# Emergent Consensus in Local-First Autonomous Agent Swarms

**Nacho Martínez (jasperan)**

***DevRel***
***Oracle***
*ignacio.martinez@oracle.com*

**Abstract**

As Large Language Models (LLMs) evolve from simple chatbots to autonomous agents, the potential for simulating complex social dynamics increases exponentially. In this paper, I present EmotionSim, a local-first framework for running high-fidelity multi-agent simulations. Unlike cloud-based solutions that face latency and privacy hurdles, EmotionSim leverages quantized local models (specifically the Gemma 3 family) to execute rich, discrete-event simulations of disaster scenarios. I demonstrate the system's capabilities through an "Airplane Crash Investigation" scenario, where 10 diverse autonomous agents—ranging from a retired pilot to an investigative journalist—must autonomously perceive their environment, share information, and reach a consensus on the cause of the accident. My findings show that even small, locally-hosted models can exhibit sophisticated emergent cooperation and information propagation patterns, supported by a system throughput of over 200 tokens/second on consumer hardware.

**Keywords:** Multi-Agent Systems Autonomous Agents Emergent Behavior Local AI Disaster Simulation

# 1 Introduction

The field of Artificial Intelligence is undergoing a paradigm shift from static prompts to dynamic, agentic interactions. Researchers are increasingly interested in how swarms of AI agents can model complex human systems, from economic markets to social crises [1]. However, conducting this research often requires massive compute resources or reliance on expensive, opaque proprietary APIs.

I developed **EmotionSim** to democratize access to high-fidelity social simulation. By optimizing for local execution using the Ollama runtime and efficient discrete-event scheduling, EmotionSim allows researchers to "run a disaster in their terminal," observing the micro-dynamics of human cooperation and panic without data leaving their machine.

# 2 System Architecture

EmotionSim is built on a decoupled, event-driven architecture designed to balance narrative richness with simulation determinism.

## 2.1 Discrete Event Engine

At the core is a deterministic simulation engine. Unlike real-time game loops, the engine advances in discrete "ticks," allowing agents to deliberate for as long as necessary without desynchronizing the world state.

```
async def _execute_step(self):
    # 1. Process pending world events (e.g., flood rising)
    self._process_event_queue()

    # 2. Agent Deliberation Cycle
    actions = []
```

```python
    for agent in self.agents.values():
        if agent.is_active:
            # Agents make decisions based on local perception
            action = await agent.tick(self.state)
            actions.append(action)

    # 3. Resolve Actions and Update World State
    self._resolve_conflicts_and_apply(actions)
```

Listing 1: Deterministic Simulation Step

### 2.2 Rich Persona System

Agents are not merely prompt templates; they are stateful entities with defined psychological profiles. Each agent is initialized with Big Five personality traits, a specific skillset, and a hidden "backstory" that influences their perception.

```python
class Persona(BaseModel):
    name: str
    occupation: str
    # Big Five Traits (0.0 - 1.0)
    openness: float
    conscientiousness: float
    extraversion: float
    agreeableness: float
    neuroticism: float

    # Dynamic State
    stress_level: int
    health: int
    inventory: List[str]
```

Listing 2: Persona Definition

This granularity ensures that a "Retired Pilot" agent notices engine debris that a "Civilian" agent would ignore, driving information asymmetry and the need for communication.

## 3 Experiment: Autonomous Investigation

To test the system's capacity for emergent logic, I designed the "Airplane Crash Investigation" scenario.

### 3.1 Scenario Setup

The simulation initializes 10 agents in a suburban neighborhood immediately following a light aircraft crash.

- **Agents**: Includes a Retired Pilot (witness), an ER Doctor (first responder), a Journalist (information broker), and several local residents.
- **Environment**: Locations include the Crash Site, a Hilltop vantage point, and a Community Center.
- **Information Distribution**: Key clues (e.g., "left engine smoke") are only visible to specific agents based on their location and backstory.

### 3.2 Emergent Behaviors

During the simulation, I observed distinct phases of autonomous behavior:

1. **Triage Phase (Steps 1-3)**: High-empathy agents (Doctor, Pastor) immediately prioritized the "Crash Site" to help survivors, disregarding investigation.

2. **Information Consensus (Steps 4-7)**: The Journalist agent actively moved between groups, aggregating witness reports. The Retired Pilot provided technical context to the Journalist's questions.
3. **Conclusion (Steps 8+)**: Without explicit instruction, the agents converged on a "mechanical failure" theory rather than "pilot error," synthesizing disjointed observations.

# 4 Benchmarks and Feasibility

While the primary focus is behavior, performance determines feasibility for large-scale research. I benchmarked the system using the 'gemma3:270m' model to test the limits of efficiency.

Table 1: Performance on Consumer Hardware (10 Agents)

| Metric | Result |
|---|---|
| Total Duration (1 Step) | 42.47 sec |
| Total Tokens Processed | 12,549 |
| Throughput | 200.71 tokens/sec |
| Avg Agent Latency | 4.72 sec |

The system achieves over **200 tokens/second** throughput. This high efficiency implies that simulating complex social interactions is becoming accessible to individual researchers without the need for H100 clusters.

# 5 Conclusion

EmotionSim validates that local-first AI models have reached a threshold of capability sufficient for complex social simulation. By embedding these models in a robust discrete-event framework, I enable a new class of privacy-preserving, reproducible agentic research. Future work will focus on scaling to 100+ agents using hierarchical swarms.

# Code Availability

The code is available at `https://github.com/jasperan/emotion-engine`.

# References


[1] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. *arXiv preprint arXiv:2304.03442*, 2023.

[2] Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, et al. The rise and potential of large language model based agents: A survey. *arXiv preprint arXiv:2309.07864*, 2023.