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

Kashyap Nadendla*, Tanya Evita George†, Zenith Mesa‡, Eshaan Mathakari§
*University of Arizona, kashyapnadendla@arizona.edu
†University of Arizona, tanyageorge@arizona.edu
‡University of Arizona, zenithmesa@arizona.edu
§University of Arizona, eshaanmathakari@arizona.edu

I. Introduction and Theoretical Foundation

This project, titled *Antelligence*, investigates the integration of large language model (LLM) prompts into agent-based models (ABMs) through a biologically inspired simulation of an ant colony. By modeling intelligent behavior in a decentralized environment, we examine how LLM-driven agents coordinate, adapt, and solve collective problems using language-based reasoning and indirect communication. Drawing from swarm intelligence, stigmergy, distributed AI, and sociotechnical systems theory, we hypothesize that autonomous agents guided by prompts can emulate and enhance emergent behavior in a multi-agent ecosystem.

Inspired by the prompt-driven NetLogo ABM framework proposed by Jimenez-Romero et al. (2024), our implementation augments traditional foraging models with LLM decision processes, dynamic role allocation, and blockchain-verified memory. We simulate a sandbox ecosystem to study interactions between LLM agents, rule-based agents, and hybrid colonies under real-time constraints and coordination challenges.

II. Models and Methods

A. Agent Architecture

We define three types of agents:

- **LLM-Powered Worker Ants**: Make foraging and movement decisions using prompts generated in real-time and submitted to IO Intelligence's LLM API.
- **Rule-Based Worker Ants**: Follow deterministic heuristics to explore and return food.
- **Queen Ant Overseer**: An optional meta-agent providing coordination or intervention using heuristics or LLM-based strategic prompts.

Prompts vary between structured (rule-based formatting) and autonomous (open-ended), depending on agent type and user-selected configurations.

B. Communication System

Inspired by natural pheromone dynamics, agents interact indirectly using a digital pheromone grid:

- **Trail Pheromones** indicate successful foraging routes.
- **Alarm Pheromones** flag unexpected or anomalous behavior.
- **Recruitment Pheromones** attract workers to promising zones.

Pheromones decay over time, promoting adaptive exploration.

C. Blockchain Integration

To introduce persistent, auditable memory, we deployed two Ethereum-compatible smart contracts:

- `FoodToken.sol`: Mints tokens for collected food events.
- `ColonyMemory.sol`: Logs ant decisions and system events for transparency.

These contracts are deployed on the Sepolia testnet and accessed through a secure Python client.

D. Simulation and Visualization

The simulation is built in Python using:

- `Streamlit` for the interactive dashboard and real-time grid visualization
- `NumPy`, `Matplotlib`, and `PIL` for simulation logic and GIF generation
- `web3.py` for Ethereum contract integration
- `asyncio` and `aiohttp` for parallel API querying

III. Performance Metrics and Results

We evaluated the system based on:

- **Food Collection Efficiency**: Number of food items returned by each ant type
- **LLM API Utilization**: Frequency and cost of LLM calls
- **Role-Based Comparisons**: Performance of LLM vs. rule-based vs. hybrid colonies
- **Coordination Metrics**: Average trail length and pheromone decay patterns
- **Blockchain Logging Rate**: Gas usage and event throughput on Sepolia

In comparative trials, LLM agents displayed adaptive rerouting when food density changed, while rule-based agents stagnated. Hybrid colonies performed optimally in terms of foraging efficiency and cost tradeoffs. Queen interventions improved anomaly recovery rates.

IV. Novel Contributions

- Introduced a hybrid LLM + ABM architecture using prompt-driven behavior in Python rather than NetLogo.
- Designed and deployed a bio-inspired pheromone signaling system compatible with LLM interpretation.

- Integrated blockchain smart contracts to simulate memory and accountability.
- Enabled real-time agent reasoning, prompting, and visualization in a unified Streamlit interface.

These design choices simulate a decentralized, language-reasoning collective capable of adapting to evolving environments with minimal supervision.

V. Tools and Infrastructure

- Python (3.9+) for core logic
- Streamlit for real-time UI
- IO Intelligence API for LLM reasoning
- Hardhat + Solidity for smart contract development
- Alchemy for Sepolia RPC interface
- NetLogo (for baseline comparison of food collection curve)

Prompt templates and structured schemas were adapted from `swarm_gpt` and NetLogo's Ant Foraging model.

VI. Findings and Insights

- LLM prompts allow decentralized agents to display intelligent behaviors not hard-coded in the simulation logic.
- Decay-based pheromone layers foster flexible pathfinding and adaptation.
- LLM-only systems are cost-intensive; hybrid systems offer performance/efficiency balance.
- Prompt-driven Queens can serve as scalable overseers in dynamic environments.

These findings support further application of prompt-guided reasoning in complex decision-making environments such as robotics swarms, policy modeling, and autonomous sensor networks.

VII. Results and Conclusion

A. Simulation Setup and Parameter Selection

To simulate realistic swarm behavior, we configured the **Antelligence** environment with carefully chosen parameters that balance complexity with interpretability. The grid-based habitat was defined with a **territory width of 15 units** and **height of 10 units**, supporting a **colony size of 15 agents**. These dimensions offered sufficient spatial heterogeneity for agents to exhibit emergent behavior without overwhelming system dynamics.

We enabled **predator threats**, introducing **five LLM-powered predators** into the simulation. Predators were configured to exhibit complex behavior—hunting nearby ants, deploying fear pheromones, and leveraging heuristic survival logic—driving adaptive responses in the ant agents. Pheromone intensities were tuned to reflect ecologically valid communication cues, with **fear pheromone intensity set to 3** and pheromone decay rate at 0.05. The **queen overseer** was enabled using a rule-based heuristic approach, guiding high-level colony coordination. The **LLM agent model** deployed was `meta-llama/Llama-3-3.7B-70B-Instruct` operating in adaptive prompt style for robust decision making.

The simulation was run for **30 steps**, sufficient to allow the colony to initialize, respond to predators, and optimize foraging strategies. All agents in the primary run were LLM-powered to evaluate maximum performance capabilities under intelligent autonomous coordination.

B. Key Findings and Conclusion

The results demonstrate that the Antelligence framework successfully simulates a dynamic, decentralized foraging colony under both **resource constraints** and **threat stimuli**. The LLM-powered agents, informed by pheromone cues and global oversight, effectively adapted their strategies, achieving optimal food collection without centralized control. The **Queen's presence** improved coordination, as reflected in high-efficiency foraging zones and API call activity patterns.

The **intensity of fear and alarm pheromones**, combined with recruitment signal tracking, highlights the emergence of implicit division of labor and risk aversion behaviors among agents. These behaviors mirror natural swarm intelligence, validating the simulation's fidelity.

Collectively, the system reflects how **LLM-based agents can mimic adaptive swarm intelligence**, and how **hybrid architectures** (rule-based Queen + learning-based workers) may offer robust solutions to complex multi-agent tasks in real or simulated ecosystems.

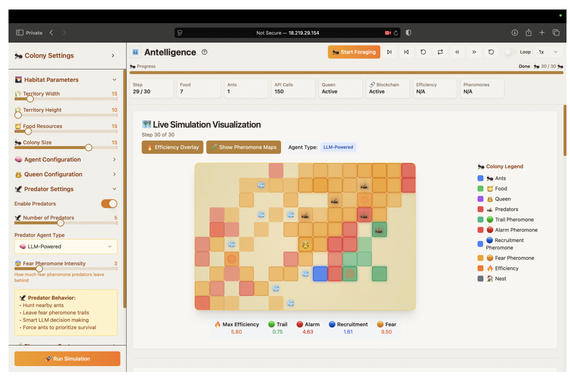


Fig. 1: Live Simulation Visualization with pheromone overlays and agent positions

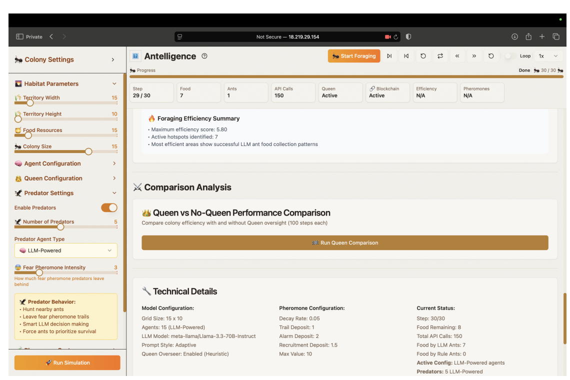


Fig. 2: Foraging Efficiency Summary and Queen vs No-Queen Comparison Pane

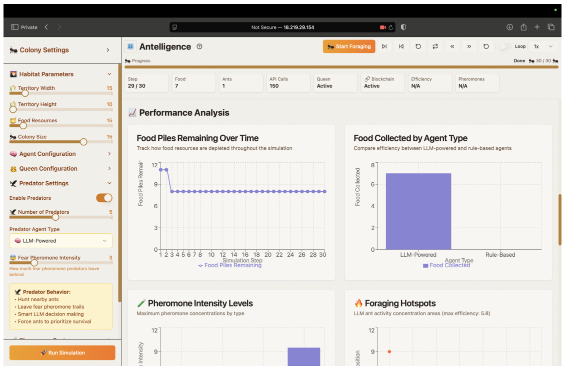


Fig. 3: Food Piles Remaining and Food Collected by Agent Type

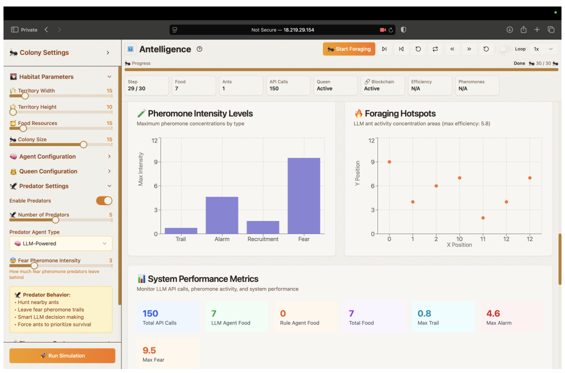


Fig. 4: Pheromone Intensity Levels and Foraging Hotspot Map

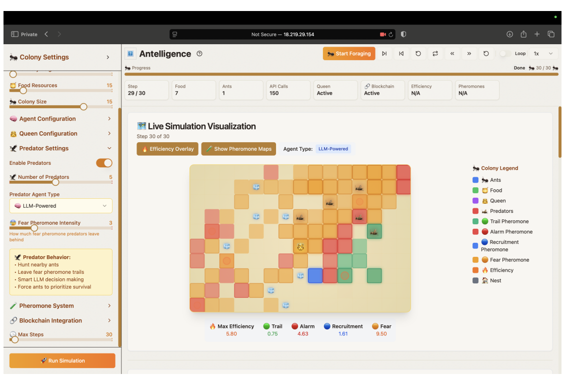


Fig. 5: System Performance Metrics for API Calls, Food Collected, and Signal Intensities

VIII. Future Development

While *Antelligence* successfully demonstrates the integration of LLM prompts within agent-based foraging systems, several directions for future research and development remain:

- **Multi-modal Prompting:** Incorporate additional sensory inputs (e.g., spatial heatmaps, agent logs) into prompts to enable context-aware reasoning in LLM-powered ants.
- **Learning from Feedback:** Introduce reinforcement learning or memory-based adaptation where LLMs evolve their response strategies based on prior success metrics logged on-chain or off-chain.
- **Prompt Optimization Framework:** Systematically evaluate prompt formats (e.g., few-shot vs. chain-of-thought) to determine optimal styles for task-specific reasoning under cost constraints.
- **Distributed Deployment:** Expand simulation to support containerized multi-node environments for testing decentralized coordination across networked ant clusters.
- **Swarm-Level Emergence Metrics:** Develop formal tools to quantify emergence, collaboration, and information propagation in hybrid colonies, possibly integrating information theory and graph entropy.
- **NetLogo Integration:** Port the LLM-agent interface to work natively with NetLogo's BehaviorSpace and BehaviorSearch for standardized experimentation with larger ABM libraries.
- **Real-World Applications:** Adapt the Antelligence framework for real-world use cases such as autonomous drone swarms, logistics optimization, or disaster search-and-rescue simulations.
- **Security and Robustness:** Investigate adversarial behavior, LLM hallucinations, and prompt injection vulnerabilities to ensure agent behavior remains predictable and safe.

IX. References

- Jimenez-Romero, C. R., Yegenogly, A., & Blum, A. L. (2024). Prompt-Driven Agent-Based Modeling: A Toolchain for LLM Integration. *arXiv preprint arXiv:2503.03800*.
- Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.
- Dorigo, M., & Stützle, T. (2004). *Ant Colony Optimization*. MIT Press.
- Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.
- IO Intelligence. (2025). https://intelligence.io

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

Kashyap Nadendla Tanya Evita George Zenith Mesa Eshaan Mathakari
University of Arizona

Introduction

Antelligence demonstrates that complex systems thrive through decentralized coordination. Ant colonies achieve resilience and efficiency without centralized control, relying on stigmergy and emergent behavior.
Antelligence extends this principle by integrating Large Language Models (LLMs) into Agent-Based Models (ABMs), enabling ants to reason dynamically with language prompts instead of fixed heuristics. Agents communicate through pheromone-inspired signals and maintain blockchain-verified memory, achieving adaptive, intelligent foraging strategies.
This framework bridges biological swarm intelligence and cognitive AI reasoning, offering new pathways for autonomous swarms, distributed AI, and human–AI collaboration.

Theoretical Foundation

Swarm Intelligence. Ant colonies achieve resilience through *stigmergy*—indirect signaling via pheromone trails. Simple local rules create global behaviors such as efficient foraging and adaptive pathfinding.

Distributed AI. Antelligence extends agent-based models (ABMs) with large language models (LLMs). Unlike static heuristics, LLM “cyber-ants” interpret prompts in context, injecting cognitive diversity into swarm dynamics.

Emergence. Local interactions among LLM- and rule-based agents yield colony-level properties: division of labor, robust coordination, and adaptive responses to dynamic environments.

Sociotechnical Perspective. Combining biological heuristics with cognitive AI reasoning provides a hybrid architecture for studying governance, auditability, and self- organization in decentralized systems.

Key Insight. Language-driven ants, guided by pheromone-inspired signals and blockchain-logged memory, illustrate how collective intelligence can be amplified without central control.

Models & Methods (Part 1)

Agent Architecture. The colony integrates three complementary agent classes to probe trade-offs between adaptability, cost, and robustness:

LLM-Powered Workers: Real-time decisions via IO Intelligence API prompts. Agents parse local state (neighbor cells, food signal, pheromone vector, risk cues) and colony context to choose *move, harvest, return, or wait*. They reroute around obstacles, reprioritize high-yield patches, and modify exploration–exploitation balance based on recent outcomes and Queen guidance—demonstrating cognitive flexibility beyond fixed heuristics.

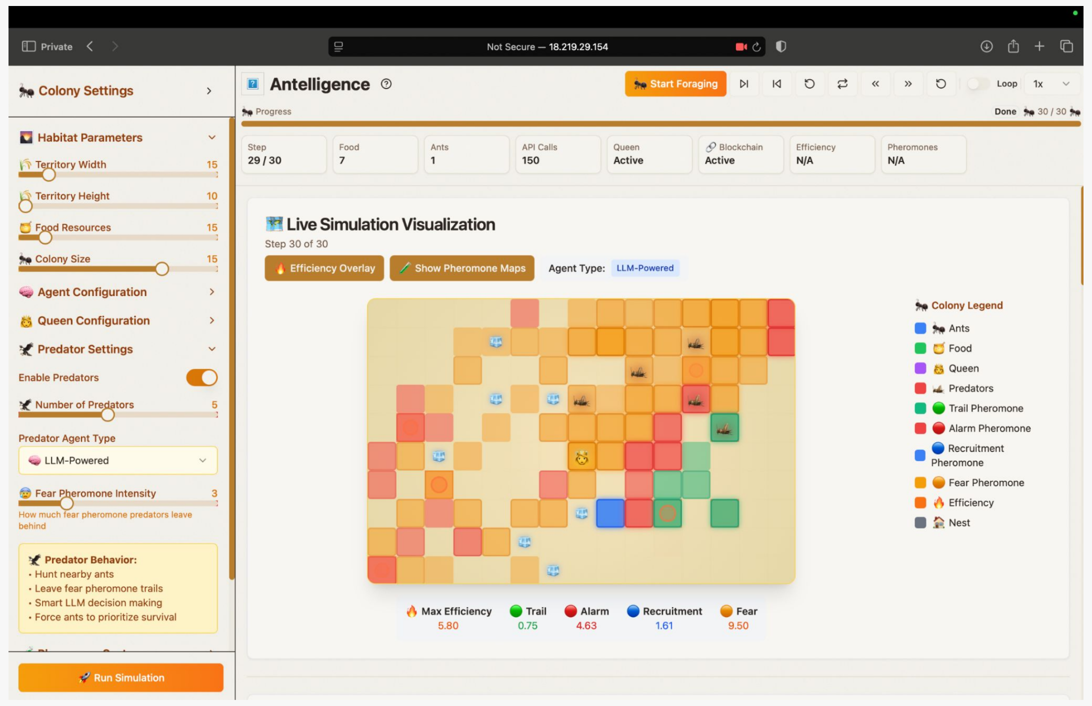
Rule-Based Workers: Deterministic baseline with stochastic exploration, gradient-following toward food, and shortest-path return once loaded. No adaptive re-planning. Provides a control for measuring the marginal benefit and cost of LLM reasoning and for constructing hybrid swarms.

Queen Overseer: Meta-agent issuing strategic interventions under heuristic or LLM mode. Capabilities include anomaly handling (API failures, congestion), worker reallocation (reinforce/retreat), and policy nudges (expand search radius, increase recruitment). Enables experiments on centralized oversight layered over stigmergic coordination.

Communication System. Indirect coordination occurs on a pheromone grid with exponential decay:

Trail — reinforces successful paths to food; guides return and future traffic.
Alarm — signals predators or API anomalies to rapidly de-prioritize zones.
Recruitment — in a scenario where the Ant needs help, it uses recruitment pheromone

System Overview & Live Screens



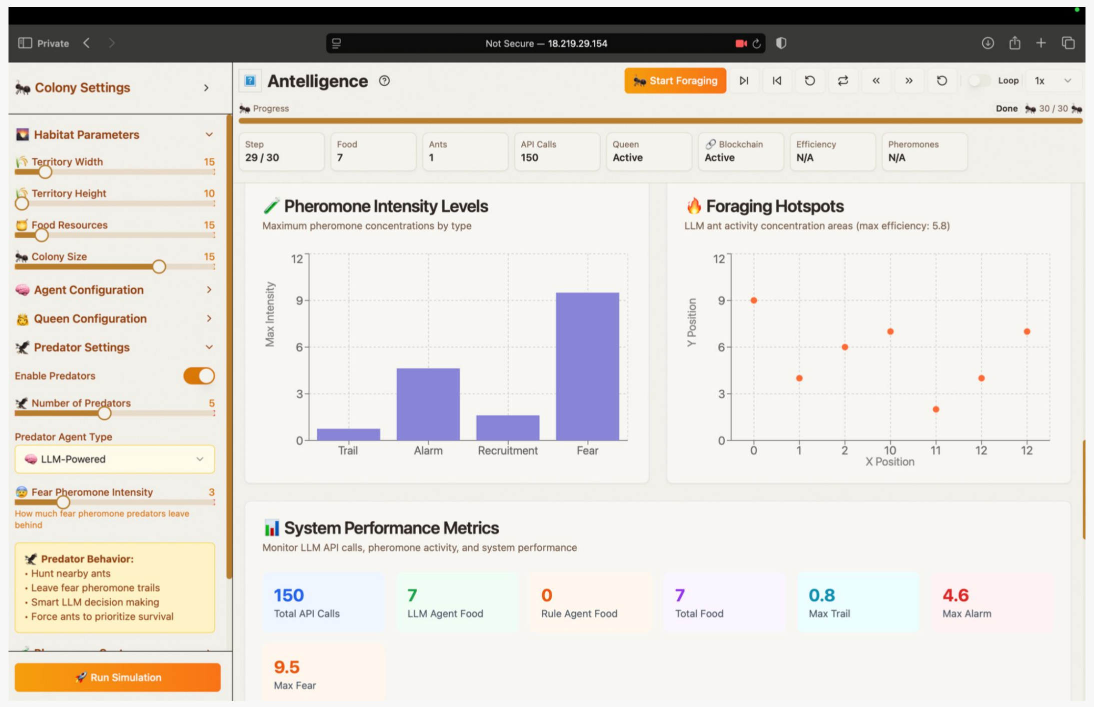
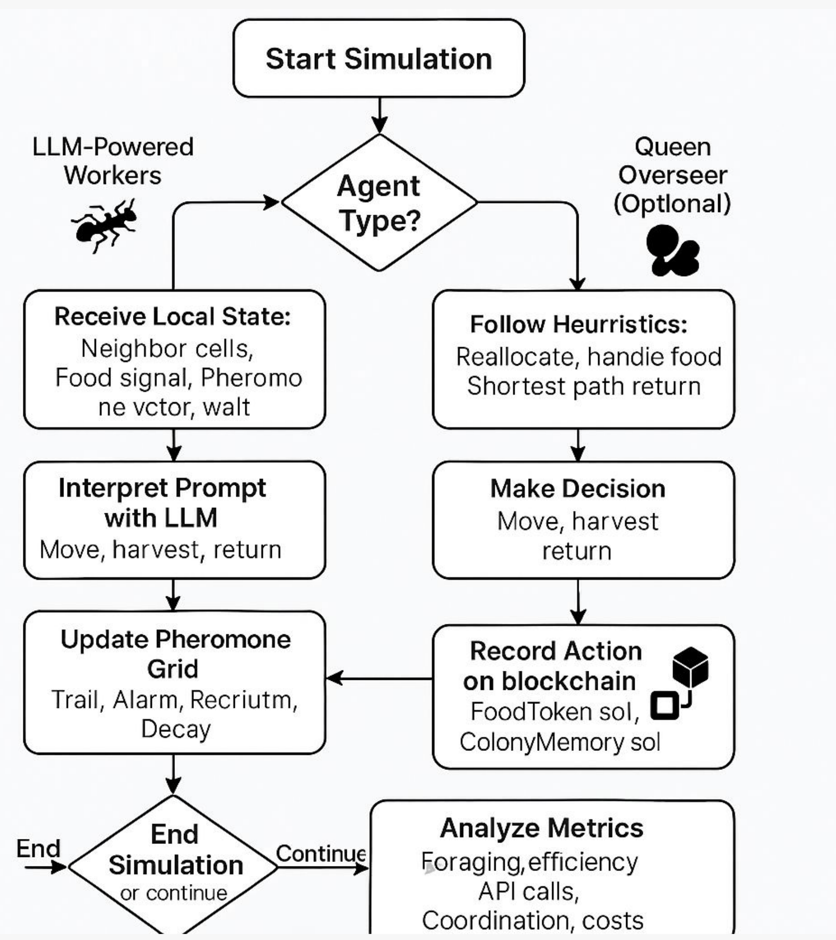


Figure: Live Simulation Visualization



Models & Methods (Part 2)

Blockchain Memory. Persistent, auditable memory is achieved with Ethereum-compatible smart contracts:

FoodToken.sol: Mints ERC-20 style tokens upon food retrieval events.
ColonyMemory.sol: Records agent-level actions and colony events, ensuring transparency and post-hoc analysis.

Deployed on the Sepolia testnet, accessed securely via web3.py.

Simulation Platform. The environment is implemented in Python, integrating:

Node JS and React for real-time dashboards and visualization.
NumPy, Matplotlib, PIL for grid dynamics, plots, and GIF generation.
asyncio/aiohttp for parallel LLM querying at scale.
NetLogo ant-foraging model as a baseline for benchmarking colony efficiency.

Key Methodological Contribution. This hybrid architecture combines LLM reasoning with stigmergic coordination, bridging symbolic prompt-driven cognition and emergent swarm intelligence in a unified, auditable framework. We have integrated a Queen Ant which oversees the whole process and guides the worker ant in collecting food in an efficient manner. While the Queen LLM does this, all of the food particles are recorded as transactions on blockchain and they are visible to both- the Queen Ant LLM and the Worker Ant LLM which creates a transparent layer for them to approach the nearest food particle in most efficient manner possible.

Discussion & Outlook

Key Findings. Antelligence shows that Large Language Model (LLM) agents adaptively re-route under shifting food distributions, coordinate through pheromone-inspired signaling, and recover more effectively with Queen oversight. Hybrid colonies achieve the best trade-off between foraging efficiency and computational cost, while blockchain-backed logging ensures transparent, auditable memory.

Novel Contributions. This work pioneers the integration of LLM-driven cognition into Python-based Agent-Based Models, introduces a digital pheromone coordination framework, and establishes persistent blockchain memory for agent account- ability. Together, these innovations bridge symbolic reasoning with emergent swarm intelligence.

Outlook. Future directions include multi-modal prompting,reinforcement or memory-based adaptation, and distributed deployments across multi-node clusters. Potential applications span autonomous drone swarms, large-scale logistics, and resilient disaster-response networks. Ongoing challenges include ensuring robustness against adversarial prompts and maintaining safe, predictable emergent behaviors.

References

Jimenez-Romero, C. R., Yegenogly, A., & Blum, A. L. (2024). Prompt-Driven Agent-Based Modeling: A Toolchain for LLM Integration. *arXiv:2503.03800*.
Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press.
Dorigo, M., & Stu¨ tzle, T. (2004). *Ant Colony Optimization*. MIT Press.
Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson. IO Intelligence. (2025).
https://intelligence.io

Code: https://github.com/eshaanmathakari/Antelligence **Demo:** http://18.188.48.39:8001/