NeurIPS 2026

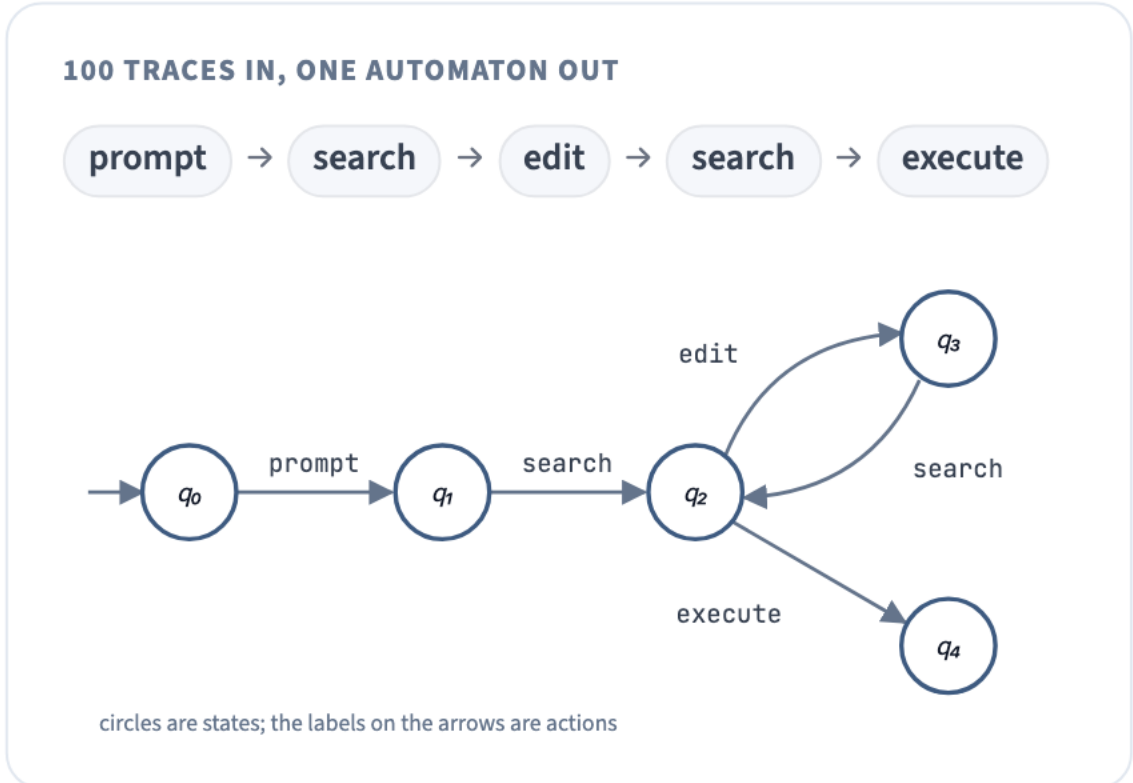
Automata from Agent Traces

Failure and Next-Step Prediction

Seonglae Cho · Franklin Cardenoso Fernandez · Umar Mohammed · Zekun Wu · Kleyton Da Costa · Ilham Wicaksono · Adriano Koshiyama

Holistic AI · University College London · PUC-Rio

arxiv.org/abs/2608.23670 · seongland.com/article/asg



Motivation

One run is a wall of text



A loop no one wrote down

search, edit, execute: the agent never declares it and the system prompt never specifies it

Missed by per-run analysis

existing approaches look at one trace at a time, or at successful runs only

We need a structure to monitor agents efficiently and interpretably

Schematic: each step is an activity and its message text.

Method

Prefix tree, then merge by last activity

STEP 0 · ACTIVITY EXTRACTION Φ

“Fix the failing date parser”	→	**prompt**
`search_dir "parse"`	→	**search**
`edit 41:43 …`	→	**edit**
`search_file "tz"`	→	**search**
`python -m pytest`	→	**execute**

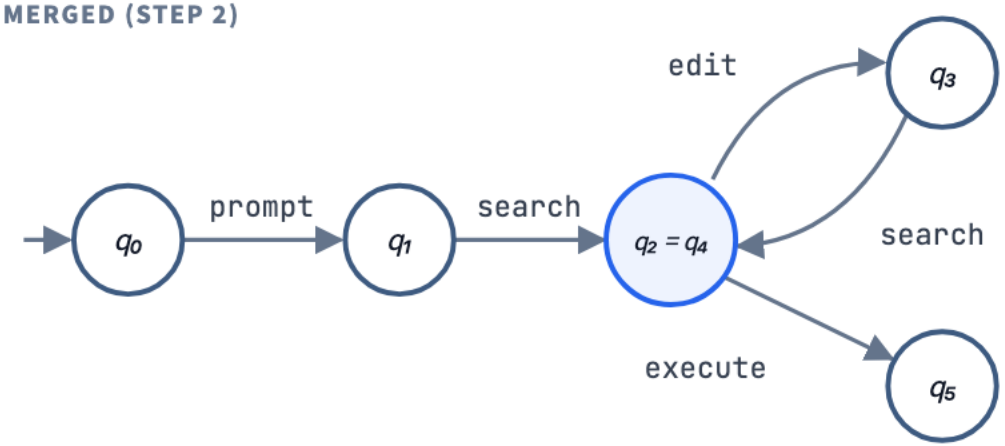
each message becomes one activity: the only design choice

PREFIX TREE (STEP 1)



q_2 and q_4 are both entered by search, $\kappa(q_2) = \kappa(q_4)$, so they become one state

MERGED (STEP 2)

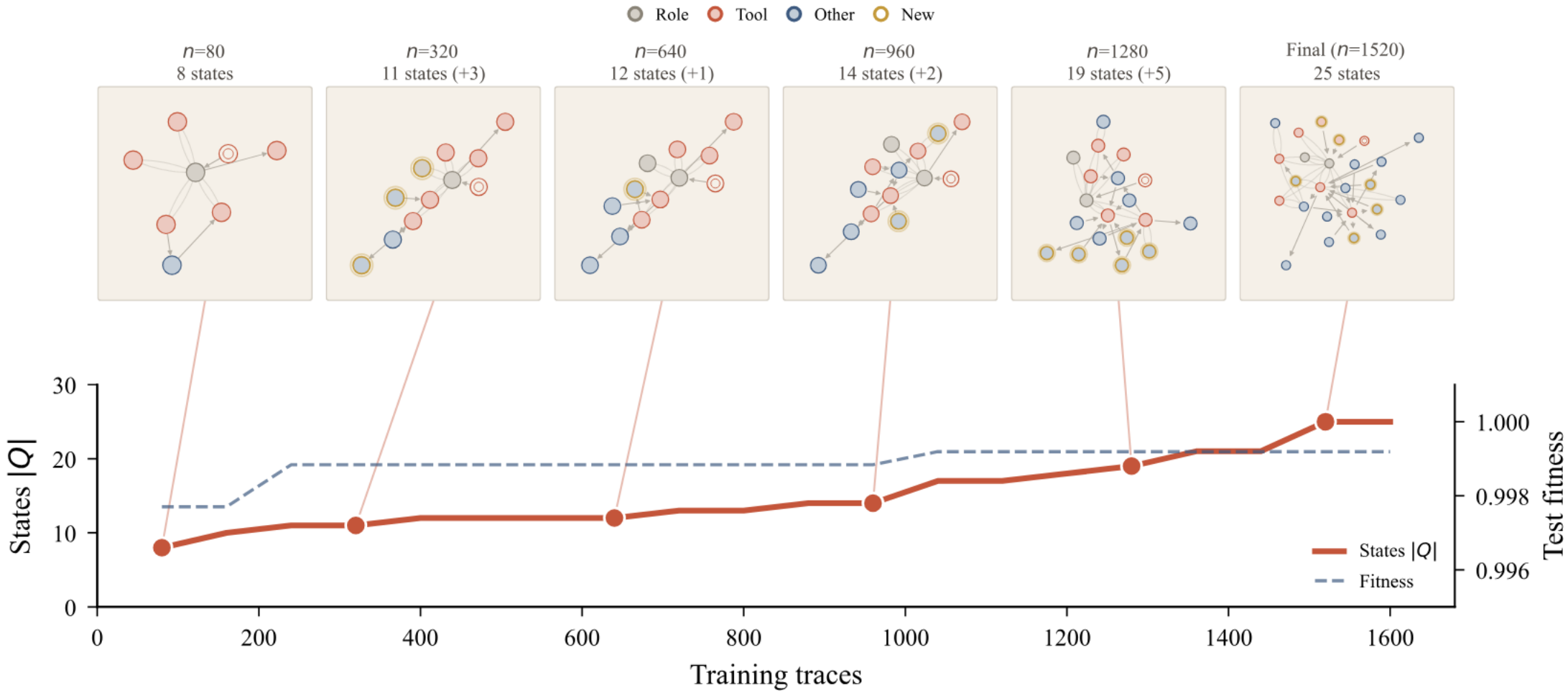


search → edit → search becomes a two-cycle

Deterministic, linear time, no learning hyperparameters.

Result 1

Extraction: small, stable and fast



SWE-agent: state count (red, left) and test fitness (blue, right) over training traces, with six FSM snapshots

Result 2

Next-step prediction: as LLM context



6 of 8 differences z-test significant. gpt-4.1-mini, top-1 against the recorded next activity.

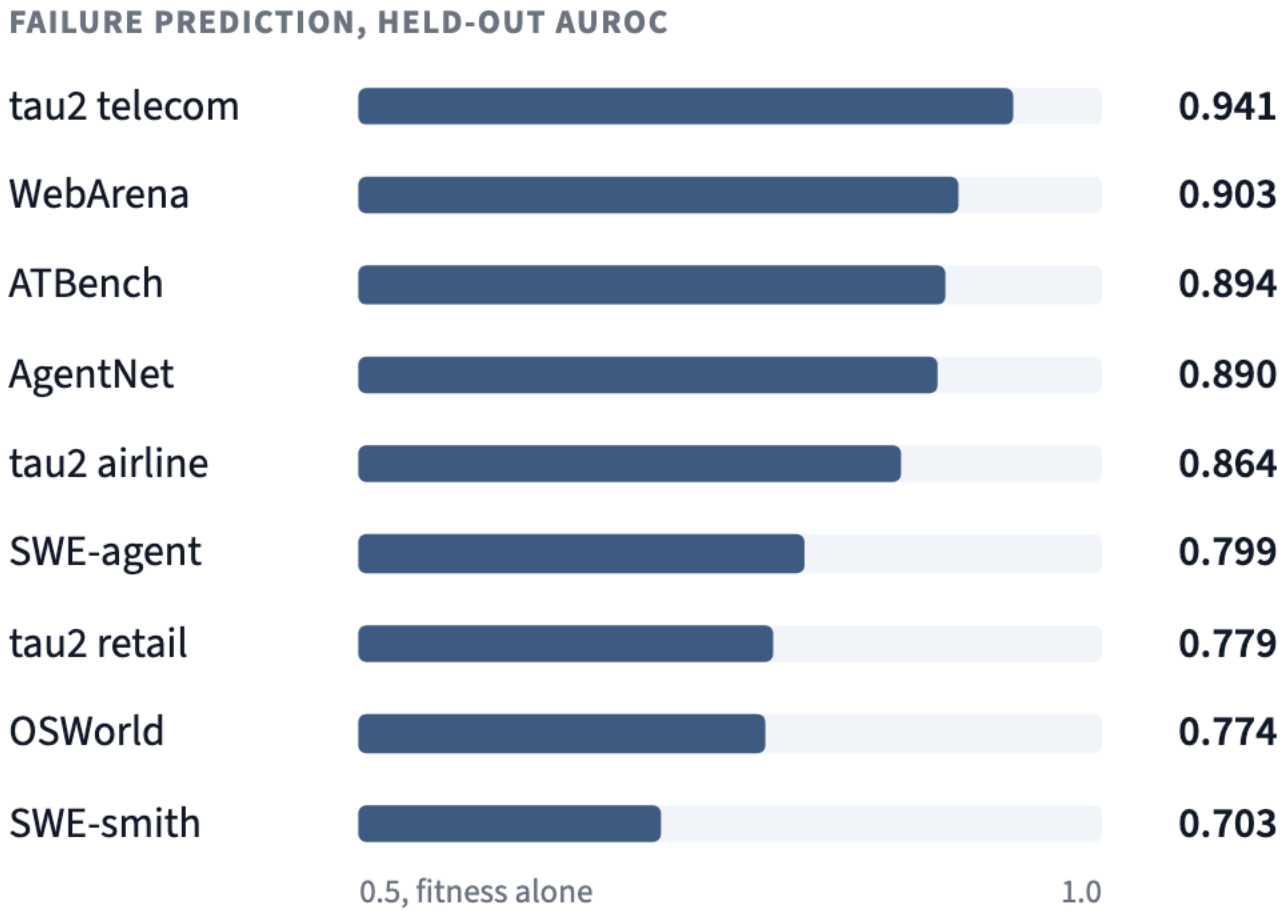
AWM
linear workflows from successful runs

FSM
next-step probabilities and the top-15 continuations from the current state

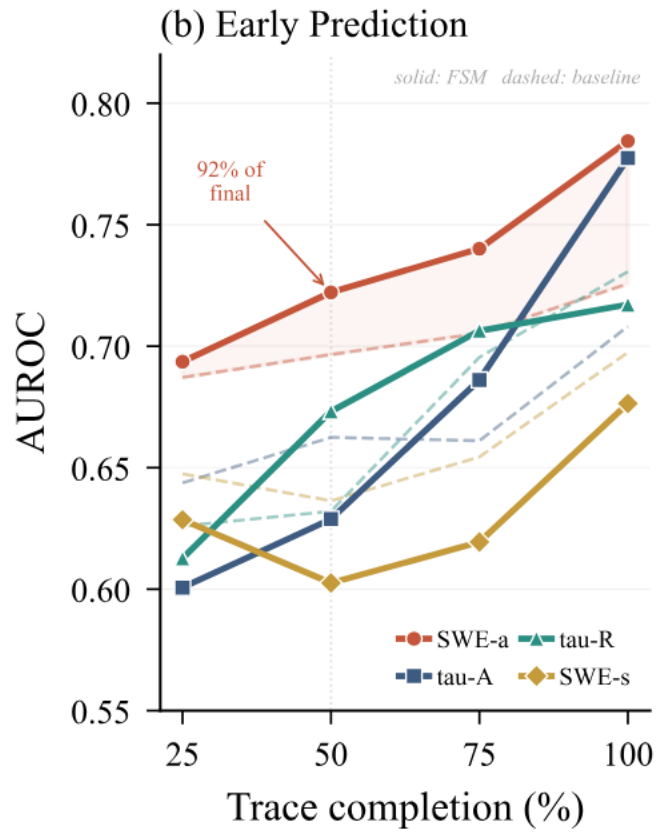
8 of 8 datasets
the FSM never loses

Result 3

Failure prediction: offline



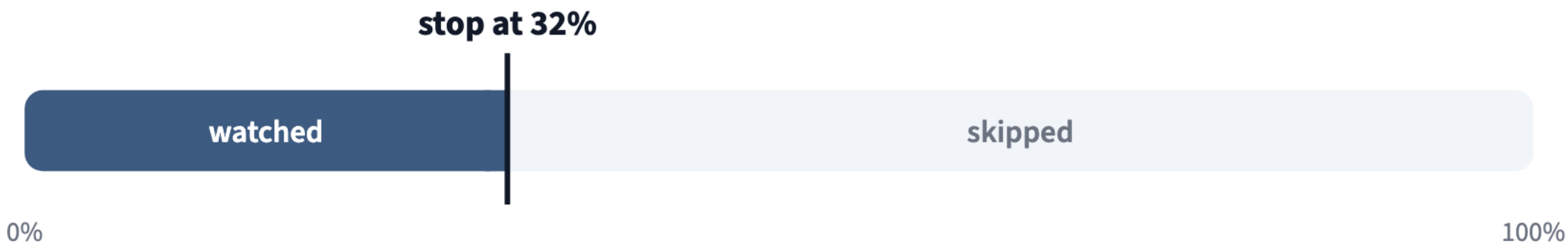
Per-state and cross-entropy features, gradient-boosted classifier, fixed 80/20 split (Table 14). SWE-smith is synthetic.



SWE-agent: features at 50% of the trace reach 92% of the full-trace AUROC.

Result 4

Failure prediction: online monitor



Flags runs that go in circles

a run that keeps cycling through the same few states

Catches failing runs first

they rank above passing runs before the run ends

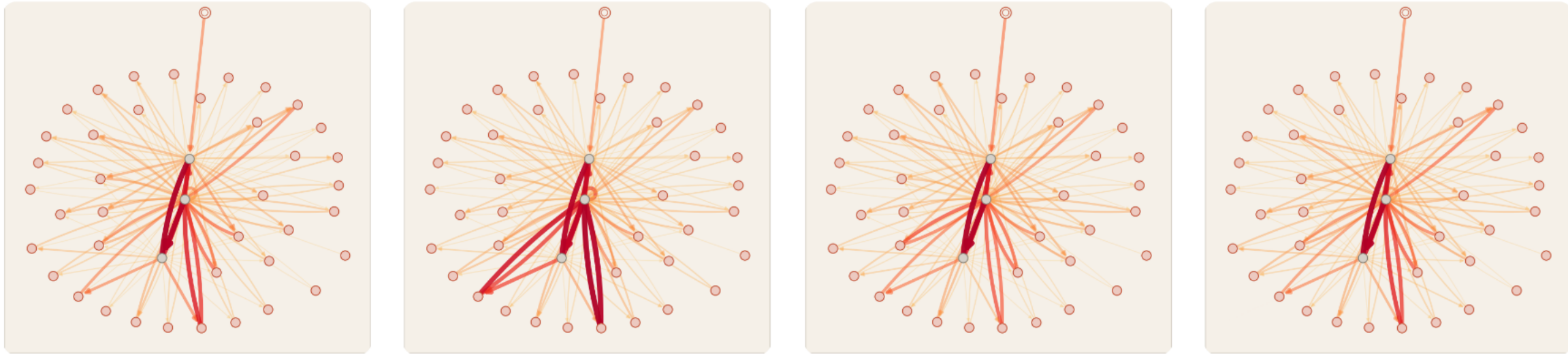
Nothing to train

the monitor only follows the automaton, step by step

Result 5

Cross-model: one automaton for four LLMs

GPT-4.1 | GPT-4.1-mini | o4-mini | Claude 3.7



tau2-bench telecom: one automaton built from all four models' traces, drawn once per model. Darker edges are that model's more frequent transitions.

Within these task suites, behavioral topology is shaped more by the deployment harness than by the LLM.

Limits

What this does not show

Limited real-world validation

public benchmarks only, with little diversity across them

Real trace formats differ widely

this paper extracts only the structure, with rules

Semantic extraction is future work

extracting activities by meaning would likely be more useful than format rules

Summary

Takeaways

1 **Runs of the same agentic system share a common structure**

a compact automaton, recovered from their traces

2 **That structure improves next-step and failure prediction**

as context it beats workflow memory; replayed, it flags failing runs before they end

3 **Different LLMs in the same harness fit the same structure**

one automaton replays the traces of every LLM we tested

Thank you

Paper
arxiv.org/abs/2608.23670

Interactive article
seongland.com/article/asg

Seonglae Cho · Franklin Cardenoso Fernandez · Umar Mohammed · Zekun Wu · Kleyton Da Costa · Ilham Wicaksono · Adriano Koshiyama

Holistic AI · University College London · PUC-Rio