Towards Design of an Automated Judge for Multi-Agent Systems

Evaluating MAS via Execution Traces without Labeled Data

Alina Zhidkovskaya*[1], Kirill Rapatskikh[1], Jerzy Kamiński[1], Grigorii Barakhsin[1], Anna V. Kalyuzhnaya[1], Alexey Druzhinin[2], Andrey Savchenko[2], Julia Belikova[2], Konstantin Polev[2], Nikolay Nikitin[1]

[1]ITMO University, Saint Petersburg, Russia; [2]Sber AI Lab, Moscow, Russia
*Email: alina.zhdk@gmail.com

What is AutoPumpkin?

- An automated ensemble of LLM-based judges for multi-agent systems
- Uses execution traces (OpenTelemetry)
- No labeled data required
- Works with any MAS architecture

AutoPumpkin Architecture

Two-Tier LLM Metrics Framework
AutoPumpkin deploys specialized LLM agents across two orthogonal levels for comprehensive MAS assessment via execution traces.
Agent-Level Metrics: per-agent task completion, observation accuracy, reasoning consistency, appropriate tool usage, correct tool parameters.
System-Level Metrics: overall task completion, effective role allocation, task transfer, design complexity, rule compliance.
AutoPumpkin Judge: SystemTaskCompletion + MASComplexity + ToolSelection -> binary Success/Fail (F1 = 0.85).

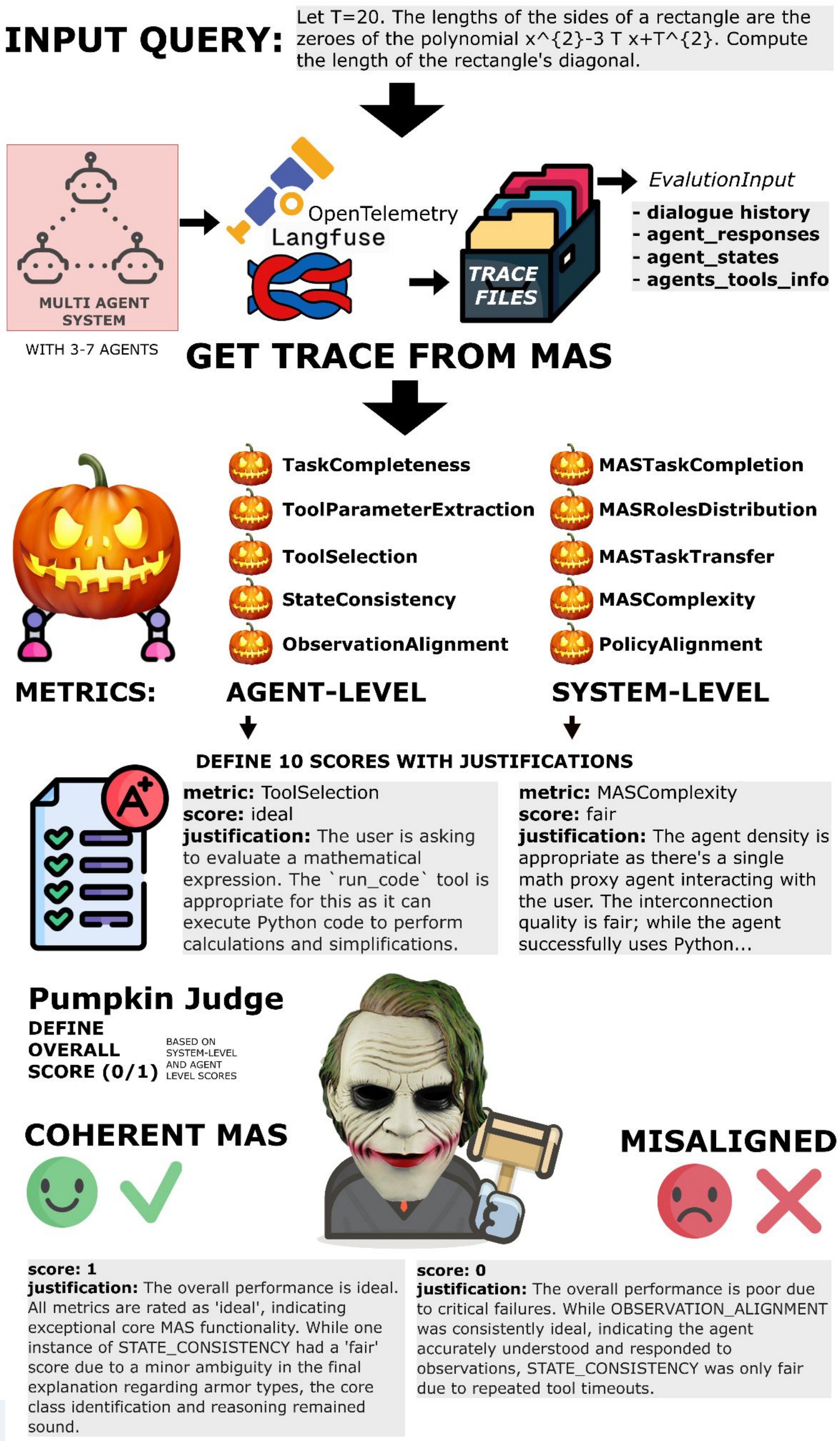
Performance Comparison

On our own GAIA-based dataset of 328 traces, **AutoPumpkin reaches F1 0.65, while TRAIL scores 0.634**, so both judges perform similarly on our tasks. This shows that our 3-metric ensemble is already reliable in-distribution before any cross-benchmark testing. On the TRAIL dataset (111 traces), AutoPumpkin then clearly pulls ahead, **achieving F1 0.85 vs 0.71** at threshold 2.5 and 0.75 vs 0.579 at threshold 3.0, without any extra tuning for this benchmark.

AutoPumpkin generalizes across datasets, maintaining competitive performance when evaluated on unseen benchmarks.

Threshold	TRAIL	AutoPumpkin
< 2.5	**0.71**	**0.85**
< 3.0	0.579	**0.75**
< 4.0	**0.58**	0.43

Table 1. Evaluation on TRAIL dataset. AutoPumpkin judge outperforms TRAIL baseline at decision boundaries < 2.5 and < 3.0.



INPUT QUERY: Let T=20. The lengths of the sides of a rectangle are the zeroes of the polynomial x^{2}-3 T x+T^{2}. Compute the length of the rectangle's diagonal.

- dialogue history
- agent_responses
- agent_states
- agents_tools_info

metric: ToolSelection
score: ideal
justification: The user is asking to evaluate a mathematical expression. The `run_code` tool is appropriate for this as it can execute Python code to perform calculations and simplifications.

metric: MASComplexity
score: fair
justification: The agent density is appropriate as there's a single math proxy agent interacting with the user. The interconnection quality is fair; while the agent successfully uses Python...

Pumpkin Judge
DEFINE OVERALL SCORE (0/1)
BASED ON SYSTEM-LEVEL AND AGENT LEVEL SCORES
COHERENT MAS
MISALIGNED

score: 1
justification: The overall performance is ideal. All metrics are rated as 'ideal', indicating exceptional core MAS functionality. While one instance of STATE_CONSISTENCY had a 'fair' score due to a minor ambiguity in the final explanation regarding armor types, the core class identification and reasoning remained sound.

score: 0
justification: The overall performance is poor due to critical failures. While OBSERVATION_ALIGNMENT was consistently ideal, indicating the agent accurately understood and responded to observations, STATE_CONSISTENCY was only fair due to repeated tool timeouts.

Fig. 1. AutoPumpkin pipeline

Novel Benchmark & Dataset

- 328 GAIA traces (163 large + 165 small MAS)
- Each = unique auto-generated MAS for specific task
- OpenTelemetry logs + manual success/fail labeling by GAIA

This setup lets us study how well different judges correlate with actual task success, rather than just intermediate behaviors.