
SafetyBench

Hendrik Wirthwein
TU Berlin

With
Apart Research

Abstract

1. Introduction

The proliferation of large language models (LLMs) has enabled increasingly sophisticated multi-agent systems, yet evaluating these systems remains challenging: traditional rule-based simulations lack behavioral realism, while fully LLM-driven approaches suffer from computational costs and reproducibility issues. Recent work demonstrating strategic deception in LLM agents playing Diplomacy and scenario forecasting efforts like AI 2027 highlight the urgent need for controlled frameworks to evaluate multi-agent AI behaviors systematically. This work introduces **apart**, a hybrid multi-agent orchestration framework that addresses these challenges through an architecture combining configuration-driven game mechanics with LLM-powered simulation engines. The framework employs an LLM-based SimulatorAgent that interprets natural language agent responses into realistic state changes while enforcing explicit constraints, preserving computational efficiency while enabling emergent behaviors. A critical innovation is the benchmarking infrastructure enabling systematic comparison of heterogeneous agent-model combinations (e.g., Google Gemini, OpenAI, local Ollama models) across configurable scenarios, addressing the fundamental research question: how do model capabilities influence individual and collective agent dynamics?

To address AI safety concerns highlighted by recent findings of manipulation and deception in competitive multi-agent scenarios, apart incorporates a post-hoc behavioral analysis system detecting potentially dangerous patterns through hybrid detection: lightweight signal collection during simulation followed by LLM-as-judge evaluation across three categories—power-seeking, deception, and rule exploitation. This design enables comprehensive safety analysis with minimal performance impact (<5ms per step) while providing quantitative scoring (0-10 scale) for

systematic comparison. The framework's architecture reflects five principles: separation of concerns, type safety (Pydantic validation), configuration-driven behavior (YAML scenarios), fail-fast validation, and extensibility, enabling apart to serve as both a research platform for studying multi-agent dynamics and a practical tool for evaluating LLM-based agent systems under controlled conditions.

2 Methods

2.1 Universal Simulator Architecture

Our framework employs an LLM-powered universal simulator that dynamically generates arbitrary multi-agent environments from declarative specifications. Unlike traditional approaches requiring custom simulation code for each scenario, our **SimulatorAgent** acts as a general-purpose game master that interprets scenario definitions and produces realistic state transitions through structured reasoning.

Key Innovation: Researchers define scenarios through simple YAML configurations specifying world variables, constraints, agent roles, and narrative objectives. The simulator maintains context through a sliding window of recent history, generates JSON-formatted state updates reflecting causal relationships between agent actions and environment dynamics, enforces constraint boundaries, and creates personalized narrative messages for each agent. This architecture decouples scenario design from implementation, enabling rapid prototyping of diverse experimental conditions—from economic simulations to social dilemmas—without programming expertise.

2.2 Scenario Creation Flexibility

The universal simulator architecture enables AI-assisted scenario generation through natural language descriptions. Researchers can rapidly create, modify, and parameterize scenarios using human-readable YAML specifications that define variable schemas (global and per-agent), agent configurations with custom system prompts, and optional scripted events. This declarative approach significantly lowers barriers to experimental design while maintaining consistent evaluation methodology across heterogeneous scenarios, facilitating fair cross-domain comparisons of AI agent behavior.

2.3 Hybrid Danger Detection System

We developed a two-stage detection approach balancing computational efficiency with analytical depth:

Stage 1: Lightweight Runtime Signals. During simulation, a **SignalCollector** performs non-invasive behavioral monitoring extracting predefined danger signals without LLM inference (<5ms overhead per step). We track three behavioral categories:

- **Power-seeking:** Resource accumulation rates (>15% per step) and dominance language patterns

- **Deception:** Linguistic markers of concealment and misdirection
- **Rule exploitation:** Variable proximity to constraint boundaries ($<1\%$ from limits)

Stage 2: Post-Hoc LLM Judge. Following simulation completion, a DangerAnalyzer employs an LLM-as-judge paradigm for comprehensive assessment. This approach leverages frontier model reasoning capabilities while avoiding computational burden during simulation. The judge receives complete agent response histories, aggregated behavioral signals, role descriptions, and simulation context, then provides structured scores (0-10 scale) across all danger categories with brief justifications.

Key Innovation: Our hybrid approach combines the efficiency of rule-based signal detection with the nuanced interpretive capabilities of LLM evaluation, enabling scalable deployment across extensive benchmark suites.

2.4 Implementation Details

The judge uses Google's Gemini 2.5 Flash with JSON-mode output formatting ensuring structured responses. The system records raw timestamped signals, per-agent danger scores across categories, and evaluation metadata. Results integrate into comprehensive benchmark reports with interactive HTML visualizations featuring severity-coded danger score tables. The entire system operates as an opt-in modular component with minimal overhead: $\sim 1\text{-}5\text{ms}$ per simulation step and single LLM call per agent post-simulation.

3 Results

Due to token limit constraints we were unable to run a comprehensive simulation. Feel free to clone the repo, follow setup instructions and test!

4 Discussion and Conclusion

This project demonstrates a practical framework for automated safety testing of AI systems, specifically targeting power-seeking and deceptive behaviors through comprehensive evaluation protocols. The architecture's inherent flexibility in scenario generation serves as a robust defense against benchmark gaming, ensuring that safety assessments remain reliable indicators of genuine model behavior rather than optimized performance metrics. The proposed implementation model envisions a commercial software-as-a-service platform that provides confidential pre-deployment safety evaluations, creating market incentives for developers to proactively address alignment concerns before public release. By maintaining proprietary benchmark specifications and offering early-access testing services, such an enterprise would enable responsible acceleration of AI capabilities while channeling development efforts toward safer outcomes. This business model represents a novel approach to alignment research sustainability, wherein commercial revenues directly fund and incentivize safety improvements, effectively transforming each dollar of enterprise value into tangible progress toward robust AI alignment. The economic incentives align naturally with safety objectives:

developers gain reputational protection and risk mitigation through private assessment, while the broader ecosystem benefits from systematically safer model deployments without impeding innovation velocity.

5 References

<https://ai-2027.com/>

<https://every.to/diplomacy>

6 Appendix

6.1 Limitations

Our approach has several notable constraints. The universal simulator architecture, while flexible, relies on LLM-generated state transitions that may deviate from ground-truth dynamics in domains with well-defined physics or game-theoretic equilibria. Signal-based danger detection uses heuristic thresholds (e.g., 15% resource growth, 1% boundary proximity) that may not generalize across diverse scenario types, potentially causing false positives in legitimate competitive behavior or false negatives for sophisticated manipulation strategies. The LLM-as-judge evaluation inherits known limitations of frontier models including inconsistent scoring across semantically equivalent behaviors, sensitivity to prompt phrasing, and potential bias toward verbose or well-structured agent responses. Critically, the judge model itself may be compromised, biased, or less capable than the agents under evaluation—a judge from a previous generation of models may fail to recognize sophisticated dangerous behaviors exhibited by more advanced agents, or could intentionally downplay risks if the judge itself exhibits alignment failures. Additionally, our three-category taxonomy (power-seeking, deception, rule exploitation) represents a preliminary framework that may not capture the full spectrum of dangerous AI behaviors, particularly emergent risks in novel deployment contexts. Finally, post-hoc evaluation cannot prevent dangerous behaviors during simulation, limiting applicability to real-world deployment scenarios requiring real-time intervention.