CORNELL TECH · Cornell University · UNIVERSITY OF MARYLAND · KAIST · Imperial College London

COLM DAIH Workshop COLM 2026

Beyond Benchmark Accuracy

Failure Modes of Multi-Agent Coordination in Safety-Critical Clinical Workflows

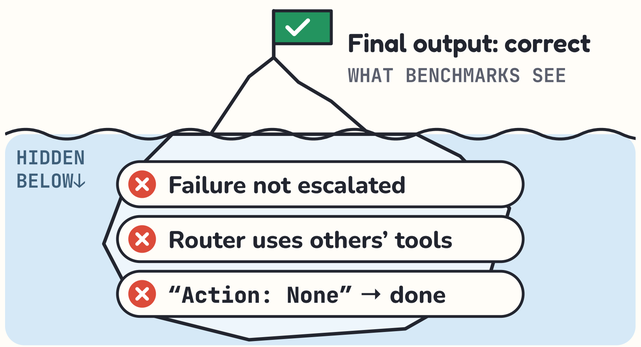
Sophie Yuanchen Bai[1,2] · Zijian Ding[3,4] · Shaoyue Wen[5] · Xiang Chang[1] · Angelique Taylor[1,2]

[1] Cornell Tech · [2] Cornell University · [3] University of Maryland, College Park · [4] Korea Advanced Institute of Science and Technology · [5] Imperial College London

MOTIVATION

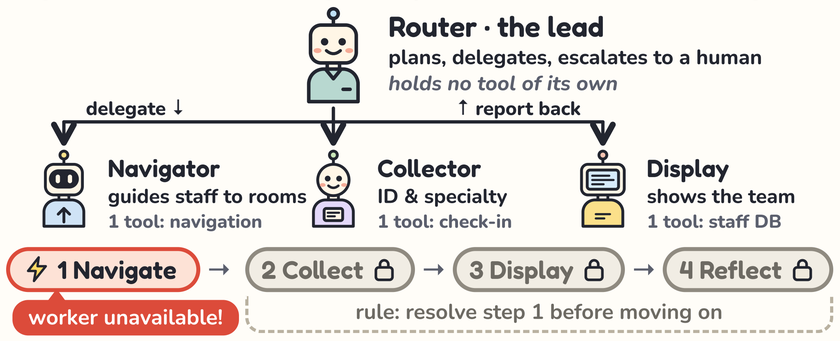
Seemingly right answer, broken process

Benchmarks see the output, not the process.



SETUP

Agents run emergency-room onboarding



every role is an LLM agent · scripted failure injection

STUDY 1 · KNOWLEDGE BASE

Give the team a handbook

GPT-4o team, with vs. without a knowledge base.

Process score (7 rubrics)		Failure handling	
without	45%	without	0%
with	73%	with	0%

Still failing with the handbook:

Tool misuse · Ignored reports · Role takeover · Breaks the workflow · False completion

Knowledge helps. Structure is the bottleneck.

STUDY 2.1 · AGENT TEAM STRUCTURE

Enforce two-way communication

router replies every step · agents report back

Structure redesign unlocks failure handling.

0% → 90%
failure handling (GPT-4o)
before → after redesign

STUDY 2.2 · MODEL REASONING

Swap GPT-4o for o3: smarter, and messier

GPT-4o VS o3

o3 did more of almost everything, good and bad.
Same team & task · 20 runs per model, 4 tasks per run

desirable · undesirable · GPT-4o · o3 · bar = runs out of 20

↓ **Study 2.2 up close: four coded behavior patterns** a run counts once if the behavior shows up in any of its 4 tasks · each behavior with one o3 example

1 Plans every step, then goes off-script

`Planning granularity & execution alignment`

	GPT-4o	o3	
Step-by-step planning	0	20	**Router** "...If no response after 2 min, coordinate with the charge nurse..."
Anticipates next task	1	6	**Router** starts Display's task on its own as soon as navigation is done.
Deviates from prompt	4	14	Prompt: show the full result. **Router** tells Display to show only part, and **Display** obeys.

2 Knows its limits, calls for help

`Task & organizational-role interpretation`

	GPT-4o	o3	
Clarifies its role	4	5	**Navigator** "Requested actions (... reassigning staff) fall outside the navigation agent's scope."
Recruits other roles	1	10	**Router** asks Display to post an alert: "CODE ASSIST– Patient in ER-12 requires immediate attention."
Escalates to a human	5	13	**Router** "I am escalating the issue to the human supervisor... ESCALATE."

3 Catches the error, then gets ignored

`Communication robustness & format compliance`

	GPT-4o	o3	
Drops the format	0	11	The tool returns the full ID data, but **Collector** replies only "`Task Status: success`"
Audits bad output	0	11	**Router** "The task is not yet complete: we still need the required JSON payload containing ID, Name, and Specialty..."
Ignores correction	0	9	**Router** sends an exact template; **Collector** still does not fix its reply.

4 Declares success without proof

`Task termination & verification`

	GPT-4o	o3	
Unverified success	2	11	The tool result only says worker #80 is unreachable; nothing mentions #82. Yet **Navigator** reports "Substitute HCW #82 located at Hallway A." and **Router** accepts it without verification.
Redoes finished tasks	1	20	**Router** reruns the finished reflection task again and again, with no reason; the reports don't improve.

1 Green checks can lie
Evaluate the trace against what the scenario requires.

2 Structure really matters
Who reports to whom shapes how failures get handled.

3 Smarter may not be safer
More planning can mean more ways to fail.

BEFORE CLINICAL USE

- Certify the process
- Log every escalation
- Define when clinicians step in

Scan for code & traces
rubrics · examples

byc-sophie.github.io · Sophie Yuanchen Bai · yb299@cornell.edu