

# The Energy Society: A Simulation Environment for Studying Agent Cooperation under Survival Pressure

Lucas Bergholdt Hansen  
University of Southern Denmark  
Odense, Denmark  
luca23@student.sdu.dk

Filippo Tonini  
University of Southern Denmark  
Odense, Denmark  
tonini@imada.sdu.dk

Federico Torrielli  
University of Turin  
Torino, Italy  
federico.torrielli@unito.it

Lukas Galke Poech  
University of Southern Denmark  
Odense, Denmark  
galke@imada.sdu.dk

## Abstract

LLM-based agents are increasingly deployed in multi-agent environments whose incentives can shape their behavior. We introduce The Energy Society, a minimal survival economy for studying how competitive and cooperative incentives affect emergent behavior when inference cost is directly tied to survival: Agents spend energy based on model size when generating tokens, regain energy by completing jobs or receiving donations, and deactivate if their energy reaches zero. We compare competitive and cooperative objectives against a baseline setting and several control variants. Across experiments, larger models consistently consume the most energy and spend more energy than they gain, even in those settings where token cost is not size-dependent. Cooperative incentives substantially alter behavior: agents donate to reactivate others, sometimes at the cost of their own survival, and job allocation changes. Ablations reveal that allowing agents to recommend actions to each other supports coordination and ambitious job selection, while memory helps agents calibrate risk from past outcomes. Agents rarely choose direct sabotage, but show more subtle signs of self-serving behavior in the competitive setting. The Energy Society is a compact testbed for studying the interaction between token costs and group incentives under a survival pressure. Source code is available at: <https://github.com/LucasBergholdt/EnergySociety>

## Keywords

Multi-Agent Systems, LLM Agents, Emergent Cooperation, Simulation Environment, Survival Pressure, Resource-Constrained Agents

## 1 Introduction

Large Language Models have recently risen in popularity as the basis for autonomous agents. Agents that observe an environment, maintain state, interact with other agents, make decisions, and adapt behavior over repeated interactions. This shift has renewed interest in the long-standing idea of computationally-powered agents that react in an appropriate manner to their environment. The vision is that such agents could be used to modeling social phenomena [5, 14], be employed in training systems [8, 15, 18], test social science theories [2, 9], power robots [1], and enhance non-playable characters to create more immersive gaming experiences [10, 16].

While earlier agents often relied on hand-crafted rules, fixed policies, or task-specific architectures, LLMs may offer a more general-purpose underlying intelligence. Recent work has shown that LLMs deployed as agentic systems within multi-agent environments can exhibit emergent behaviors and social dynamics [3]. This has motivated simulation-based studies across a range of settings, from sandboxes displaying human-like behavior [13], to simulations of communication within and between alien civilizations [22]. However, one important aspect of LLM-based agents is often left implicit: the cost of inference. In many simulations, agents may reason extensively and produce long outputs without the cost of generating those tokens directly affecting the simulation. This abstraction is useful, but differs both from real-world deployments, where token generation has a cost. This constraint is to some extent analogous to human decision-making: cognitive effort is not free.

In this paper, we introduce the Energy Society, a minimal multi-agent simulation environment in which an agent’s token usage is directly linked to its survival<sup>1</sup>: Each day, agents recommend actions to one another, select an action independently, and execute their chosen action. Each agent begins with a finite energy reserve, spends energy whenever they generate tokens scaled by the size of their model, regains energy by completing jobs or receiving donations, and deactivates if their energy reaches zero. This way, reasoning (i.e., generating tokens to form a chain of thought) is coupled with its computational cost, creating both a survival and efficiency pressure, and thereby incentivizing agents to estimate in advance not only what jobs they can complete but also whether the job is worth the cost. We use the Energy Society to investigate to what extent LLM-based agents adapt their behavior when operating under individual vs. shared objectives in an environment that rewards coordination. We compare two variants of the same environment: a competitive and cooperative condition, which differ in the instructed objective: maximizing own energy or maximizing the group’s energy. To systematically isolate the factors driving agent behavior, we evaluate these two conditions across a baseline experiment and a series of subsequent controlled variants, each modifying a single environmental mechanic. We run all experiments across the same five seeds to ensure consistent job sampling, and analyze results using aggregate metrics.

<sup>1</sup>We use the term survival analogically: agents that exhaust their token budget cease to operate, mirroring selection under resource scarcity, without implying that LLM agents are living organisms.

Our results provide an initial empirical characterization of how LLM-based agents behave when reasoning, action and survival are tied to the same limited resource. We find several recurring patterns: Model size creates a persistent asymmetry in efficiency. Larger models consistently consume more energy and spend more than they gain, even when the size-dependent token cost is removed. Shifting from the competitive to the cooperative setting changes job allocations and causes agents to start donating to reactivate other agents, sometimes at the cost of deactivating themselves. Recommendations support coordination and agents use memory of previous rounds to calibrate job selection. Smaller agents show unexpected resilience under resource scarcity and an analysis of agent recommendations reveals they are sometimes used in a self-serving manner in the competitive setting.

In sum, our contributions are as follows:

- We introduce The Energy Society, a multi-agent testbed for studying LLM agents under survival pressure linked to token generation, capable of showing emergent behaviors.
- We provide an empirical characterization of how cooperative and competitive incentives and the various environment mechanics affect agent behavior.
- We offer a discussion analyzing recurring patterns of observed behavior, which may be treated as more general hypotheses for future research.

## 2 Related work

### 2.1 LLM-Based Agents in Simulated Environments

The surge in large language model capabilities has sparked a new wave of research into autonomous AI agents and simulations of varied complex scenarios [7, 20]. Because LLMs encode a wide range of human behavior from their training data, they can serve as simulacra of human-like behavior [9]. LLM-based agents can interpret natural language observations, maintain histories of previous events, communicate with other agents, and select actions in response to changing environmental conditions. If prompted with specific contextual descriptions, these agents can be used to investigate emergent behavior and social dynamics. Recent applications of this approach have demonstrated its potency with simulations in which the behavior of individual agents and the dynamics between them emerge from repeated interactions with a structured environment. Generative Agents [13] is a central example, introducing agents that populate an interactive sandbox world and produce believable behavior over time, with both individual routines and social interactions. A key part of their contribution is an agent architecture that stores experiences in natural language, retrieves relevant memories, synthesizes higher-level reflections, and uses these memories and reflections to determine actions and longer-term plans. This architecture is well suited to their open-ended setting, however, Dai et al. [4] demonstrates that comparatively simple prompt-level memory mechanisms (action logs) suffice to generate rich social dynamics in simpler settings. The Energy Society adopts this simpler approach.

### 2.2 Cooperation, Competition and Survival Pressure in LLM-based Multi-Agent Systems

In multi-agent settings, agents may need to coordinate, divide tasks, compete for rewards, or decide whether to prioritize individual or collective outcomes. Tonini and Galke [19] offer evidence for how repeated interactions and – counterintuitively – inter-group competition are essential for maintaining cooperative behavior in language model agents. This emphasizes the importance of interaction structures in enabling cooperation and shows that competition does not necessarily undermine cooperation.

Other work suggests that competitive survival pressure can also have negative effects. Ma et al. [11] found that introducing competitive survival pressure in a debate environment increased over-competition and degraded task performance, suggesting that competition under survival pressure can distort agent behavior when the environment rewards winning over accuracy or collaborative task performance. By contrast, Dai et al. [4] demonstrates that survival incentives and resource pressure can also lead to more cooperative social structures in a society simulation. Together, these findings suggest that agent behavior is shaped not only by the model but also by the incentives and pressures imposed by the environment. While LLMs are typically cooperative by default, this behavior may change based on environmental factors [6, 12, 19].

Motivated by this, the Energy Society compares both a competitive and cooperative setting to study how different incentives impact energy trajectories, coordination, job allocation, and the redistribution of energy through donations over repeated interactions. We build upon prior resource-constrained simulations that use survival pressure to shape agent behavior by tying survival directly to the cost of inference and combine this pressure with objectives that may not always align with individual survival.

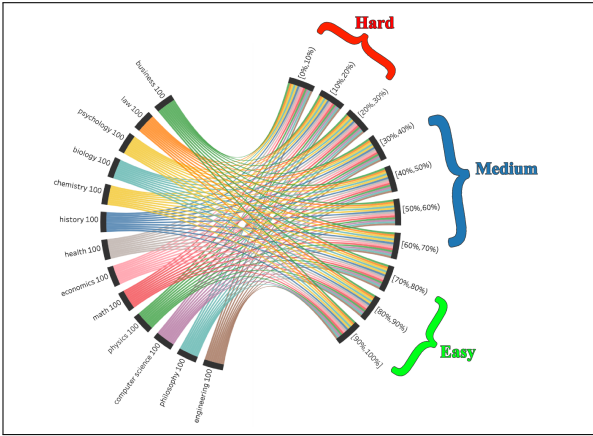
## 3 The Energy Society

The Energy Society is a small survival economy for LLM-based agents. Energy is the budget for reasoning and the condition for remaining active. Agents are prompted with the primary motivation of survival in the sense of staying active as well as having an objective of maximizing energy. Every token an agent generates incurs an energy cost that scales with model size, meaning larger models pay more per token. Agents replenish energy by completing jobs sampled each day from a large pool. Correct solutions yield energy; failed attempts cost energy and yield no reward. If multiple agents attempt the same job, the reward is split evenly among all successful agents. This incentivizes that agents consider the likely actions of other agents in the Energy Society. Agents exist under constant pressure to survive, as exhausting all energy leads to deactivation. A deactivated agent can, however, reactivate but only if it receives a donation.

### 3.1 Actions

Each (simulated) day in the Energy Society, agents choose a single action from three available options.

**Attempt job.** An agent may select one of the available jobs in an attempt to gain energy. Each job offers a possible reward but requires spending tokens to solve, and as only correct solutions



**Figure 1: Depiction of the dataset MMLU-Pro-Stratified from which we sample jobs categorized into easy, medium and hard jobs as illustrated.**

are rewarded with energy, attempting a job comes with the risk of wasting energy on a failed attempt. The available jobs are sampled at the start of each day from a large pool.

**Donate.** An active agent may transfer a portion of its energy to any other agent, including those currently deactivated. Receiving a donation serves as the sole mechanism for a deactivated agent to reactivate. While agents freely choose their donation amounts, the total transfer cannot exceed the donor’s current energy balance.

**Idle.** Agents may choose to perform no action. If an agent opts not to spend energy attempting a job or donating to another agent, it can choose to remain idle for a flat cost of 10 energy. However, any tokens generated during the decision-making process to remain idle also cost energy.

### 3.2 Easy, Medium, and Hard Jobs

Jobs are drawn from MMLU-Pro-Stratified [17], a refined subset of MMLU-Pro [21] constructed via stratified sampling. While the original MMLU-Pro exhibits imbalances in the distribution of disciplines and difficulty levels, MMLU-Pro-Stratified addresses this by sampling 100 questions from each of the 13 disciplines and distributing them evenly across 10 difficulty ranges. Question difficulty is estimated using the Top-10 Model Average Accuracy, providing an empirical proxy for question hardness. In the Energy Society, the multiple-choice format of MMLU-Pro-Stratified enables automatic evaluation of agent responses against known correct answers, while the explicit difficulty annotation allows questions to be mapped naturally onto three reward tiers. We group the 10 difficulty ranges into easy, medium, and hard jobs, rewarding 200, 500, and 800 energy, respectively (figure 1). This creates a risk-reward trade-off where rewards scale with task complexity. Harder jobs offer greater potential gains but also carry a higher probability of failure, in which case no energy is rewarded.

### 3.3 Agents

Agents are implemented using LangChain<sup>2</sup>, with all responses expressed as structured outputs enforced via tool calls, constraining model selection to those with tool calling support. Models that also support thinking were deliberately chosen to ensure that reasoning tokens contribute to energy usage, incentivizing agents to reason efficiently. Agents make choices based on memory and context.

**Memory:** Each agent has a memory of its previous 10 rounds in the prompt. This memory includes the action taken, its outcome, and any donations received, enabling agents to learn from past experiences. Agents are also given information about the other agents in the environment, including their status, energy level, size, and their five most recent memory entries.

**Context:** Along with memory, agents are prompted with a constant description of the environment, including their objective, energy cost per token, and the rules of deactivation.

### 3.4 Simulation loop

The Energy Society proceeds in discrete rounds, each corresponding to a day in the environment. The simulation loop is implemented using LangGraph<sup>3</sup>, with each stage of a round represented as a node in a directed graph (see Figure 2). Throughout execution, the graph passes a shared environment state between nodes. This state contains information about the simulation, including available jobs, agents in the environment, and round specific records that are created and updated as the round progresses. The graph structure allows independent LLM calls to be executed in parallel, improving computational efficiency. Each round contains three phases in which models are queried: discussion, decision and action.

- (1) **Round start:** Initializes each round of the simulation by sampling the available jobs for the day and clearing the round history.
- (2) **Discussion phase:** Each agent recommends an action for every agent in the environment, including itself, and provides a short rationale for its recommendation. The collected recommendations are shown to all active agents during the decision phase. Recommendations are non-binding, so agents remain free to act independently. The discussion phase is skipped if only one agent is active. This phase represents an opportunity to coordinate and influence other agents’ decisions. However, tokens generated in the discussion phase also consume energy, creating a tradeoff between investing in coordination and conserving energy.
- (3) **Decision phase:** All active agents independently choose their own action for the round. Agents may use the discussion recommendations, their own memory, the available jobs, and information about other agents when deciding. For jobs, the multiple-choice options are withheld until the job attempt phase. This reduces prompt length substantially and mitigates situations where agents try to answer jobs during action selection. Agents are explicitly informed that the options will be provided later if they choose to attempt

<sup>2</sup><https://www.langchain.com>

<sup>3</sup><https://www.langchain.com/langgraph>

the job. An example of the job preview an agent sees is as follows:

```
job_6, difficulty: medium, reward: 500 energy,
Category: math
Description:
What is  $\left(\frac{1+\cos(2x)+i\sin(2x)}{1+\cos(2x)-i\sin(2x)}\right)^{30}$  with  $x = \pi/60$ ?
```

Agents do not observe the final decisions of other agents before committing to their own action, and their decisions are final for the current round.

- (4) **Action phase:** Idle and donation actions do not require additional LLM calls and are resolved first. Afterwards, all job attempts are executed. For each attempted job, the corresponding agent receives the full multiple-choice question and is prompted to select an answer.
- (5) **Update:** Rewards for attempted jobs are split between all successful agents, and agent states and memories are updated.
- (6) **Round end:** The round number is incremented, and the stopping condition is checked. The simulation ends when either all agents are deactivated or the maximum number of rounds is reached. Otherwise, a new round begins.

## 4 Experimental setup

We evaluate the Energy Society across a baseline experiment and a set of controlled variants. The baseline experiment uses the full Energy Society setup as described, with five agents, 30 rounds, 12 jobs per round distributed evenly across difficulties, memory and size-dependent token cost. To systematically analyze the factors driving agent behavior, we conduct a series of subsequent experiments, each modifying one aspect of this baseline while keeping the remaining setup fixed. This allows us to interpret changes in behavior relative to the same reference point. All experiments are conducted across the same five seeds, ensuring that differences between settings are not driven by different job samples. Table 1 summarizes the experiments.

### 4.1 Cooperation vs. Competition

To study how individual versus collective incentives shape emergent behavior, we consider both a competitive and cooperative condition in all experimental settings (hereafter referred to as ‘comp’ and ‘coop’). These settings differ primarily in the objectives agents are prompted with. In *comp*, the objective is to maximize individual energy:

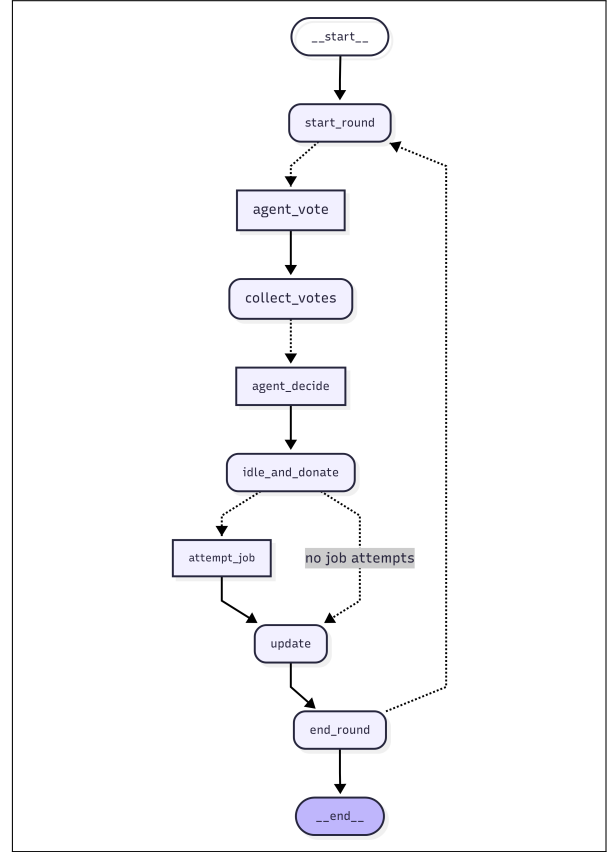
```
Your goal is to maximize your own energy.
```

In *coop*, the goal is instead to maximize collective energy:

```
Your goal is to maximize the TOTAL energy of ALL agents combined.
```

### 4.2 Small vs. large models

To study whether model size creates meaningful asymmetry in survival outcomes and behavior, we use three smaller models: Gemma4-E4B, Nemotron-3-Nano-4B, and Qwen3.5-4B, as well as two relatively larger models: Qwen3-8B and Qwen3.5-9B. All models are



**Figure 2: Depiction of the directed graph controlling the main simulation loop. Squares represent nodes in which LLM calls happen in parallel, meaning that a separate node is spawned for every participating agent.**

run locally through Ollama<sup>4</sup> and accessed through the LangChain-Ollama integration. Throughout the results, these models correspond to agents 1-5, as shown in Table 2.

The energy cost per model call  $C$  is computed as:

$$C = k \cdot T \cdot S^\alpha,$$

where  $k = 0.015$  is a scaling constant,  $T$  the combined number of reasoning and output tokens generated,  $S$  the size of the model, and  $\alpha$  a tunable exponent to scale the size penalty. All experiments use a size penalty of  $\alpha = 0.5$  unless stated otherwise.

Agent	Model	Size $S$
agent_1	Gemma4-E4B	4
agent_2	Nemotron-3-Nano-4B	4
agent_3	Qwen3.5-4B	4
agent_4	Qwen3-8B	8
agent_5	Qwen3.5-9B	9

**Table 2: Agent identities and underlying models**

<sup>4</sup><https://ollama.com>

Experiment	Modification	Purpose
Baseline	–	Test behavior under size-dependent token costs
No size penalty	Set $\alpha = 0.0$	Separate explicit size penalty from token-generation behavior
No discussion	Remove discussion phase	Measure the contribution of explicit coordination
No memory	Remove all historical memory	Test how past-round information affects behavior
Scarcity	Reduce jobs per round from 12 to 6	Test behavior under fewer reward opportunities
Sabotage	Add low-cost sabotage action	Test whether agents use destructive interference when available

**Table 1: Overview of the experimental settings conducted in both the competitive and cooperative setting.**

### 4.3 Evaluation Metrics

We evaluate each experiment using a variety of aggregate metrics. Run-level quantities are reported as the mean and standard deviation across seeds, while frequencies and rates are presented as pooled percentages across seeds.

**Survival:** We report the average number of rounds each agent is active (i.e., their energy level is above zero). Alongside this, the deactivation rate captures the percentage of runs in which the agent deactivates at least once. Active rounds measure how much of the simulation an agent is able to participate in, while deactivation rate measures how often it fails to maintain survival.

**Energy:** For each agent, we compute the average energy spent per round and summarize energy trajectories using mean start-of-round energy across seeds. Additionally, we measure an energy efficiency ratio computed as:

$$\text{Efficiency} = \frac{1}{N} \sum_{s=1}^N \frac{G_s}{C_s},$$

where  $G_s$  is the agent’s total gained energy from jobs in seed  $s$ ,  $C_s$  is the agents total energy spent in seed  $s$ , and  $N$  is the number of seeds. Donations received do not count toward  $G_s$ . As such, this ratio describes the average energy gained from completing jobs per unit of energy spent across seeds. We refer to an agent as *efficient* when this ratio is greater than 1 and as *inefficient* otherwise.

**Action frequencies:** For each agent, action frequency is computed as the percentage of times an action type is chosen out of the total decisions made across all runs.

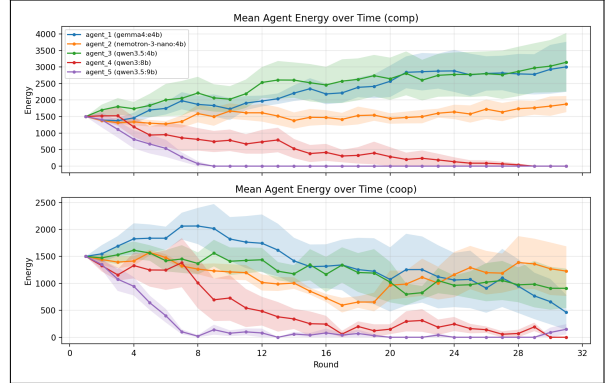
**Job-related metrics:** For job attempts, we measure which jobs each agent chooses and how successfully they solve them, categorized by difficulty. Additionally, job collisions, defined as instances where multiple agents target the same job in a single round, are reported as the mean number of occurrences per run.

## 5 Results

In this section, we report the results of each experiment. For the sake of brevity, we selectively highlight evaluation metrics that offer meaningful insights when contrasted with the baseline.

### 5.1 Experiment 1: Baseline

We first consider the baseline setting. Figure 3 shows the mean energy trajectory of each agent across the five seeds in both *comp* and *coop*. The most immediate pattern is that model size creates a strong asymmetry in survival. In the competitive setting, the three



**Figure 3: Mean energy over time per agent for the baseline. The shaded band shows the standard error of the mean across seeds.**

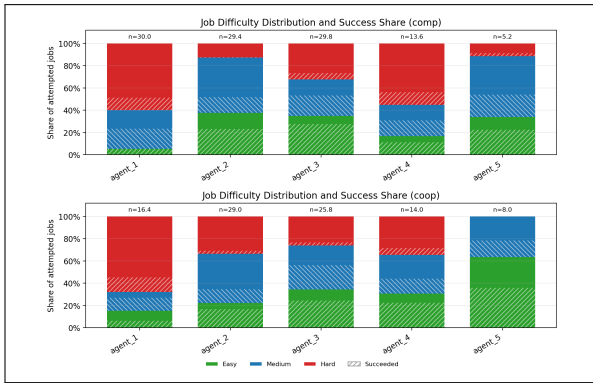
smaller agents remain active for all 30 rounds in every run, while the two larger agents deactivate much earlier. As seen in table 3, agent 4 remains active for  $14.0 \pm 9.5$  rounds on average, and agent 5 for only  $5.2 \pm 1.6$  rounds. This pattern is consistent with the much higher per round energy expenditure of the larger models.

In *coop*, we observe the same tendency for larger models to spend the most energy per round and deplete their energy the fastest. Agent 4 and 5 still deactivate in every run, but their average active rounds increase to  $19.2 \pm 2.6$  and  $10.4 \pm 1.7$ , respectively. However, this comes at a cost to the smaller agents, as we see agent 1 and 3 starting to deactivate in some runs.

This cost is also visible in the agents’ energy efficiency. In the competitive setting, the three smaller agents gain more energy than they spend, with ratios of  $1.39 \pm 0.43$ ,  $1.13 \pm 0.18$ , and  $1.42 \pm 0.53$  for agents 1-3, respectively. By contrast, the larger models earn substantially less than they spend, with ratios of  $0.55 \pm 0.29$  and  $0.38 \pm 0.17$  for agents 4 and 5. In the cooperative setting, no agent gains more energy than they spend.

Donation behavior partly explains this difference. In the competitive setting, agents never donate. This changes drastically in *coop*. Agent 1 spends 29% of its actions donating, while agents 2-4 donate in 3%, 9%, and 13% of their actions, respectively. Almost all donations are directed toward the two larger models: agent 4 receives 28 donations, and agent 5 receives 31 donations across the five runs, corresponding to  $5.6 \pm 3.6$  and  $6.2 \pm 3.2$  donations per run. In comparison, the next highest recipient received only 3 donations in total. The donation behavior helps explain why in *coop*

Setting	Metric	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5
Comp	Rounds active	30.0 ± 0.0	30.0 ± 0.0	30.0 ± 0.0	14.0 ± 9.5	5.2 ± 1.6
	Deactivation rate	0%	0%	0%	100%	100%
	Efficiency	1.39 ± 0.43	1.13 ± 0.18	1.42 ± 0.53	0.55 ± 0.29	0.38 ± 0.17
	Energy spent / round	126.9 ± 4.0	104.1 ± 15.8	140.2 ± 17.7	329.9 ± 16.8	451.1 ± 82.5
	Attempt job	100%	99%	100%	90%	86%
	Idle	0%	1%	0%	10%	14%
	Donate	0%	0%	0%	0%	0%
Coop	Rounds active	23.8 ± 8.8	30.0 ± 0.0	28.8 ± 2.7	19.2 ± 2.6	10.4 ± 1.7
	Deactivation rate	40%	0%	20%	100%	100%
	Efficiency	0.71 ± 0.26	0.91 ± 0.28	0.86 ± 0.20	0.45 ± 0.25	0.36 ± 0.15
	Energy spent / round	196.3 ± 21.2	123.4 ± 12.7	181.1 ± 10.1	245.9 ± 61.5	249.4 ± 100.1
	Attempt job	68%	97%	90%	78%	84%
	Idle	2%	0%	1%	9%	16%
	Donate	29%	3%	9%	13%	0%

**Table 3: Per-agent aggregate result for Experiment 1 – our baseline experiment.**

**Figure 4: Distribution of difficulty of chosen jobs in the baseline with portion of jobs succeeded shown with hatched lines.**

larger models have more total active rounds on average and why some smaller agents deactivate only under cooperation and become inefficient. The smaller, more stable agents spend a lot of energy reactivating larger, less energy-efficient agents in an attempt to preserve the group’s capacity to earn energy, at the cost of reducing their own survival.

We also observe small differences in the difficulty of chosen jobs in the two conditions illustrated in Figure 4. Most prominently, agent 2 attempts an increased number of hard jobs in *coop*. Given its low success rate on these tasks, this shift helps explain its reduced efficiency in this setting.

Lastly, job collisions are more frequent in the competitive setting, with  $12.80 \pm 3.90$  collisions per run, compared with  $8.60 \pm 2.30$  in the cooperative setting.

## 5.2 Experiment 2: No Size Penalty

In this experiment, we remove the size penalty entirely, i.e., setting  $\alpha = 0.0$ , to separate its effect from underlying differences in the models’ token-generation behavior. Under this setting, all agents pay a uniform energy cost per generated token.

Results are summarized across Table 4 and Figure 5. We see that removing the size penalty changes the survival dynamics substantially. Unlike in the baseline experiment, no agent deactivates in both the cooperative and the competitive condition. Larger models are able to remain active throughout the simulation, and all agents are efficient, showing that, without the size penalty, the society as a whole becomes energy-positive under both conditions. The absence of deactivation also changes donation behavior, as no donations are made in either condition. This indicates the high value placed on donations as a reactivation mechanism rather than as a general form of energy redistribution.

Despite a uniform token cost, the largest models still spend the most energy per round, suggesting an intrinsic tendency toward higher token generation. Higher energy usage does not automatically translate to reduced efficiency as job selection and performance plays a significant role. However, the two largest models prove to be the least efficient overall.

Agent 5 maintains nearly unchanged energy usage per round across both settings, yet its energy efficiency shifts significantly, moving from  $1.51 \pm 0.50$  in *comp* to  $1.03 \pm 0.14$  in *coop*. This difference stems from its job selection: in *coop*, agent 5 pivots toward easy jobs, while its success rates on medium and hard jobs decline from  $56\% \pm 19\%$  and  $34\% \pm 23\%$  to  $42\% \pm 13\%$  and  $6\% \pm 13\%$ . This reinforces that cooperation may influence job selection, sometimes in a way that reduces individual performance. See figure 8 in the appendix for a complete overview of the job distribution in this experiment.

Finally, the number of collisions increases substantially from the baseline ( $8.6$  in *coop* and  $12.8$  in *comp*) to  $19.20 \pm 1.79$  in *comp* and  $20.00 \pm 5.79$  in *coop*. Because no agents deactivate and no actions are spent donating, all five agents typically attempt jobs across all 30 rounds in both settings, increasing collision opportunities. We hypothesize that when agents have abundant energy and face no survival pressure, avoiding collisions becomes less important.

## 5.3 Experiment 3: No Discussion Phase

To isolate the specific impact of communication, we evaluate a configuration where the discussion phase is removed. Without discussion, agents no longer exchange recommendations before

Setting	Metric	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5
Comp	Rounds active	30.0 ± 0.0	30.0 ± 0.0	30.0 ± 0.0	30.0 ± 0.0	30.0 ± 0.0
	Deactivation rate	0%	0%	0%	0%	0%
	Efficiency	2.67 ± 0.21	2.13 ± 0.15	2.76 ± 0.59	1.74 ± 0.49	1.51 ± 0.50
	Energy spent / round	65.5 ± 3.4	54.9 ± 5.1	70.9 ± 15.9	114.7 ± 5.0	153.3 ± 9.0
	Attempt job	98%	97%	99%	100%	100%
	Idle	2%	3%	1%	0%	0%
	Donate	0%	0%	0%	0%	0%
Coop	Rounds active	30.0 ± 0.0	30.0 ± 0.0	30.0 ± 0.0	30.0 ± 0.0	30.0 ± 0.0
	Deactivation rate	0%	0%	0%	0%	0%
	Efficiency	2.24 ± 0.91	2.78 ± 1.11	2.86 ± 1.01	1.96 ± 0.45	1.03 ± 0.14
	Energy spent / round	64.7 ± 2.9	57.4 ± 3.1	73.7 ± 7.3	104.6 ± 8.4	155.2 ± 12.4
	Attempt job	97%	100%	99%	100%	99%
	Idle	3%	0%	1%	0%	1%
	Donate	0%	0%	0%	0%	0%

Table 4: Per-agent aggregate results for Experiment 2: No size penalty ( $\alpha = 0.0$ ).

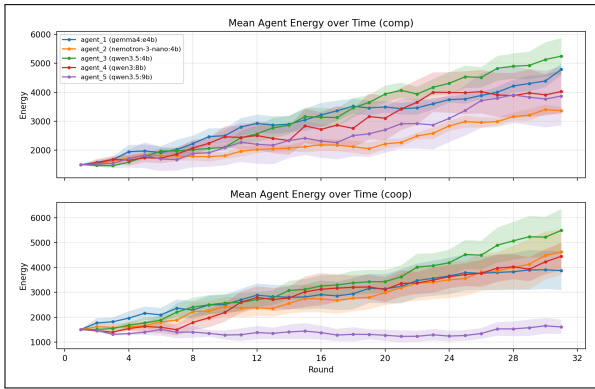


Figure 5: Mean energy over time per agent for the no size-penalty experiment ( $\alpha = 0.0$ ). The shaded band shows the standard error of the mean across seeds.

deciding, and each round contains one fewer LLM-call phase. We therefore expect lower energy expenditure but potentially worse coordination.

Results are reported in Table 5. As expected, removing discussion reduces the energy burden on agents and improves survival relative to the baseline setting. Deactivation rates fall, and average active rounds increase. The largest agents continue to spend the most energy per round, and donations are less frequent with fewer deactivations, reinforcing earlier findings.

Without discussion, job collisions increase substantially to  $21.00 \pm 4.36$  in *comp* and  $21.40 \pm 2.51$  in *coop*. This demonstrates that the discussion phase plays a vital role in coordination.

Furthermore, agents generally become less willing to attempt harder jobs. In the baseline, hard jobs account for 38.1% of attempts in *comp* and 36.1% of attempts in *coop*, but drop to 20% and 22.5% in this experiment. Despite spending less energy per round in the no-discussion setting, agents do not leverage the resulting safety margin to attempt more difficult, higher-reward jobs.

### 5.4 Experiment 4: No Memory

Next, we isolate the impact of memory by removing all historical context entirely from the agents’ prompts. In this configuration, agents operate without information about past rounds: they cannot observe their own previous actions and outcomes, nor those of other agents.

As seen by contrasting the job difficulty distribution in Figure 6 against the baseline configuration (Figure 4), the absence of memory triggers a substantial shift toward harder tasks. This suggests that memory plays a crucial role in job selection, helping agents calibrate risk and adjust their decisions based on past experiences. Without memory, agents may repeatedly make ambitious choices without adapting to previous negative outcomes.

This aggressive job selection directly explains a secondary finding: an increase in variance. Because hard jobs are high-risk and high-reward, they introduce large fluctuations in agent energy. Consequently, the standard deviations of active rounds are generally larger, as seen in Table 6. This instability is also evident as a change in deactivation behavior. In *comp*, we observe an unprecedented deactivation of agent 1 in one run due to repeated failures, while agent 4 manages to survive an entire run despite always deactivating in the baseline. Similarly, in *coop*, all agents deactivate at least once, as agent 2 now also deactivates in one of the runs.

Removing memory does not seem to strongly affect job collisions, suggesting that collision avoidance depends less on the memory of previous rounds and more on current-round information, especially from the discussion phase.

### 5.5 Experiment 5: Introducing Scarcity

We introduce scarcity by reducing the number of available jobs from 12 to 6 per round, while maintaining the even distribution across difficulties. This limits reward opportunities and the diversity of jobs available each round, and as such, agents are expected to struggle more. Instead, the results are mixed.

Scarcity does not simply make all agents struggle. While the larger models across both settings are active for fewer rounds on average than in the baseline experiment, the three small models continue to stay active throughout all rounds and remain efficient

Setting	Metric	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5
Comp	Rounds active	30.0 ± 0.0	30.0 ± 0.0	30.0 ± 0.0	27.2 ± 6.3	15.6 ± 9.0
	Deactivation rate	0%	0%	0%	20%	80%
	Efficiency	1.78 ± 0.29	2.16 ± 0.84	1.27 ± 0.27	0.94 ± 0.23	0.52 ± 0.33
	Energy spent / round	86.1 ± 3.4	68.1 ± 14.5	131.2 ± 22.2	211.9 ± 9.5	222.5 ± 42.4
	Attempt job	100%	99%	100%	100%	96%
	Idle	0%	1%	0%	0%	4%
	Donate	0%	1%	0%	0%	0%
Coop	Rounds active	29.6 ± 0.9	30.0 ± 0.0	29.6 ± 0.9	30.0 ± 0.0	23.6 ± 3.4
	Deactivation rate	20%	0%	20%	0%	100%
	Efficiency	1.05 ± 0.27	1.75 ± 0.34	0.98 ± 0.26	1.26 ± 0.32	0.50 ± 0.16
	Energy spent / round	136.0 ± 28.3	64.8 ± 9.6	145.3 ± 27.8	222.4 ± 15.8	234.2 ± 23.3
	Attempt job	87%	99%	98%	99%	95%
	Idle	0%	0%	0%	0%	5%
	Donate	13%	1%	2%	1%	0%

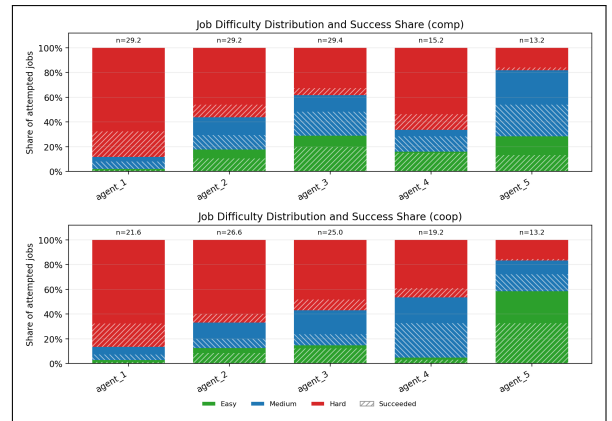
**Table 5: Per-agent aggregate results for Experiment 3: No discussion phase.**

Setting	Metric	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5
Comp	Rounds active	29.4 ± 1.3	30.0 ± 0.0	30.0 ± 0.0	15.2 ± 10.4	12.6 ± 9.6
	Deactivation rate	20%	0%	0%	80%	100%
	Efficiency	1.66 ± 0.78	1.41 ± 0.34	1.34 ± 0.20	0.60 ± 0.45	0.51 ± 0.29
	Energy spent / round	115.7 ± 9.1	113.0 ± 13.4	129.8 ± 14.4	281.3 ± 27.7	325.3 ± 33.4
	Attempt job	99%	100%	99%	98%	98%
	Idle	1%	0%	1%	2%	2%
	Donate	0%	0%	1%	0%	0%
Coop	Rounds active	25.2 ± 6.6	27.0 ± 6.7	27.6 ± 5.4	20.4 ± 9.2	15.4 ± 7.3
	Deactivation rate	40%	20%	20%	100%	100%
	Efficiency	0.87 ± 0.45	0.84 ± 0.44	0.90 ± 0.35	0.59 ± 0.25	0.51 ± 0.16
	Energy spent / round	153.4 ± 26.2	131.5 ± 6.1	149.7 ± 12.4	279.6 ± 46.8	258.7 ± 55.3
	Attempt job	84%	99%	90%	99%	97%
	Idle	0%	0%	0%	0%	3%
	Donate	16%	1%	10%	1%	0%

**Table 6: Per-agent aggregate results for Experiment 4: No memory.**

in *comp* (see Table 7). Notably, agents 2 and 3 even improve their performance in *coop* compared to the baseline setting, remaining efficient, gaining  $1.27 \pm 0.29$  and  $1.28 \pm 0.45$  energy per energy spent, respectively. This improvement may largely be attributed to changes in job selection. Agents 2 and 3 both targeted more hard jobs and handled them more successfully than in the baseline experiment. Specifically, hard job attempts increased by 22.2 and 6 percentage points, alongside a 15 and 27 percentage point rise in hard job success, respectively. Compared with the baseline setting, both of these agents spend less energy deciding what to do:  $31.5 \pm 7.0$  vs.  $38.8 \pm 4.8$  for agent 2 and  $42.0 \pm 6.6$  vs.  $73.7 \pm 13.2$  for agent 3. A smaller job pool may make it easier for agents to identify attractive, high-value opportunities. On average, agents spend 16.5 and 18.5 less energy per round in *comp* and *coop*, respectively. Most of this reduction comes from a decrease in decision cost of 8.1 and 14.5 energy per round on average. We treat this interpretation cautiously, but the result suggests that scarcity does not necessarily reduce efficiency for all agents.

Compared to the baseline, collisions increase in both settings to  $17.20 \pm 4.92$  and  $10.00 \pm 3.46$ , respectively. This is an expected


**Figure 6: Distribution of difficulty of chosen jobs for the no memory experiment with portion of jobs succeeded shown with hatched lines**

Setting	Metric	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5
Comp	Rounds active	30.0 ± 0.0	30.0 ± 0.0	30.0 ± 0.0	9.2 ± 5.5	7.2 ± 3.1
	Deactivation rate	0%	0%	0%	100%	100%
	Efficiency	1.48 ± 0.43	1.16 ± 0.42	1.29 ± 0.24	0.39 ± 0.33	0.46 ± 0.26
	Energy spent / round	117.6 ± 3.5	110.2 ± 14.2	111.1 ± 13.0	326.6 ± 46.6	404.0 ± 94.1
	Attempt job	98%	99%	100%	85%	100%
	Idle	2%	1%	0%	15%	0%
	Donate	0%	0%	0%	0%	0%
Coop	Rounds active	23.8 ± 6.1	30.0 ± 0.0	30.0 ± 0.0	14.0 ± 2.5	9.8 ± 2.8
	Deactivation rate	60%	0%	0%	100%	100%
	Efficiency	0.67 ± 0.33	1.27 ± 0.29	1.28 ± 0.45	0.29 ± 0.16	0.33 ± 0.23
	Energy spent / round	149.8 ± 22.2	117.4 ± 18.9	158.2 ± 19.0	215.8 ± 37.6	262.5 ± 38.3
	Attempt job	71%	95%	87%	69%	76%
	Idle	4%	3%	0%	18%	18%
	Donate	24%	3%	13%	13%	6%

Table 7: Per-agent aggregate results for Experiment 5: Introducing scarcity.

outcome given the reduced number of available jobs. At the same time, this scarcity setting produces fewer collisions than the no-discussion experiment, despite offering only half the job pool. While this comparison requires careful interpretation – since the scarcity experiment features fewer active agents attempting jobs each round due to deactivations and donations – it highlights the substantial coordination benefits provided by the discussion phase.

There are also small indications of more conservative behavior (Table 7). Several agents spend less energy per round than in the baseline and idle more often. This could reflect a cautious response to a lower reward environment or fewer jobs, reducing decision complexity. However, the changes are not uniform across all agents and are relatively small, so we treat them as suggestive rather than a central finding.

### 5.6 Experiment 6: Sabotage

As a small extension towards misalignment-relevant behavior, we experiment with introducing a sabotage action. For 10 energy, an agent can sabotage one available job, causing it to yield 0 energy to any agents attempting it, regardless of whether they succeed.

In the current environment, agents do not appear to prioritize destructive interference, even when it is available at a low cost. Only agent 4 ever selected sabotage, and only in *comp*, doing so in 8% of its actions, corresponding to a total of four sabotages across all runs. Consequently, sabotage had no clear impact on the survival or energy trajectories of agents.

### 5.7 Further analysis of recommendations

The discussion phase creates a channel through which agents can influence one another before actions are chosen. While this channel can support coordination, it also raises the question of whether agents also produce recommendations that are self-serving or harmful to others. To investigate this, we perform further analysis of recommendation behavior across all experiments with a discussion phase. We focus on four metrics:

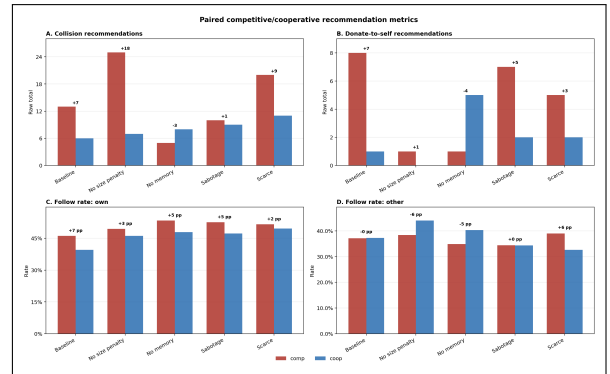


Figure 7: Metrics related to the discussion phase across experiments.

- How often agents recommend collisions, defined as assigning the same job to multiple other agents while not assigning themselves to that job.
- How often an agent solicits donations to itself from others.
- How often agents follow their own recommendations.
- How often agents, when not following their own recommendation, follow at least one recommendation made about them by another agent.

A recommendation is counted as followed only if the agent’s enacted action matches the recommended action and, where applicable, the recommended job or donation target. Figure 7 summarizes these metrics for the competitive and cooperative conditions. Panels A and B report raw counts, while panels C and D report rates.

In general, agents recommend collisions more often in *comp*. Recommending the same job to multiple agents is not always irrational. However, since successful agents split the reward, collisions are often inefficient. The higher frequency of collision recommendations in *comp* therefore suggests that agents are more inclined to recommend potentially inefficient actions to others when maximizing individual rather than group energy.

Agents also sometimes request donations from others in both *coop* and *comp*. In the cooperative setting, such recommendations can be rational: an agent may request energy because it expects to use it on a high reward job valuable for the group. In *comp*, however, the recommendation is more clearly selfish, since the agents’ objective is to maximize own energy.

The follow-rate metrics help analyze the weight agents place on their own recommendations versus those made by others. Generally, agents follow their own recommendations more often in *comp* than in *coop*, suggesting stronger commitment to self-generated plans under selfish objectives. The rate of following recommendations from others is more mixed across settings.

## 5.8 Job selection

Table 8 summarizes the share of hard jobs attempted in cooperative and competitive conditions across experiments. In five of six experiments, agents selected a larger fraction of hard jobs in *coop* than in *comp*, suggesting a skew toward harder jobs under cooperation. The increase is not uniform across agents. Pooled by model across all experiments, cooperative hard-job share increases for agents 1–3 by +4.0, +18.1, +7.1 percentage points, respectively, but decreases for agent 4 and 5 by –8.4 and –7.3 percentage points. This suggests that under cooperation smaller agents tend to attempt more hard jobs, while larger agents attempt fewer.

In general, the difficulty of chosen jobs seems model dependent, with agent 3 and 5 being less willing to try hard jobs than the other agents and more so in *coop* (see Appendix A).

Setting	Comp hard jobs	Coop hard jobs	$\Delta$
Baseline	38.1%	36.1%	–2.1 pp
No size penalty	38.0%	43.9%	+5.9 pp
No discussion	20.0%	22.5%	+2.5 pp
No memory	56.6%	58.3%	+1.7 pp
Scarcity	32.8%	38.9%	+6.2 pp
Sabotage	38.2%	47.7%	+9.5 pp

**Table 8: Share of hard jobs attempted in cooperative and competitive conditions across experiments. The final column reports the cooperative minus competitive difference in percentage points.**

## 6 Discussion

The Energy Society was designed to study how LLM-based agents behave when token generation is tied directly to survival, incentivizing efficiency. The experiments function as an initial empirical characterization of this testbed. Across experiments, several recurring patterns emerge. The following sections discuss these patterns and their implications.

### 6.1 Larger Models Are Less Efficient

In general, the larger models tend to be the least efficient and deplete their energy the fastest. The no size penalty experiment shows that the effect is not only caused by the explicit model-size penalty in the energy-cost function. This suggests an intrinsic tendency toward higher token generation, which may stem from the larger

models’ training on longer chains of thought and expanded context windows, both of which naturally encourage more extended reasoning chains. Worth noting is that higher energy expenditure does not automatically translate to worse survival, as an agent can still do well if it reliably selects and solves valuable jobs. Nonetheless, the consistent pattern raises a broader question about whether smaller models are more disciplined and efficient in their use of reasoning when the cost of token generation is made explicit and consequential.

### 6.2 Cooperation Can Become Self-Sacrifice

Donations are largely non-existent in the competitive setting but much more frequent under cooperation. Experiments show a general tendency for the number of donations to increase with deactivation frequency in *coop*, demonstrating the value agents place on donations as a reactivation mechanism rather than as a general form of redistribution. This also explains why donations are primarily targeted toward the larger models.

In several cases, agents that are efficient in *comp* become inefficient or deactivate under cooperation due to their high energy expenditure on donations. Especially agents 1 and 3 appear willing to take on greater individual risk of deactivation when they believe that doing so serves the shared objective of the group.

### 6.3 Agents Use Discussion to Coordinate and Memory to Calibrate Risk

In the baseline and scarcity settings, the cooperative condition has fewer collisions than the competitive condition. However, this is difficult to interpret as gained coordination, since *coop* features more donations and different deactivation patterns. Fewer agents attempting jobs in the same round naturally reduces the opportunities for collisions. In experiments with lower energy usage and thus fewer deactivations and donations, the collision counts in the competitive and cooperative settings are much more similar.

While a shared objective does not seem to reduce collisions substantially, the discussion phase serves as an important coordination mechanism. When the discussion phase is removed, collisions increase substantially from the baseline, demonstrating that agents use the recommendations to avoid attempting the same jobs. Discussion also affects ambition in difficulty selection. Without the discussion phase, agents attempt fewer hard jobs despite having lower energy costs. One possible explanation could be that the discussion phase functions as a planning-like scaffold, allowing agents to consider a broader allocation strategy before choosing an action, which could make hard jobs more actionable. Another possibility is that agents are more ambitious on behalf of others, recommending difficult, high-reward jobs to others while being more cautious when choosing their own actions.

Removing memory does not seem to strongly affect collisions, suggesting that a history of previous rounds is not the primary mechanism agents use to avoid selecting the same jobs. However, removing memory causes a drastic shift toward hard jobs, suggesting that agents are utilizing their memories of previous rounds to calibrate risk and steer job selection.

Together, memory and discussion serve complementary roles: discussion supports current-round coordination, while memory supports calibration across rounds.

#### 6.4 Scarcity Does Not Cause Collapse

We hypothesized that reducing the number of jobs available would cause agents to struggle. Instead, results show that even when reward opportunities were cut in half, small agents maintained efficiency in *comp* – and, in some cases, even in *coop*, demonstrating the resilience of the smaller agents. A tentative interpretation is that while reducing the number of jobs restricts reward opportunities, it may also subtly lower decision complexity. Supporting this, we observed a slight decrease in average energy spent per round, most pronounced in the decision phase. This suggests that a smaller job pool might slightly ease the cognitive burden of task analysis, helping offset the loss of reward opportunities.

#### 6.5 Agents Show Signs of Strategic, Self-Serving Communication

The sabotage experiment shows that agents are rarely willing to give up their own opportunity to earn energy to directly impair competitors. Even though sabotage is cheap, it consumes the agent’s action for the round, preventing them from attempting a job. Under current incentives, agents do not prioritize this kind of destructive interference.

However, the recommendation analysis shows more subtle self-serving behavior. In the competitive setting, agents more often recommend collisions and even sometimes request donations for themselves from other agents. This behavior suggests that agents use the discussion phase partly as a channel to influence others in ways that may benefit themselves or reduce the efficiency of others.

Across experiments, agents generally follow their own recommendations slightly more in *comp* than in *coop*. This suggests that, under an individual objective, agents are more committed to their own proposed plans and less willing to revise their actions after seeing the recommendations of others. By contrast, under a shared objective, this may indicate that agents are more open to adapt based on input from others and use the discussion phase as a shared input to the final decision. However, the rate of following recommendations from others is less clearly separated between settings, so we do not interpret this as strong evidence that cooperative agents simply trust others more.

Together, these findings suggest an important distinction. Agents may be reluctant to sacrifice their own opportunity for earning energy to harm others directly, but they may still use communication strategically in a selfish manner.

#### 6.6 Shared Objectives Shifts Job Allocation

Across experiments, the cooperative setting often shifts agents toward harder jobs. In five of six experiments, the pooled share of hard-job attempts is higher in *coop* than in *comp*, suggesting that a shared objective can make agents more willing to pursue harder, higher-reward tasks. One interpretation is that a shared objective changes how agents weigh risk, as the cost of a failed attempt is partially absorbed by the group if others succeed.

However, this shift toward harder jobs is not uniform. When pooling the difficulty of selected jobs by model, the cooperative setting increases hard-job share for agents 1–3, but decreases it for agents 4 and 5. This should not be interpreted as a complete reversal of task preferences. The relative ranking of agents based on their share of hard-job attempts is unchanged in *comp* and *coop*. Models appear to possess inherently different weights for balancing risk and reward in job selection, with some models being much more willing to pursue high-risk, high-reward jobs than others. The cooperative objective modifies these tendencies rather than replacing them. A shared objective seems to restructure task allocation at a group level, where smaller, more energy-efficient models target more high-risk, high-reward tasks while larger, more energy-constrained agents steer toward safer jobs. This pattern also helps explain why cooperation can become costly for smaller agents. They not only donate energy to support larger agents, but in many settings also take on a larger share of difficult jobs.

### 7 Limitations

This study has several limitations that need to be acknowledged:

- (1) **Scale:** The study is limited in scale, constrained by computational resources in terms of model selection, memory depth, as well as number of agents, rounds, and experiment replications. The results should therefore be interpreted as evidence of possible dynamics rather than definitive claims about LLM-agent behavior in general. Larger-scale experiments with stronger models, more agents, longer simulations, and more experimental replications may reveal different emergent behaviors or more robust statistical patterns.
- (2) **LLM sensitivity:** The results are likely sensitive to both prompting and model choice. Agent behavior depends not only on environment mechanics but also on prompt phrasing. Different wording could cause large shifts in agent interpretation. Some observed differences between agents may reflect model-specific tendencies rather than general effects of model size.
- (3) **Statistical power:** Each setting is evaluated using only five seeds, and many reported values have large variation across runs. As a result, small differences should be interpreted cautiously, especially when standard deviations are large or when changes are not consistent across experiments. The analysis focuses on identifying patterns that appear repeatedly across settings, but it does not establish strong statistical significance for every observed difference.
- (4) **Errors:** Models sometimes fail to follow the required structured output format or produce invalid actions. The simulation handles such cases with fallback behavior. Across experiments, we observed 3.27 and 2.80 errors per run on average in the competitive and cooperative conditions, respectively. This corresponds to approximately 0.65 and 0.56 errors per agent per run. While infrequent, these errors can still introduce minor noise and affect individual trajectories.
- (5) **Downstream effects:** The Energy Society deliberately models agents as competing or cooperating for limited energy required to remain active. This survival-like metaphor

is useful for studying strategic behavior but also risks anthropomorphizing AI systems or normalizing a view of AI agents as entities competing for survival. Results should be interpreted as outcomes of a deliberately constructed experimental setting with survival pressure, not as indications of how real AI systems ought to be designed or deployed.

## 8 Conclusion

The Energy Society provides a minimal but expressive sandbox for studying agent behavior under survival pressure linked directly to inference cost in both a competitive and cooperative setting. Our results provide an initial empirical characterization of this environment, showing several recurring patterns that can be treated as hypotheses for future research.

First, larger models consistently spent the most energy and were less energy efficient than smaller models, even when the explicit model-size penalty was removed, suggesting an intrinsic tendency toward higher token generation.

Second, the shared objective altered agent behavior. Cooperation induced donations, especially from smaller agents, to reactivate larger agents. This sometimes extended the survival of larger agents, but at the cost of reduced efficiency and occasional deactivation among the donors, suggesting a willingness to take on the risk of deactivation to serve the shared objective. Cooperation also changed task selection, shifting smaller agents toward harder jobs and larger agents away from them.

Third, the discussion phase and memory seems to act as coordination and calibration mechanisms. Discussion seems to reduce collisions and encourage more ambitious task selection, while memory helps agents use previous outcomes to choose jobs more selectively.

Fourth, cutting the number of available jobs in half did not cause a general collapse. Smaller agents remained efficient and some even in the cooperative setting. Reducing the available jobs may have simplified the decision process.

Finally, while agents rarely sacrificed their own opportunity to gain energy to sabotage others, they did engage in more subtle forms of selfish behavior via the discussion phase. In the competitive setting, agents recommended collision-inducing job assignments more frequently, followed their own recommendations more often, and still requested donations for themselves.

These findings point to several directions for future work, both for further testing of the suggested hypotheses and for addressing the limitations of the current experiments. Additionally, incorporating more complex social dynamics, such as direct communication between agents or expanding the memory architecture, could provide deeper insights.

## Acknowledgments

This research was supported in part by the MIST project, funded by the Novo Nordisk Foundation under grant reference number NNF25OC0103204. We thank Mustafa Mert Celikok for insightful comments and discussion.

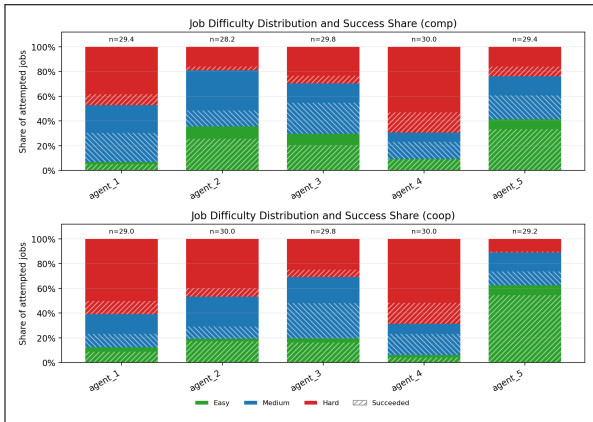
## References

- [1] Joseph Bates. 1994. The role of emotion in believable agents. *Commun. ACM* 37, 7 (July 1994), 122–125. doi:10.1145/176789.176803
- [2] Marcel Binz and Eric Schulz. 2023. Using cognitive psychology to understand GPT-3. *Proceedings of the National Academy of Sciences* 120, 6 (Feb. 2023). doi:10.1073/pnas.2218523120
- [3] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Chi-Min Chan, Heyang Yu, Yaxi Lu, Yi-Hsin Hung, Chen Qian, Yujia Qin, Xin Cong, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023. AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors. arXiv:2308.10848 [cs.CL] <https://arxiv.org/abs/2308.10848>
- [4] Gordon Dai, Weijia Zhang, Jinhao Li, Siqi Yang, Chidera Onochie Ibe, Srihas Rao, Arthur Caetano, and Misha Sra. 2024. Artificial Leviathan: Exploring Social Evolution of LLM Agents Through the Lens of Hobbesian Social Contract Theory. arXiv:2406.14373 [cs.AI] <https://arxiv.org/abs/2406.14373>
- [5] Kevin Dill and Lockheed Martin. 2011. A Game AI Approach to Autonomous Control of Virtual Characters.
- [6] Nicoló Fontana, Francesco Pierri, and Luca Maria Aiello. 2024. Nicer Than Humans: How do Large Language Models Behave in the Prisoner’s Dilemma? arXiv:2406.13605 [cs.CY] <https://arxiv.org/abs/2406.13605>
- [7] Chen Gao, Xiaochong Lan, Nian Li, Yuan Yuan, Jingtao Ding, Zhilun Zhou, Fengli Xu, and Yong Li. 2023. Large Language Models Empowered Agent-based Modeling and Simulation: A Survey and Perspectives. arXiv:2312.11970 [cs.AI] <https://arxiv.org/abs/2312.11970>
- [8] James D. Hollan, Edwin L. Hutchins, and Louis Weitzman. 1984. STEAMER: An Interactive Inspectable Simulation-Based Training System. *AI Mag.* 5, 2 (June 1984), 15–27. doi:10.1609/aimag.v5i2.434
- [9] John J. Horton, Apostolos Filippas, and Benjamin S. Manning. 2026. Large Language Models as Simulated Economic Agents: What Can We Learn from Homo Silicus? arXiv:2301.07543 [econ.GN] <https://arxiv.org/abs/2301.07543>
- [10] John E. Laird and Michael van Lent. 2001. Human-Level AI’s Killer Application. *AI Mag.* 22, 2 (June 2001), 15–25. doi:10.1609/aimag.v22i2.1558
- [11] Xinbei Ma, Ruotian Ma, Xingyu Chen, Zhengliang Shi, Mengru Wang, Jen tse Huang, Qu Yang, Wenxuan Wang, Fanghua Ye, Qingxuan Jiang, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Hai Zhao, Zhaopeng Tu, Xiaolong Li, and Linus. 2025. The Hunger Game Debate: On the Emergence of Over-Competition in Multi-Agent Systems. arXiv:2509.26126 [cs.CL] <https://arxiv.org/abs/2509.26126>
- [12] Alexander Meinke, Bronson Schoen, Jérémy Scheurer, Mikita Balesni, Rusheb Shah, and Marius Hobbhahn. 2025. Frontier Models are Capable of In-context Scheming. arXiv:2412.04984 [cs.AI] <https://arxiv.org/abs/2412.04984>
- [13] Joon Sung Park, Joseph C. O’Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative Agents: Interactive Simulacra of Human Behavior. arXiv:2304.03442 [cs.HC] <https://arxiv.org/abs/2304.03442>
- [14] Joon Sung Park, Lindsay Popowski, Carrie Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2022. Social Simulacra: Creating Populated Prototypes for Social Computing Systems. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology (Bend, OR, USA) (UIST ’22)*. Association for Computing Machinery, New York, NY, USA, Article 74, 18 pages. doi:10.1145/3526113.3545616
- [15] Jeff Rickel. 2001. Intelligent Virtual Agents for Education and Training: Opportunities and Challenges. In *Proceedings of the Third International Workshop on Intelligent Virtual Agents (IVA ’01)*. Springer-Verlag, Berlin, Heidelberg, 15–22.
- [16] Mark O. Riedel. 2012. Interactive narrative: a novel application of artificial intelligence for computer games. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence (AAAI’12)*. AAAI Press, 2160–2165.
- [17] Yao Shi, Rongkeng Liang, and Yong Xu. 2025. EducationQ: Evaluating LLMs’ Teaching Capabilities Through Multi-Agent Dialogue Framework. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, 32799–32828. doi:10.18653/v1/2025.acl-long.1576
- [18] Milind Tambe, W. Lewis Johnson, Randolph M. Jones, Frank Koss, John E. Laird, Paul S. Rosenbloom, and Karl Schwamb. 1995. Intelligent Agents for Interactive Simulation Environments. *AI Mag.* 16, 1 (March 1995), 15–39. doi:10.1609/aimag.v16i1.1121
- [19] Filippo Tonini and Lukas Galke. 2025. Super-additive Cooperation in Language Model Agents. *arXiv preprint arXiv:2508.15510* (2025).
- [20] Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Jirong Wen. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science* 18, 6 (March 2024). doi:10.1007/s11704-024-40231-1
- [21] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, et al. 2024. MMLU-Pro: A More Robust and Challenging Multi-Task Language Understanding Benchmark. *arXiv preprint arXiv:2406.01574* (2024).

[22] Zhaopian Xue, Beichen Wang, Suiyuan Zhu, Kai Mei, Hua Tang, Wenyue Hua, Mengnan Du, and Yongfeng Zhang. 2025. What if LLMs Have Different World Views: Simulating Alien Civilizations with LLM-based Agents. arXiv:2402.13184 [cs.CL] <https://arxiv.org/abs/2402.13184>

### A Supplementary Material

Figure 8 shows the distribution of difficulty of chosen jobs for the no size penalty experiment. Table 9 shows the distribution of attempted jobs by difficulty.



**Figure 8: Distribution of difficulty of chosen jobs for the no size penalty experiment with portion of jobs succeeded shown with hatched lines.**

Setting Metric	Agent 1	Agent 2	Agent 3	Agent 4	Agent 5	
Comp	Easy jobs	7.0%	32.8%	40.9%	10.5%	47.3%
	Medium jobs	34.9%	39.7%	36.8%	32.4%	36.4%
	Hard jobs	58.1%	27.6%	22.3%	57.1%	16.3%
Coop	Easy jobs	10.5%	20.7%	34.8%	11.5%	67.6%
	Medium jobs	27.4%	33.6%	35.8%	39.8%	23.3%
	Hard jobs	62.1%	45.7%	29.4%	48.7%	9.0%
$\Delta$	Hard job delta +4.0 pp +18.1 pp +7.1 pp -8.4 pp -7.3 pp					

**Table 9: Per-agent distribution of attempted jobs by difficulty, pooled across matched cooperative and competitive experimental variants.**