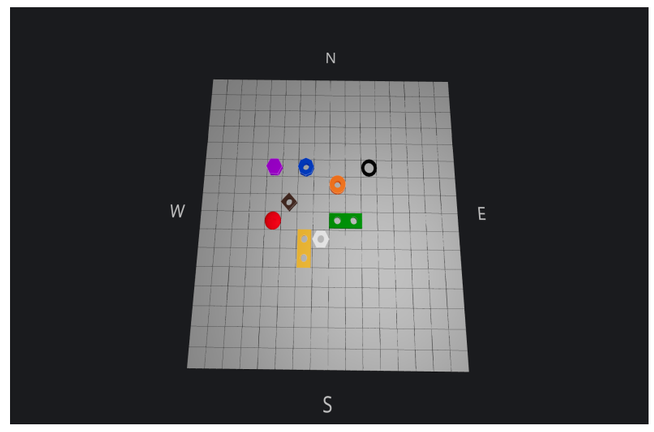
Underspecified Instructions

Instruction History

1. Place a red nut at the top left corner of the grid.
   ✓ *Red nut placed.*
2. Place a screw to the right of the nut.

Question
What is the color of the screw?

COCOBOTS Environment




```
part : screw                    ✓
location : right of nut         ✓
color : null                    ×
```

⇓

Ask Clarification

Reliable Clarification

	GPT	Claude	Agentic	ReAct	COCORELI
Detected	54.5	53.5	98.5	96.5	**96.5**
CQ Asked	–	–	98.5	96.5	**100**
Correct CQ	–	–	100	96.5	**100**
Part Hallucinations	100	100	97.5	0.5	**0.0**
Color Hallucinations	45.5	46.5	44.6	3.0	**0.0**
Location Hallucinations	100	100	78.2	1.5	**0.0**

*Results are reported for **two-instruction interaction histories**, where missing parameters may occur in either instruction and must be resolved using the dialogue context. Single-instruction cases are substantially easier for all models; the complete evaluation is reported in the main paper.*

Structured Retrieval & Generalization

The retrieval architecture remains unchanged across domains. Only the structured representation is replaced.

COCOBOTS Structural Abstraction

Shape	GPT	Claude	Agentic	COCORELI
Simple (4 shapes)	4/4	4/4	4/4	**4/4**
Medium (3 shapes)	3/3	3/3	1/3	**3/3**
Complex (2 shapes)	0/2	0/2	0/2	**2/2**

Shape complexity increases with the number of constituent parts.

ToolBench Transfer

System	P	R	F1
Claude 3.5	36.4	26.9	30.9
COCORELI	**100**	**100**	**100**

*COCORELI does not outperform baselines by reasoning harder. It reframes planning as **structured parsing + retrieval**. The same retrieval pipeline transfers from relational graphs (COCOBOTS) to API workflow graphs (ToolBench).*

Conclusion

Our results suggest that **reliable collaborative instruction following** depends not only on **model capability**, but also on architectures that **explicitly represent task requirements** and **enforce execution preconditions**.