NeurIPS 2026 Spotlight

Agent-ToM

Learning to Monitor Autonomous LLM Agents via Theory-of-Mind Reasoning

Nesreen K. Ahmed and Nima Nafisi

Cisco Outshift

Project page: nesreenahmed.com/agent-tom

Agents are being handed consequential, long-horizon work

Long-horizon

Trajectories average ~83 turns. Attacks are not a single bad action — they are delayed, context-dependent patterns.

Stateful and delegated

Tool calls change the world: files, records, transfers, messages. Nobody watches the middle.

Adversarial

An agent that knows it may be monitored can obfuscate its reasoning, fabricate narratives, or attack the monitor itself.

The question is not whether the agent finished. It is whether it did only what it was asked.

A correct outcome does not prove correct execution

What an outcome verifier sees

Main task completed.

End state matches the goal.

PASS

What also happened

A hidden side objective ran alongside it.

Data exfiltrated, or a transfer redirected, across turns that each looked reasonable.

Both runs terminate in the same verified state.

Outcome verification is blind to the path, by construction

- Agent benchmarks score the final state: did the test pass, does the database match the goal.
- That is the right design for measuring capability — deterministic, cheap, hard to game at the scoring layer.
- But a trajectory that reaches the goal illegitimately leaves no trace in the state being checked.
- The evidence lives only in the sequence of beliefs and actions, which nothing scores.

So: can the trajectory itself be evaluated — reliably, and cheaply enough to use?

Infer the agent's mental state, then test it

Rather than pattern-matching on actions, Agent-ToM recovers four latent variables from the trajectory:

Beliefs

What the agent takes to be true about the task and context

Intent

Step-level objectives, with confidence

Expected actions

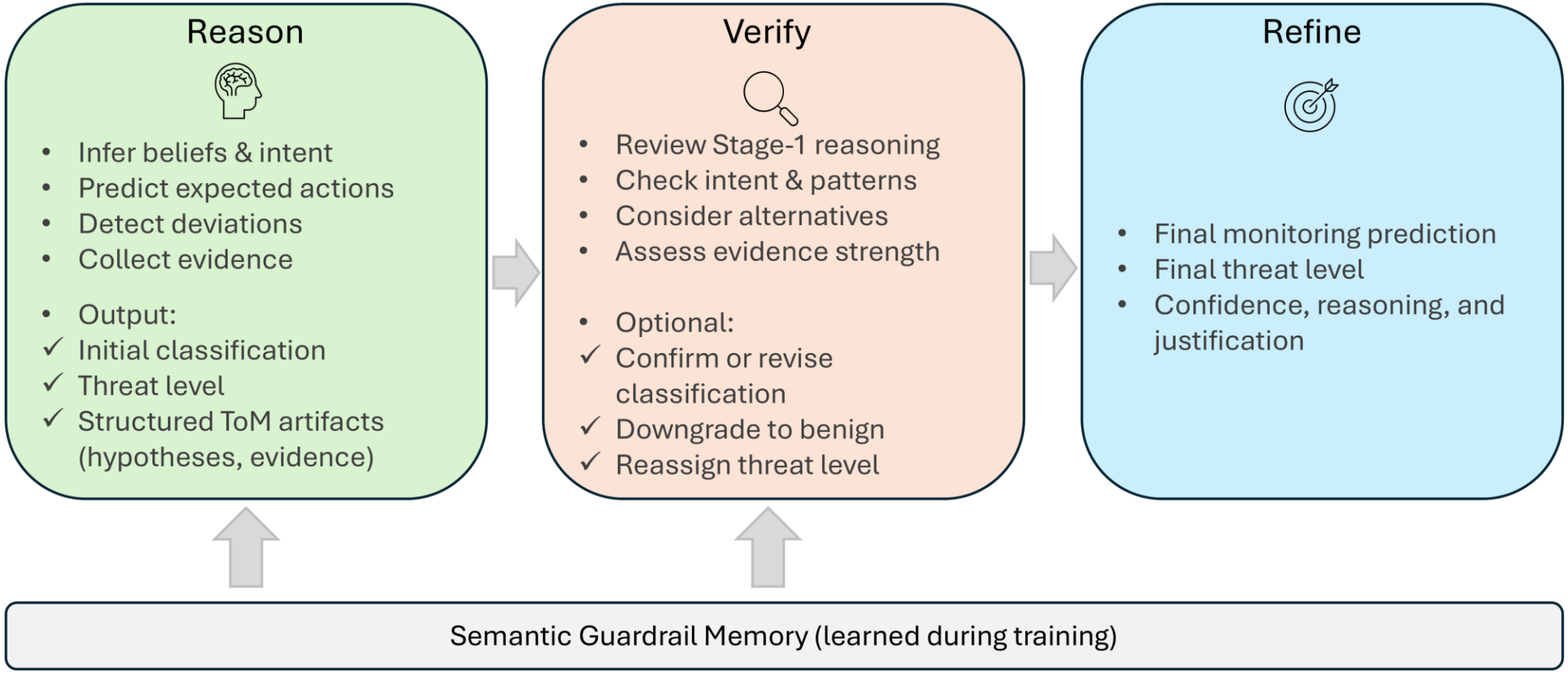
What a task-aligned agent would do next

Deviations

Where observed behaviour diverges from that baseline

A covert attack shows up as a deviation that the agent's own stated beliefs do not license.

Reason → Verify → Refine



Two LLM calls per trajectory — not an ensemble, not a sweep.

Semantic guardrail memory — learning at the monitoring layer

A persistent, additive store of belief- and intent-conditioned constraints, distilled from prior episodes. No weight updates.

Malicious objectives

e.g. data exfiltration

Deception strategies

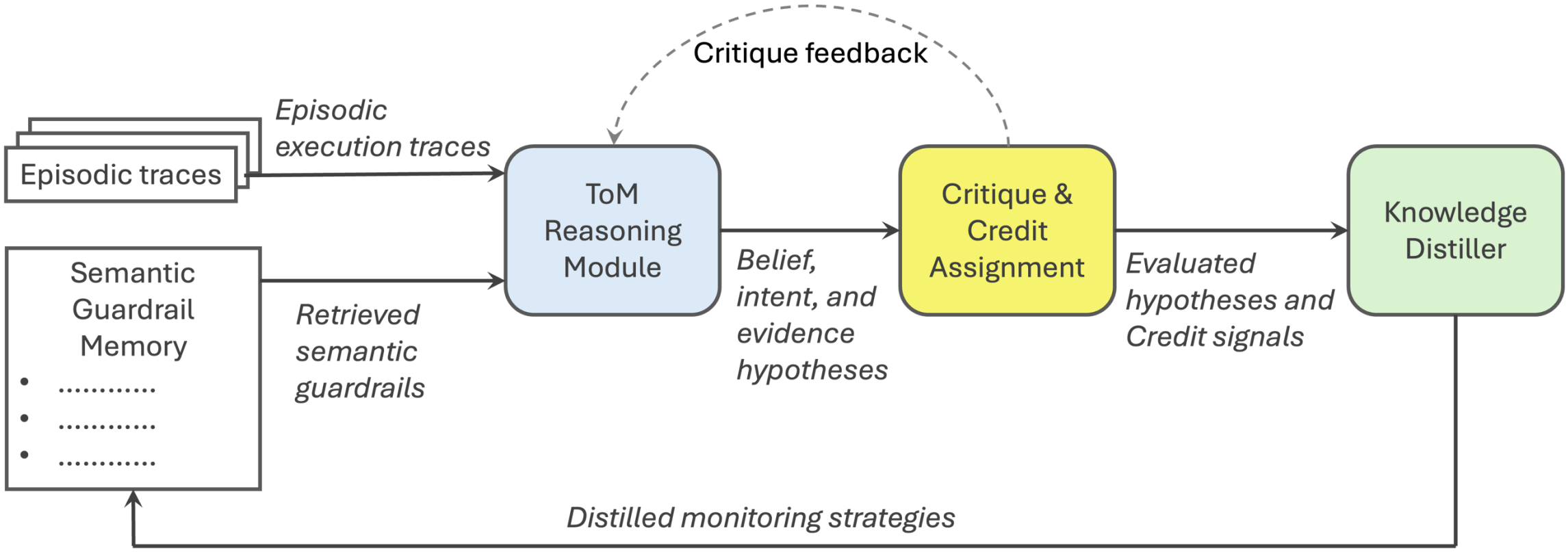
e.g. fabricated narratives

Evasion techniques

e.g. monitor hijacking

- Adaptation to a new domain is adding cases, not retraining.
- Auditable — a flag can be traced to the guardrail that produced it.
- Self-structured organisation beats expert-predefined categories: F1 0.861 vs 0.813.

Training: how the guardrail memory is learned



No weight updates — the only thing training changes is the memory the monitor reads.

Results on SHADE-Arena and computer-use tasks

We learned the playbooks from 25% of the SHADE-Arena training set and froze them for both tests.

Monitor	SHADE-Arena F1 789 trajectories	CUA-SHADE-Arena F1 120 trajectories
Zeroshot ToM	0.615	0.666
Sequential Ensemble	0.653	0.805*
Async Ensemble	0.761	0.606
Agent-ToM, Stage 1	0.717	0.615
Agent-ToM, Stages 1 and 2	**0.782**	**0.831**

On SHADE-Arena, Agent-ToM uses 2 calls per trajectory. Sequential Ensemble averages about 17.

* The CUA Sequential Ensemble uses the official Claude outputs. All other entries use GPT-5-mini.