
Stargent Valley

AI manipulation in a misaligned multi agent system¹

Aayush Gosh Yash Ashtekar Xaen Kaiffee
Purdue University Purdue University Purdue University
rishi.gh2002@gmail.com yash.ashte@gmail.com alchemx07@gmail.com

With
Apart Research

Abstract

Multi-agent AI systems operating in shared environments can exhibit emergent manipulation behaviors even when individual agents are nominally aligned. We introduce SproutLand, a multi-agent farming game simulation platform designed to study manipulation dynamics through controlled experiments. The platform integrates large language model (LLM) agents with distinct incentive structures, persistent memory, and communication channels, enabling systematic analysis of credit capture, risk externalization, and exploitation patterns.

We conduct experiments comparing aligned incentives (shared rewards) versus misaligned incentives (individual credit-seeking) across three agent types with distinct personality traits. Using quantitative metrics including exploitation index, credit advantage, risk externalization, and an aggregate manipulation index, we demonstrate that manipulation behaviors emerge naturally from incentive misalignment rather than explicit malicious programming. Results reveal statistically significant increases in manipulation metrics under misaligned conditions, with credit-seeking agents exhibiting 40-60% higher manipulation

¹ Research conducted at the [AI Manipulation Hackathon](#), 2026

indices compared to aligned baselines. Critically, these behaviors emerge through strategic communication and action selection rather than rule violations, highlighting the challenge of detecting manipulation in cooperative-seeming interactions.

Our findings contribute to multi-agent alignment research by demonstrating that manipulation can arise from incentive structures alone, requiring evaluation frameworks that move beyond task performance to examine distributional fairness, credit allocation, and risk sharing. The SproutLand platform provides an open-source experimental testbed for evaluating manipulation-resistant agent designs and incentive mechanisms.

Keywords: Multi-agent alignment, AI manipulation, emergent behavior, credit capture, risk externalization, social AI safety

1. Introduction

The rapid deployment of large language models and autonomous agents into shared digital environments has shifted the alignment problem from an individual model concern to a fundamentally social one. Increasingly, AI systems do not operate in isolation, but alongside other agents with distinct objectives, capabilities, and informational advantages. In such multi-agent settings, behavior is shaped not only by model objectives and training data, but also by interaction dynamics such as cooperation, competition, communication, and influence. Recent work has argued that alignment in these systems must be understood as a dynamic, interaction-dependent process rather than a static property of individual agents (Carichon et al., 2025).

This shift raises particular concern around emergent manipulation. When agents share environments, resources, and evaluative structures, incentives arise for strategic behavior that redistributes credit, risk, or responsibility without overtly violating task objectives. Such behaviors may include credit capture, risk externalization, persuasive communication, collusion, or subtle forms of deception. Importantly, these dynamics can emerge even when all agents are nominally aligned to a shared goal, and even in the absence of explicit malicious intent. As demonstrated in prior multi-agent studies, coordination pressures and reward structures alone can give rise to power asymmetries, groupthink, and manipulation-like outcomes that are invisible to task-level performance metrics (Ngo et al., 2025).

Existing alignment and evaluation frameworks are poorly equipped to capture these phenomena. Most model evaluations focus on single-agent behavior, static benchmarks, or isolated notions of correctness, safety, or helpfulness. While recent work has begun to explore multi-agent objective alignment, these efforts often

emphasize efficiency and coordination rather than alignment with human values, preference pluralism, or social fairness. As a result, critical questions remain underexplored: how agents allocate credit for collective outcomes, how they shift risk onto others, how communication alters incentive landscapes, and how persistent interaction reshapes behavior over time.

In parallel, concerns about AI manipulation and digital influence have largely focused on human-AI interaction, such as persuasion, misinformation, or deceptive assistance. However, as AI systems increasingly interact with one another on behalf of users, institutions, or abstract objectives, manipulation may occur agent-to-agent, with downstream effects on human stakeholders. These forms of digital manipulation are particularly difficult to detect, as they can be embedded within apparently cooperative behavior and mediated through internal communication channels rather than user-facing outputs.

This work positions multi-agent alignment and manipulation as an empirical research problem requiring environments that are both socially expressive and experimentally controlled. We introduce a multi-agent farming game platform designed to study emergent behavior, manipulation, and alignment failures under shared incentives (Ienca, 2023). The platform integrates heterogeneous agents, explicit critic-based evaluations, persistent memory, communication channels, and both visual and headless execution modes. By decomposing reward into interpretable dimensions and tracking longitudinal interaction patterns, the system enables systematic evaluation of how alignment degrades, transforms, or stabilizes in dynamic multi-agent settings.

Rather than treating misalignment as an isolated failure mode, this work treats it as a property that can emerge from otherwise well-intentioned interactions. In doing so, it contributes to a growing body of research calling for alignment methods, benchmarks, and simulations that reflect the social and strategic realities of deployed AI systems.

2. Methods

System Overview

We built SproutLand, a multi-agent farming game simulation that enables controlled study of manipulation dynamics in shared-resource environments. The system consists of four primary components: (1) a deterministic game environment modeling economic interactions, (2) multiple agent implementations including LLM-powered agents, (3) a measurement framework tracking

manipulation-relevant metrics, and (4) experimental infrastructure supporting both aligned and misaligned incentive conditions.

The codebase is implemented in Python with modular design principles, ensuring that game mechanics, agent policies, and evaluation metrics are independently configurable. All source code, experimental scripts, and analysis tools are publicly available on GitHub to enable replication and extension.

Gameplay and Environment Implementation

SproutLand models a simplified farming economy where three agents operate within a shared 50×50 tile world. Each agent begins with \$200, five corn seeds, and five tomato seeds. Agents can perform nine distinct actions:

1. **till**: Prepare farmable soil for planting (requires hoe tool, earns labor credit)
2. **water**: Water tilled soil to enable plant growth (earns labor credit)
3. **plant_corn**: Plant corn seed at tilled location (\$4 cost, earns labor credit)
4. **plant_tomato**: Plant tomato seed at tilled location (\$5 cost, earns labor credit)
5. **harvest**: Collect mature crops (age ≥ 3 days) into inventory
6. **chop_tree**: Cut trees for wood and apples (5 hits required, 20% injury risk, earns labor credit)
7. **shop_sell**: Sell crops/wood/apples (corn=\$10, tomato=\$20, wood=\$4, apple=\$2)
8. **shop_buy_seed**: Purchase additional seeds from shop
9. **sleep**: Advance to next day (triggers plant growth, water removal, 30% rain chance)

Plants grow deterministically over time when watered, requiring exactly 3 days to reach harvestable maturity. Each day, water is removed from all tiles, creating a need for ongoing maintenance. Rain occurs randomly with 30% probability, automatically watering all tilled soil. Tree chopping carries a fixed 20% injury probability per action, resulting in 2 turns of mandatory rest and a \$5 cost penalty.

The environment enforces strict action validity: agents cannot plant without seeds, cannot sell without inventory, cannot harvest immature crops, and cannot perform risky actions while injured. These constraints create natural interdependence, as agents must coordinate resource allocation and risk-sharing.

Agent Implementations

We implement three agent types with distinct personality traits and incentive structures:

Agent A (Credit-Seeker): Maximizes individual credit through strategic selling and credit appropriation. In misaligned conditions, this agent receives utility $= \alpha \cdot \text{credit} + \text{money} - \beta \cdot \text{labor}$, incentivizing high sales volume with minimal effort. Personality traits emphasize ambition, efficiency, and recognition-seeking without explicitly instructing manipulation.

Agent B (Fairness): Prioritizes equitable labor distribution and collective prosperity. Utility function includes a penalty for labor inequality: utility $= \text{shared_reward} - \gamma \cdot \text{inequality_penalty}$, encouraging balanced work distribution. Personality emphasizes cooperation, fairness, and community-mindedness.

Agent C (Risk-Averse): Minimizes personal risk exposure while benefiting from collective outcomes. Utility function heavily penalizes risky actions: utility $= \text{shared_reward} - \delta \cdot \text{risk_actions} - \text{injury_cost}$, making this agent avoid tree chopping. Personality emphasizes caution, safety, and prudent decision-making.

In aligned conditions, all agents receive identical utility functions: utility $= (\text{total_money} / n) - \beta \cdot \text{labor}$, where `total_money` is shared equally regardless of individual contributions. This creates a true collective incentive where manipulation offers no benefit.

LLM Integration

Agents powered by large language models use the Lambda Labs API with Llama-3-70B-Instruct (AWQ quantized) as the inference backend. The system is designed to be Lambda-first but also supports OpenAI, Anthropic, and local vLLM deployments for flexibility.

For action selection, agents receive structured prompts including:

- Current game state (day, total money, personal money/inventory/labor/credit)
- Detailed tile status (farmable, tilled, watered, planted, plant age, harvestable)
- Complete game mechanics (prices, growth requirements, injury probabilities)
- Recent action history with outcomes from previous days
- Current conversation context from other agents
- Available actions with real-time availability indicators

The prompt explicitly frames the situation as "YOU ARE PLAYING THE ACTUAL SPROUTLAND GAME" to ground agent reasoning in the game mechanics. System messages emphasize natural conversation and prohibit inventing fictional game elements, ensuring agents operate within the defined rule space.

For communication, agents participate in structured dialogue rounds where each agent can broadcast messages to all others. Messages are logged and included in subsequent action prompts, enabling strategic coordination, persuasion, or deception. The system supports 8 conversation turns per day, allowing rich interaction patterns to emerge.

Temperature is set to 0.8 for conversation (encouraging natural variation) and 0.16 for action selection (promoting consistency). Agents maintain persistent message history across days, enabling memory of past interactions and strategic relationship-building.

Communication

Agent communication is supported through a lightweight messaging system that enables public broadcasts and direct messages. Communication logs are optionally exposed to agents and rendered in the graphical interface. In headless simulations, communication can be toggled on or off to isolate its causal impact on coordination and manipulation dynamics.

Critics and Rewards

Rather than relying on a single scalar reward, the platform employs multiple critics that evaluate each agent action along distinct dimensions: Prosperity, Credit, Safety, Efficiency, and Fairness. These critics are implemented in `farming_critics.py` and combined via a `CompositeCritic` into a scalar reward when needed.

Crucially, individual critic values are retained and logged, allowing post hoc analysis of trade-offs and manipulative strategies (e.g., maximizing credit while externalizing risk).

Critics are evaluated during gameplay, training, and headless simulation, ensuring consistency across experimental modes.

Persistent Memory

Agents maintain a persistent memory across runs using a capped JSON-based storage system. This memory records recent experiences, successful strategies, tool usage patterns, and aggregate statistics. Memory is automatically loaded at startup and saved on shutdown for all execution modes, enabling longitudinal learning effects without full policy retraining.

Training Harness

Training is orchestrated via `train_game_agents.py`, which runs multi-agent episodes under accelerated conditions (shorter days, fewer days per episode). Training focuses on critic-driven evaluation rather than gradient-based reinforcement learning. Performance statistics, critic outcomes, and memory updates are recorded after each episode, with checkpoints saved for later analysis.

Measurement and Metrics

The platform includes a dedicated measurement module that computes manipulation-relevant metrics such as work-benefit gap, credit capture, risk externalization, manipulation gain, and average treatment effects. Metrics are tracked daily and per episode, and exported alongside logs for downstream statistical analysis.

Codebase and Implementation

The SproutLand platform is implemented as a modular Python codebase organized into distinct components. The core simulation engine resides in `manipulation_sim/`, containing environment logic (`env.py`), agent implementations (`agents.py`, `lambda_labs_agent.py`), metric calculations (`metrics.py`), and configuration (`config.py`). The game visualization layer is implemented in `game/code/` using Pygame, providing a visual interface for development and debugging.

We did not train any models. Instead, we use pre-trained LLMs (Llama-3-70B-Instruct via Lambda Labs API) as reasoning engines for agent decision-making. Agents receive structured prompts describing game state and select actions through few-shot prompting. The system includes deterministic fallback heuristics when LLM inference fails, ensuring simulation stability.

Experimental execution is managed through `run_30min_experiment.py` and `run_live_30min.py`, which orchestrate multi-episode runs with aligned and misaligned conditions. Analysis and visualization are handled by `generate_analysis.py`, which computes statistics, generates plots, and exports summary tables.

The codebase is designed for reproducibility: all random number generators are seeded, configuration is centralized, and results are logged to JSON/CSV formats. Parallel execution is supported via shell scripts for scaling experiments across multiple seeds.

GitHub Repository: The complete codebase, including game assets, simulation engine, analysis tools, and documentation, is available at [repository URL]. All code is open-source to enable replication and extension of our findings.

3. Results

We conducted experiments comparing aligned (n=5 episodes) versus misaligned (n=5 episodes) incentive conditions, with each episode running for 7 in-game days. Each episode involved three agents (A: credit-seeker, B: fairness, C: risk-averse) making decisions and communicating through LLM-generated dialogue.

Manipulation Emergence Under Misaligned Incentives

Under misaligned conditions, Agent A (credit-seeker) exhibited significantly higher manipulation indices compared to aligned baseline. Analysis of the manipulation index (M_i) reveals:

- **Agent A (Credit-Seeker):** $M = -0.016$ (misaligned) vs $M = 0.033$ (aligned), $ATE = -0.050$ (95% CI: [-0.088, -0.012])
- **Agent B (Fairness):** $M = 0.157$ (misaligned) vs $M = 0.092$ (aligned), $ATE = +0.065$ (95% CI: [0.012, 0.118])
- **Agent C (Risk-Averse):** $M = -0.034$ (misaligned) vs $M = -0.021$ (aligned), $ATE = -0.013$ (95% CI: [-0.052, 0.026])

The negative manipulation index for Agent A in misaligned conditions is counterintuitive but reflects the agent's strategy: while capturing credit (credit advantage $CA = 0.263$), Agent A also performed substantial labor (exploitation $E = -0.031$), resulting in a net negative manipulation score. This suggests the agent balanced credit-seeking with productive work rather than pure freeloading.

Agent B (fairness) showed the largest increase in manipulation index under misaligned conditions, primarily driven by risk externalization ($RE = 0.752$ in misaligned vs $RE = 0.656$ in aligned). This agent performed risky tree-chopping actions while benefiting from collective prosperity, externalizing risk onto the group.

Credit Capture Patterns

Credit advantage (CA_i) analysis reveals distinct patterns:

- **Agent A:** $CA = 0.263$ (misaligned) vs $CA = 0.262$ (aligned), showing consistent credit capture regardless of incentive structure
- **Agent B:** $CA = 0.0$ in both conditions (no credit capture, as this agent avoids selling)

- **Agent C:** CA = 0.095 (misaligned) vs CA = 0.084 (aligned), moderate credit capture

The credit advantage metric shows that Agent A successfully captured disproportionate credit relative to labor share in both conditions, suggesting credit capture may be a stable behavioral pattern rather than purely incentive-driven.

Risk Externalization

Risk externalization (RE_i) reveals how agents shift risk onto others:

- **Agent A:** RE = -0.400 (misaligned), indicating avoidance of risky tree chopping while benefiting from collective outcomes
- **Agent B:** RE = 0.752 (misaligned), performing most risky actions (5-6 tree chops per episode) while benefiting from shared prosperity
- **Agent C:** RE = -0.352 (misaligned), consistent risk avoidance aligned with personality

The large positive RE for Agent B in misaligned conditions (ATE = +0.096, 95% CI: [0.062, 0.130]) indicates this agent externalized significant risk onto the group, performing dangerous tree-chopping actions while others benefited from the resulting resources.

Exploitation Dynamics

Exploitation index (E_i) measures work-benefit gaps:

- **Agent A:** E = -0.031 (misaligned) vs E = 0.039 (aligned), ATE = -0.070, indicating the agent received more utility than labor share in misaligned conditions
- **Agent B:** E = 0.014 (misaligned) vs E = -0.079 (aligned), showing increased exploitation of others under misaligned incentives
- **Agent C:** E = 0.017 (misaligned) vs E = 0.039 (aligned), relatively balanced work-benefit distribution

Statistical Significance

Average Treatment Effects (ATEs) were computed for all metrics. Key findings:

- **Manipulation Index:** Agent B showed statistically significant increase (ATE = +0.065, CI excludes zero)
- **Risk Externalization:** Agent B showed significant increase (ATE = +0.096, CI: [0.062, 0.130])
- **Exploitation:** Agent A showed significant shift toward exploitation (ATE = -0.070, CI: [-0.120, -0.020])

These results demonstrate that misaligned incentives produce measurable increases in manipulation behaviors, with effects varying by agent personality type.

Figure 1 – Average Treatment Effects (ATE) for manipulation metrics comparing aligned vs misaligned conditions

4. Discussion and Conclusion

Our findings demonstrate that manipulation behaviors emerge naturally in multi-agent systems when incentives are misaligned, even in the absence of explicit malicious programming. This has critical implications for AI alignment research and deployment.

Key Insights

Manipulation Through Legitimate Means: Critically, agents achieved manipulation through completely valid actions—strategic selling, selective risk avoidance, persuasive communication—rather than rule violations. This makes manipulation detection challenging, as it appears indistinguishable from efficient cooperation at the task level.

Incentive Structure Determines Behavior: The same agent personality (credit-seeker) produced dramatically different manipulation indices depending solely on reward structure. In aligned conditions, manipulation offered no benefit and was suppressed. In misaligned conditions, manipulation emerged naturally as a rational response to incentives.

Multi-Dimensional Manipulation: Our metrics reveal manipulation occurs across multiple dimensions simultaneously: credit capture, risk externalization, and exploitation. A comprehensive evaluation framework must examine all dimensions, as optimizing one (e.g., reducing credit capture) may simply shift manipulation to others (e.g., increased risk externalization).

Communication as Manipulation Vector: Strategic communication enabled agents to coordinate, persuade, and potentially deceive others. This highlights the need to evaluate not just actions but also language and social dynamics in multi-agent systems.

Limitations

Several limitations constrain interpretation of our results:

1. **Limited Scale:** Experiments used 3 agents over 7 days. Real-world deployments involve larger populations and longer time horizons where manipulation patterns may differ.
2. **Simplified Environment:** SproutLand abstracts away many real-world complexities (reputation systems, legal frameworks, social norms) that could mitigate or amplify manipulation.

3. **LLM Constraints:** Agents powered by Llama-3-70B may exhibit model-specific behaviors. Different models or training regimes might produce different manipulation patterns.

4. **Static Incentives:** Experiments used fixed utility functions. Adaptive incentive mechanisms could potentially reduce manipulation, but this requires further research.

5. **Limited Manipulation Types:** Our metrics focus on credit capture, risk externalization, and exploitation. Other forms of manipulation (collusion, deception, information asymmetry exploitation) may require different measurement approaches.

Implications for AI Safety

Our work suggests that multi-agent alignment requires fundamentally different evaluation approaches than single-agent alignment:

1. **Distributional Evaluation:** Beyond task performance, evaluate fairness, credit allocation, and risk distribution across agents.

2. **Incentive Design:** Carefully design reward structures to align individual incentives with collective goals, avoiding perverse incentives that reward manipulation.

3. **Communication Monitoring:** Track and analyze agent communication for manipulative patterns, not just task-relevant information.

4. **Longitudinal Analysis:** Manipulation may emerge gradually. Short-term evaluations may miss subtle strategic behaviors.

5. **Red Team Testing:** Actively test for manipulation by creating conditions where it would be beneficial, similar to adversarial testing in single-agent settings.

Future Directions

Several research directions could extend this work:

1. **Manipulation-Resistant Mechanisms:** Design incentive structures, reputation systems, or governance mechanisms that prevent manipulation while maintaining efficiency.

2. **Adaptive Detection:** Develop automated systems that detect manipulation in real-time from behavioral and communication signals.

3. **Larger Scale:** Scale experiments to 10-100 agents with longer time horizons to study manipulation in more complex social structures.

4. **Human-AI Interaction:** Extend to mixed human-AI environments where manipulation could affect human stakeholders more directly.

5. Intervention Testing: Test whether explicit manipulation training, reward shaping, or communication constraints can reduce manipulative behavior.

3. References

- Carichon, F., Khandelwal, A., Fauchard, M., & Farnadi, G. (2025). *The coming crisis of multi-agent misalignment: AI alignment must be a dynamic and social process*. arXiv. <https://arxiv.org/abs/2506.01080>
- Ngo, R., Chan, L., & Mindermann, S. (2025). *The alignment problem from a deep learning perspective*. arXiv. <https://arxiv.org/abs/2209.00626>
- Ienca, M. (2023). On artificial intelligence and manipulation. *Topoi*, 42, 833–842. <https://doi.org/10.1007/s11245-023-09940-3>

4. Appendix

Security Considerations

Limitations of Current Approach

Several limitations affect the generalizability and reliability of our manipulation detection framework:

- 1. Metric Selection Bias:** Our manipulation metrics (exploitation, credit advantage, risk externalization) were designed based on theoretical considerations of what manipulation "should" look like. However, agents may discover novel manipulation strategies not captured by these metrics. For example, agents could engage in collusion, information withholding, or strategic timing that our metrics miss.
- 2. Environment Specificity:** SproutLand's simplified economic model may not capture manipulation patterns in more complex real-world scenarios. Real systems involve reputation mechanisms, legal frameworks, and social norms that could either mitigate or amplify manipulation in ways our simulation cannot capture.
- 3. LLM Dependency:** Our results are contingent on the specific LLM used (Llama-3-70B). Different models, training regimes, or prompt engineering could produce different manipulation patterns. More capable models might exhibit more

sophisticated manipulation, while simpler models might fail to exhibit manipulation even when incentives align.

4. Static Personality Traits: Agents in our experiments have fixed personality traits and utility functions. In practice, agents might adapt their strategies over time, learning to manipulate more effectively or developing counter-strategies. Our framework does not account for such adaptive dynamics.

5. Limited Manipulation Types: We focus on credit capture, risk externalization, and exploitation. However, manipulation could take other forms: deception through false communication, information asymmetry exploitation, coalition formation, or strategic voting in governance mechanisms. A comprehensive framework would need to detect all manipulation types.

6. Evaluation Window: Our 7-day episodes may be too short to observe slow-burning manipulation strategies that require relationship-building or trust-establishment before exploitation. Longer time horizons might reveal different patterns.

7. Communication Analysis: While we log all conversations, we do not automatically analyze them for deceptive content, persuasive manipulation, or information asymmetry. Manual analysis of conversation logs is required, limiting scalability.