Emergent Social Collaboration in Multi-Agent LLM Systems

A Mars Colony Simulation Study

Yevheniy Chuba — YoreAI / ARESA

2026-04-01

Abstract

We present a two-layer architecture for studying emergent collaborative behavior in large language model (LLM) multi-agent systems. The lower layer is a deterministic rule substrate governing agent needs, pairwise bond dynamics, action selection, and a hard cost ceiling. The upper layer is an LLM-powered dialogue and decision module (GPT-4o-mini / Claude Haiku) that produces natural speech but does not drive the emergent social or task statistics. Reproducible claims in this paper come from the lower layer, which ships as `experiments/run.py` and runs in ~3 seconds with stdlib Python. Across 10 seeds × 10,000 ticks (30 simulated days) we observe a mean of 3.5 strong friendships (bond > 50), 14.6 working relationships ($15 <$ bond ≤ 50), and 3.6 tensions (bond < -15) per session, with 100% of construction sites completing and a work-dominated action distribution (49.2% working). The full LLM-enabled system runs in the browser at `yev/apps/home/app/future/gaming/`; the Python rig is the stable, reproducible substrate underneath.

Table of contents

1 Introduction

Motivation

Traditional multi-agent systems rely on hardcoded rules and predefined behaviors. Recent work (Park et al., 2023; Wang et al., 2024) shows that LLM-powered agents can develop emergent collaborative patterns when given individual personalities, contextual awareness, and freedom to make autonomous decisions. The open question is not whether such patterns emerge, but which of them are **artifacts of the specific LLM** and which are **properties of the agent architecture**. This matters for reproducibility: if every number in a paper depends on which model version was called, results are unverifiable.

Contributions

1. A two-layer architecture that separates the **deterministic rule substrate** (decay, bonds, actions, cost envelope) from the **LLM dialogue layer**. Papers can cite the first with full reproducibility; the second is illustrative.
2. A 330-line Python simulator (`experiments/run.py`) that reproduces every cited social-dynamics and task-completion number with stdlib-only dependencies in under three seconds.
3. A hard cost envelope proven analytically: at 100 dialogue events per session and a GPT-4o-mini average of 7×10 USD per call, a session cannot exceed $0.007. Measured usage tracks the ceiling exactly by construction.

Research Questions

1. Under a shared rule substrate, does a random initial role mix produce reproducible strong-friendship and tension counts?
2. Does self-organized construction emerge without explicit task assignment — and if so, how many unique workers does a single site typically attract?
3. Can the system hold all agents' needs above breach thresholds (energy/social/purpose 30) for the entire session?
4. What does the work-vs-social action distribution look like when agents choose actions under a need-biased role-weighted policy?

2 System Architecture

2.1 Two-layer design

LLM layer (browser only)
- Dialogue generation (GPT-4o-mini / Claude Haiku)
- Memory summarization
- Illustrative, not cited

Rule substrate (this paper)
- Needs decay: energy / social / purpose
- Action queue: working / socializing / resting / building
- Pairwise bond dynamics with role affinity
- Construction progress
- Hard API-call budget

Each of the 10 colonists is an autonomous agent with:

- **Role + personality** — one of `commander | scientist | builder | engineer | miner | medic`; a trait list and backstory used by the LLM layer but not by the rule substrate.
- **Needs** — `energy`, `social`, `purpose`, each in [0, 100]. Decay per tick is (0.020, 0.015, 0.010) by default.
- **Bonds** — symmetric pair score. Bond gain is scaled by a per-pair **affinity** multiplier derived from role compatibility plus random jitter. Affinities above a pivot of 0.9 gain bond while socializing/co-working; affinities below it actively lose bond. Passive decay drifts every bond back toward 0.
- **Action** — one of `working | socializing | resting | building | walking`, resolved by a need-biased role-weighted selector.

2.2 Action selector

The selector prefers the action that recovers the agent's lowest standing need 75% of the time; otherwise it samples from a hand-tuned distribution weighted toward work (45%) and social (18%). Builders (and 40% of non-builders) prefer unbuilt construction sites when any are available.

2.3 Hard cost envelope

The browser system enforces 100 API calls or $0.50 per session, whichever comes first. The rig mirrors this: dialogue events are rate-limited by `DIALOGUE_CAP_PER_SESSION = 100`; the per-call cost is fixed at $7×10 (GPT-4o-mini average), so the ceiling is exactly **$0.007** regardless of prompt content.

Table 1: Relationship categories per session (mean ± std, 10 seeds × 10,000 ticks).

Category	Value per session
Mean pair bond	12.32 ± 1.79
Max pair bond	75.54 ± 16.94
Min pair bond	-50.66 ± 22.90
Strong friendships (>50)	3.50 ± 1.43
Working relationships	14.60 ± 4.25
Neutral relationships	23.30 ± 4.35
Tensions (<-15)	3.60 ± 1.17

Table 2: Construction outcomes per session.

Metric	Value
Sites completed	3.00 / 3
Mean progress (%)	100.00
Unique workers / site	7.30 ± 0.58

3 Experimental setup

Platform. Apple M2 Pro, 32 GB RAM, macOS 15, Python 3.13, stdlib only (no NumPy, no PyTorch, no network). Full run completes in approximately 3 seconds.

Configuration. `N_AGENTS = 10`, `N_TICKS = 10,000` per session, `N_SEEDS = 10`. One tick corresponds to roughly 260 simulated seconds of colony time, so a full session covers on the order of 30 simulated days. Hardware and knobs are fixed in `experiments/run.py`; changing them changes the numbers here, and the rig's tolerance guard prints a warning if dynamics collapse.

What counts as what.

- **Strong friendship**: pair bond > 50
- **Working relationship**: $15 <$ pair bond 50
- **Neutral**: -15 bond 15
- **Tension**: bond < -15

4 Results

4.1 Social dynamics

Role-affinity structure produces a heavy-tailed bond distribution: a minority of pairs cross the strong-friendship threshold, most land in working-relationship territory, and low-affinity pairs drift into tension territory despite starting neutral. Across ten seeds, tensions are **not an artifact of one unlucky session** — every seed produces at least one.

Table 3: Needs stability and action distribution.

Metric	Value
Stable fraction (all agents 30 on all needs)	100.0%
Need-breach fraction (any agent < 20 on any need)	0.0%
Action - working	49.2%
Action - socializing	20.0%
Action - resting	16.7%
Action - walking	10.7%
Action - building	3.5%

Table 4: API cost envelope per session.

Metric	Value
Dialogue events observed	100.0
Dialogue cap per session	100
Cost per call (GPT-4o-mini avg)	\$0.00007
Observed cost / session	\$0.00700
Analytic ceiling / session	\$0.00700

4.2 Task coordination (self-organized construction)

All three construction targets complete in every session (mean 3.00 / 3). Sites attract an average of **7.3 unique workers** — meaningfully more than the three roles in the `builder` population (Rex-7, Finn Reyes, occasionally Commander Chen when construction is urgent), confirming that opportunistic helpers emerge from the 40% "even non-builders try to build" branch of the selector.

4.3 Needs stability and action distribution

Needs remain stable (all 30) for the entire session in every seed, and the action distribution is **work-dominated** (49.2% working) followed by socializing (20.0%). Construction appears scarce (3.5%) because the three sites complete early in the session; after that, the builder preference falls back to general work.

4.4 Cost envelope

The hard cap is respected by construction: the rig increments a dialogue counter under the same trigger conditions the browser uses (socializing or co-working), and returns early once the cap fires. The analytic ceiling is **$0.007 per session** at the configured dialogue cap and per-call cost.

5 Discussion

5.1 What is reproducible, what is illustrative

The deterministic rig reproduces everything in §5. It does *not* reproduce the actual text of agent conversations — that layer lives in the browser simulator and depends on the specific LLM version. Any dialogue excerpt in a talk or blog post should be labelled as a sample from a real LLM run, not a claim backed by this rig.

5.2 Comparison to related work

Park et al.'s *Generative Agents* (2023) demonstrated emergent social behavior in a 25-agent town with GPT-3.5. Our contribution is orthogonal: we isolate the *rule-governed* layer from the *LLM-generated* layer so that the former is cheap, reproducible, and falsifiable, while the latter can be swapped between models without changing cited statistics. CAMEL-AI and AutoGen focus on task decomposition under LLM orchestration; they do not report reproducible social-dynamics metrics.

5.3 Limitations

- **Rule substrate LLM behavior.** Agents' dialogue quality, memory formation, and nuanced decision-making all live in the LLM layer and are not measured here.
- **Small population.** 10 agents is enough to exhibit the categorical bond distribution and multi-worker self-organization, but not to study faction formation or minority dynamics.
- **Fixed affinity table.** Role compatibility is hand-chosen. A future pass should learn affinities from real dialogue traces of the browser simulator rather than specifying them a priori.

6 Future Work

1. **Scale.** Run 100+ agents to look for faction emergence and minority-role isolation.
2. **Learned affinities.** Fit the affinity matrix from observed bond trajectories in the browser simulator rather than hand-tuning.
3. **Crisis scenarios.** Inject resource shocks and measure how the rule substrate redistributes effort under pressure.
4. **Multiple colonies.** Two rigs sharing a resource stream; measure the inter-colony cooperation–competition gradient.

7 Conclusion

A two-layer architecture cleanly separates reproducible social and task dynamics from LLM-specific dialogue generation. Under the rule substrate alone — 10 agents, 10,000 ticks, 10 seeds — a reliable profile emerges: 3.5 ± 2.8 strong friendships, 14.6 ± 2.3 working relationships, 3.6 ± 1.6 tensions, and 3 / 3 construction sites completed with 7.3 average unique workers per site. Costs stay under

$0.007 per session by construction. Every number in this paper comes from `experiments/run.py` and can be regenerated in three seconds of wall-clock time.

Try it

- **Live system** — `http://localhost:3000/future/gaming` (open the `home` app with `npm run dev -w home`).
- **Reproduce the numbers** —

```
cd genass/publications/quarto/mars_colony_collaboration
uv run python experiments/run.py
```

8 References

Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., Bernstein, M. S. (2023). *Generative Agents: Interactive Simulacra of Human Behavior*. UIST '23.

Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., Anandkumar, A. (2024). *Voyager: An Open-Ended Embodied Agent with Large Language Models*. TMLR.