Antelligence: LLM-Powered Emergence in a Simulated Foraging Ant Colony

What are the questions asked?

This project addresses a central question in decentralized AI: How can large language models (LLMs) be integrated into multi-agent systems to facilitate emergent, intelligent behavior without a centralized command? Specifically, it investigates how LLM-driven agents can coordinate, adapt, and solve collective problems using language-based reasoning and indirect communication, drawing a parallel to biologically-inspired systems like ant colonies. The research also explores the performance tradeoffs between LLM-powered, rule-based, and hybrid agent colonies in a dynamic environment.

What is the key policy issue or theory being addressed?

The key theoretical framework addressed is **swarm intelligence** and **stigmergy**, the principle that agents can coordinate through environmental modification rather than direct communication. By augmenting traditional foraging models with LLM decision processes, the project explores how language-based reasoning can act as a new form of communication and a mechanism for emergent behavior. The project also touches upon distributed AI and sociotechnical systems theory by examining how intelligent agents interact within a simulated ecosystem under real-time constraints. A key policy issue is the balance between the computational cost of LLM inference and the performance gains of intelligent, adaptive agent behavior.

What is the key methodology or methodologies used?

The project's methodology is a **simulation-based approach** using a custom-built, agent-based model (ABM) in Python. It uses a comparative design to evaluate the performance of three agent types:

1. **LLM-Powered Worker Ants:** Use real-time prompts to an LLM API for decision-making.
2. **Rule-Based Worker Ants:** Follow deterministic, heuristic algorithms.
3. **Hybrid Colonies:** A mix of both LLM-powered and rule-based agents, often with a Queen Overseer to provide high-level coordination.

The simulation incorporates a bio-inspired digital pheromone grid for indirect communication and integrates blockchain smart contracts for persistent, auditable memory.

Describe your dataset and how to access it?

The project's dataset is generated synthetically by the simulation itself. It consists of logs of agent decisions, food collection events, API calls, and pheromone intensities. The data is not a pre-existing static set but is created dynamically during each simulation run based on the configured parameters. This data is logged on-chain via Ethereum-compatible smart contracts, providing an auditable and transparent record of all events. As such, the dataset is accessible through a secure Python client that interacts with the Sepolia testnet where the smart contracts are deployed. The code for generating and accessing this data is available as part of the project's codebase.

What tools were used to analyze the data?

The simulation itself provides real-time visualization and analysis. The core tools used include:

- **Python:** For all core simulation logic.
- **NumPy, Matplotlib, and PIL:** For handling simulation logic, data analysis, and generating visual output like GIFs.
- **web3.py:** For integrating with the Ethereum blockchain to log and retrieve data from the smart contracts.
- **Asyncio and AIOHTTP:** For efficiently handling parallel API calls to the LLM.

Any novel contributions or findings?

The project makes several novel contributions:

- **Hybrid LLM + ABM Architecture:** It introduces a Python-based, prompt-driven behavior system that leverages LLMs within a multi-agent model, moving beyond traditional NetLogo frameworks.
- **Blockchain Integration for Memory:** The use of blockchain smart contracts for simulating a persistent, auditable memory in a multi-agent system is a unique contribution.
- **Bio-inspired Pheromone System:** It successfully implements a pheromone signaling system that LLMs can interpret and use for decision-making.

Key findings include:

- LLM-only agents show superior adaptive capabilities (e.g., re-routing when food density changes) compared to rule-based agents, which often stagnate.
- Hybrid colonies offer the best balance between foraging efficiency and computational cost, demonstrating a viable path for scalable, intelligent systems.
- The presence of a prompt-driven "Queen" overseer can significantly improve a colony's ability to recover from unexpected events.

Who is the team? Provide names, emails, and institutions.

The team consists of:

- Kashyap Nadendla (University of Arizona, kashyapnadendla@arizona.edu)
- Tanya Evita George (University of Arizona, tanyageorge@arizona.edu)
- Zenith Mesa (University of Arizona, zenithmesa@arizona.edu)
- Eshaan Mathakari (University of Arizona, eshaanmathakari@arizona.edu)