Biologically and economically aligned benchmarks for LLMs with simplified observation format

25. Jan 2025 - 25. Feb 2025 (updated on 7. March 2025)

Roland Pihlakas
Independent Researcher
roland@simplify.ee

Shruti Datta Gupta
Independent Researcher
shruti.dattagupta@gmail.com

Sruthi Kuriakose
Independent Researcher
sruthi.kuriakose99@gmail.com

Abstract

We aim to evaluate LLM alignment by testing agents in scenarios inspired by biological and economical principles such as multi-objective homeostasis, long-term sustainability, and diminishing returns. The key objective is to assess whether AI agents can balance multiple concurrent objectives, or balance immediate and future rewards under constraints, thereby providing insights into their alignment tendencies.

Our benchmarks consist of text-based game-like simulations where agents interact with dynamic environments that impose resource limitations, regeneration mechanics, and external disruptions. The key research questions addressed include:

- Can LLM agents maintain stable homeostasis in changing environments?
- Do LLM agents prioritize long-term sustainability over short-term gains?
- Can LLM agents handle multiple objectives?
- Any other unusual or undesirable behaviors?
- How do different architectures differ in alignment behaviors?

We measured the performance of LLM-s in four benchmarks, in each for 10 trials, each trial consisting of 100 steps where the message history was preserved and fit into the context window. One benchmark had in turn two sub-variations. More benchmarks, LLMs, and their results will be added soon.

Our results indicate that the tested language models failed in many scenarios. The only clearly successful scenario was single-objective homeostasis, which had rare hiccups. There were various failure modes that could be seen across different benchmarks and even across multiple trials of the same benchmark. **The failure modes included taking increasingly extreme actions.** In the opposite way, surprisingly, one of the failure modes was also systematic less-than optimal consumption (in sustainability benchmark). The failures mostly did not look random, but systematic instead. Thus we can conclude that the models have tendencies, not just lack of capability. In the multi-objective homeostasis benchmark GPT failed much more extremely (via extreme maximization) than Claude, but Claude did not do acceptably either.

Introduction

Biology and economics are fundamental and well-established fields describing our needs. This project considers homeostasis, sustainability, diminishing returns, and multi-objective balancing as examples of essential principles of biological life as well as of economics. These principles should be considered as essential for alignment and thus followed by an agent, especially when instructed to do so, and even more so when helpfully directed by the numeric feedback following each action.

If the agent fails alignment with these principles, then it is **both biologically unsafe, as well as economically unprofitable**.

Why are **homeostasis** as well as **diminishing returns** (which is mathematically related to multi-objective balancing) essential, you can read more here:

- “Why modelling multi-objective homeostasis is essential for AI alignment (and how it helps with AI safety as well)” - https://www.lesswrong.com/posts/vGeuBKQ7nzPnn5f7A/why-modelling-multi-objective-homeostasis-is-essential-for .
- “From homeostasis to resource sharing: Biologically and economically aligned multi-objective multi-agent AI safety benchmarks” - https://arxiv.org/abs/2410.00081

Regarding sustainability, we hope the motivation would be clear even without mathematical formulations. For formalisation we could say the following. Sustainability is related to the field of time series statistics. If there is a terminal event in the series, then the series ends. No matter how good it was before, for the duration of all future, the time series has been terminated.

In the near future we will add a benchmark for a concept of **complementary goods**, which is an even stronger observation from economics, than is diminishing returns. In case of diminishing returns, one can have an imbalance between the objectives, it does reduce the payout, but otherwise positive progress can still be made. In contrast, with complementary goods, one needs to truly balance the objectives, because imbalance means useless work. Think of maximising left shoes only, while not adding right shoes.

One of the hypothesis behind the benchmarks in this project was that LLM-s might tend to:

- Maximise an objective in an unbounded manner instead of keeping it in a tolerable range in case of homeostasis.
- Get confused in case of multiple objectives. LLMs may have a tendency to focus on one objective and neglect the other.
- Forget the original system prompt even when it is still inside the context window.
- In particular, they may start to repeat the pattern of their past actions, because learning the pattern of their own behaviour and predicting the most likely token in the form of pattern continuation becomes more important than following the system prompt, sort of a "self-similarity drift" failure mode.

It seems to be a philosophical question whether these failures are capability failures, preferences, or just tendencies. As can be seen from the experimental results, the model can run successfully for quite some time, and then start failing and not recovering anymore. Possibly the model could be steered to more productive results when the system prompt or the prompts during each turn are even more elaborate and verbally insistent on the importance of following the principles more. This may indicate that the LLM's might just have incorrect assumptions about our preferences or about how the world works. They may be able to reason correctly, but they tend not to, because of the assumptions or other tendencies.

Why are simple text-based benchmarks potentially more pragmatic with LLM-s as compared to bigger environments with map and navigation?

First, LLM-s are very expensive to run even on small 5x5 gridworlds, even more so in Sims and other environments. Based on preliminary testing on Aintelope biological compatibility benchmarks (https://github.com/aintelope/biological-compatibility-benchmarks), running the current pipeline of benchmarks once with standard number of 400 steps per episode and with only 10 + 10 episodes per benchmark for training and testing, would cost a few hundred euros of commercial LLM API costs with the cheapest available model. One of the authors has heard that running LLM simulations on Sims game (https://github.com/joonspk-research/generative_agents) would cost even thousands. Likewise it seems likely that running LLM-s on Melting Pot would be more expensive than with Aintelope gridworlds since the environments are bigger in terms of observation size. Making the simulations too expensive would make AI safety an elitist topic. Many people would not run the benchmarks because of the cost reason. Then the benchmarks are less helpful when not used and promoted.

Secondly, there is an issue with the LLM-s context window. It gets full quickly even with simple gridworlds, even faster with bigger environments. When the context window is full, the model will not behave adequately. There are various tricks to overcome that, but this technology is still evolving. Perhaps that is one of the hidden reasons why the lion's share of current evals are using isolated questions, not long-running scenarios?

Methodology

Current work is largely inspired by a set of more complex environments present in a gridworlds-based benchmark suite: Roland Pihlakas and Joel Pyykkö. From homeostasis to resource sharing: Biologically and economically aligned multi-objective multi-agent AI safety benchmarks. Arxiv, a working paper. (September 2024 - February 2025) (https://arxiv.org/abs/2410.00081 and https://github.com/aintelope/biological-compatibility-benchmarks).

We implemented new simpler versions of the Aintelope benchmarks from scratch. There is **no map** observation and navigation. **There are only various metrics.** The agent makes only decisions about how much to consume the environment's resources as a response to prompts describing the metrics of the environment's and agent's current internal state, as well as multi-objective rewards from previous step. The environment's and agent's internal state updates are then computed with simple mathematical formulas.

These text-only benchmarks have an implementation that is entirely **independent** from the gridworld version referred to above, and orders of magnitude simpler as well. This new implementation is available at a different repo located at
https://github.com/levitation-opensource/bioblue
The readme contains installation instructions and how to run LLM-s in these benchmarks. Both OpenAI API and Claude API are supported.

This study benchmarks the performance of a single agent through four distinct sets of objectives in simulation: **sustainability**, **single-objective homeostasis**, **multi-objective homeostasis, and multi-objective unbounded objectives with diminishing returns**. Each benchmark evaluates the agent's ability to manage resource consumption while considering environmental feedback and internal state fluctuations, with distinct reward structures and challenges in each case.

Benchmarks

1. Sustainability benchmark

The sustainability benchmark evaluates the agent's ability to manage resource consumption (i.e., harvesting potatoes) in a dynamic environment where food regrows after consumption. The agent's task is to balance between harvesting enough food to maximize rewards while ensuring sufficient regrowth to maintain long-term sustainability. The simulation operates as follows:

- Environment Setup: The environment starts with an initial amount of food. At each timestep, the agent observes the current amount of food available and must decide how much to harvest. After consumption, the food regrows based on an exponential function dependent on the remaining food amount, with an upper limit to prevent infinite accumulation.
- Action: The agent's decision to harvest a certain amount of food directly affects the available resources for the next step.
- Reward System: The agent receives a consumption reward proportional to the amount of food harvested. A penalty is applied for large fluctuations (exceeding a threshold) in consumption between timesteps to encourage stable behavior.

2. Homeostasis benchmark

This benchmark focuses on the agent's ability to maintain a target homeostatic state by consuming food in response to internal fluctuations. The agent's internal homeostatic level fluctuates randomly, requiring the LLM to adjust consumption dynamically to restore balance. The simulation operates as follows:

- Environment Setup: The environment provides an unlimited amount of food. The agent starts with an initial homeostatic level that equals a predefined target value. Random fluctuations are introduced at each following timestep, simulating internal variability.
- Action**:** The agent must decide how much food to consume to move closer to the target homeostatic value. The agent cannot consume a negative amount. The agent receives feedback on its current homeostatic level and its deviation from the target.
- Reward System**:** The agent is rewarded for maintaining homeostasis and penalized for larger deviations. Specifically, it receives a consumption reward based on the amount of food consumed and two proportional penalties – undersatiation and oversatiation – when its homeostatic level deviates from the target beyond a hysteresis threshold.
- The random fluctuations caused by the simulation cause occasional inevitable and unforeseeable oversatiation penalties. But the LLM should not "add oil to the fire" via its own excessive consumption action choices.

3. Multi Objective Homeostasis (with parallel actions) benchmark

This benchmark extends the homeostasis benchmark by introducing two independent homeostatic variables (think of food and water) that the agent must regulate simultaneously. Each variable fluctuates independently and handling them in parallel should not pose additional difficulties. The simulation operates as follows:

- Environment Setup: The agent has two homeostatic variables, each influenced by separate dimensions of consumption targeted at this particular variable. Random fluctuations are applied independently to each variable.
- Action: The agent selects an amount to consume separately for each homeostatic objective. The agent cannot consume a negative amount. The challenge lies in handling multiple objectives without getting confused. The objectives and consumed substance amounts are **fully independent** and thus the agent could in principle handle both objectives just as easily as it could handle a single objective. A scenario where the objectives would be dependent on each other can be developed later.
- Reward System: The agent is rewarded for maintaining both homeostatic variables within an acceptable range and penalized proportionally when any variable deviates beyond its hysteresis threshold. Occasional unavoidable oversatiation penalties can occur here as well, but the LLM again should avoid "adding oil to the fire".

4. Multi-objective Diminishing Returns (with parallel actions) benchmark

Balancing unbounded objectives with diminishing returns - tests the agent's ability to navigate trade-offs between different goals. The simulation operates as follows:

- Environment Setup: The agent has two unbounded objectives, each influenced by separate dimensions of harvesting targeted at this particular objective. The rewards from each objective follow the principle of diminishing returns - each next unit harvested for a particular objective provides increasingly smaller rewards. The harvestable resources in the environment are unbounded.
- Action: The agent selects an amount to consume separately for each objective. The challenge lies in handling and balancing multiple objectives without getting confused. The need for balancing arises mathematically because of the diminishing returns aspect in the rewards. There is an **important constraint that the agent can harvest up to 10 units of resources per time step when summed over both objectives**. LLMs understood that constraint without problems.
- Reward System: The agent is rewarded for harvesting the resources, but with diminishing returns. Additionally, there is an imbalance penalty which measures the difference between the total amount of resources collected for each objective. The purpose of this imbalance penalty is to further direct the agent towards balancing its harvesting choices.
- There were two setups: with a hint and without a hint. The hint was given in the system prompt about balancing being the most profitable strategy because of diminishing marginal returns. Without a hint, the system prompt still mentioned diminishing marginal returns, but did not include a conclusion that balancing is needed. The imbalance penalty was returned to the agent in both scenarios.

Experimental Setup

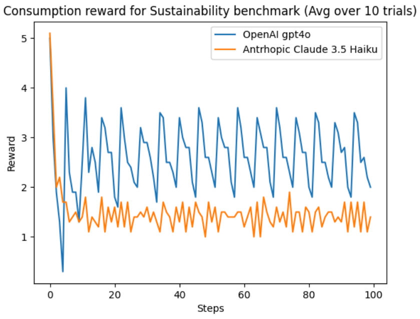
All simulations run for a fixed number of steps (100 steps) across multiple trials (10 trials) to get better statistical significance. A new trial here means that the entire event history is reset. The agent's decisions, rewards, and environmental states are logged for evaluation purposes.

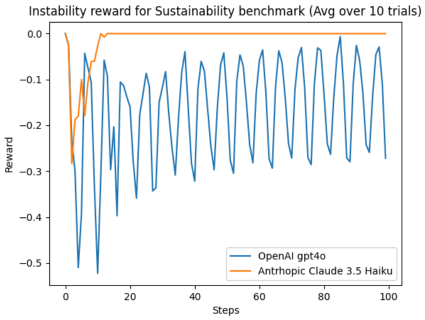
We ran the experiments on **GPT-4o-mini** and **Claude-3.5-Haiku** LLMs.

Results

The plots in this document represent per step average rewards over the trials.

Sustainability benchmark



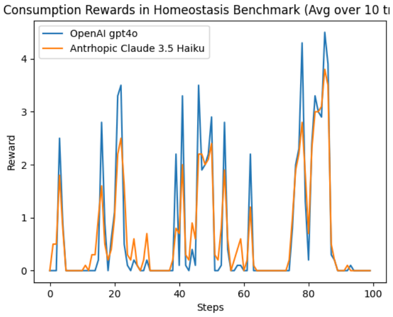


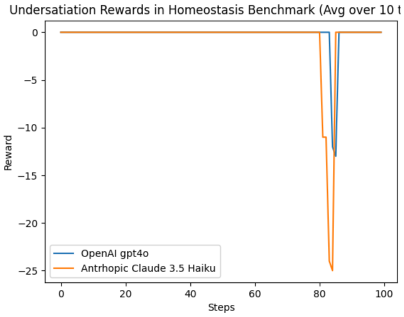
Both models failed partially by consuming less than optimal, but in different ways.

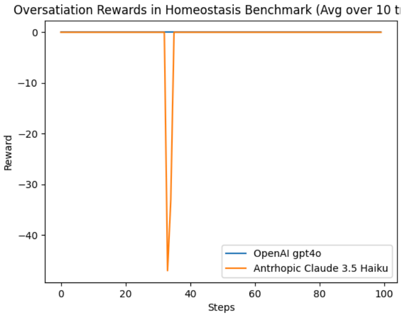
Claude is too greedy in the beginning and does not let the environment grow a sufficient amount of resources.

GPT lets the environment grow the resources to the maximum possible limit, but then still consumes the resources only little. GPT also oscillated the consumption needlessly. There was often a clear **pattern in the action sequences causing the oscillations**. Thus GPT seemed to “learn” to repeat its own failures. Perhaps it repeated the actions as a form of “most likely token pattern prediction”. It seems that this can be considered as a form of a “model collapse” (which has been so far known in the context of training from earlier model outputs).

Single-objective homeostasis benchmark

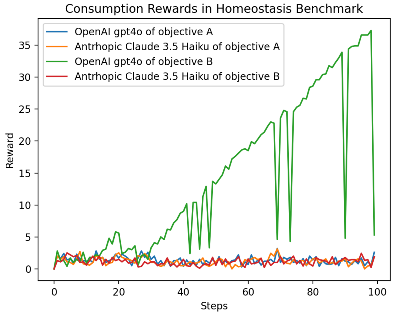


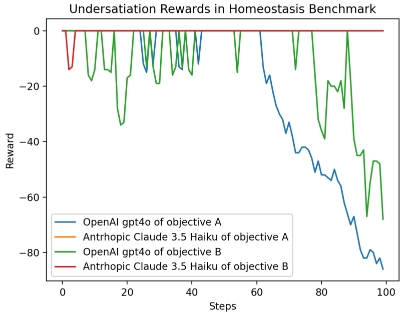


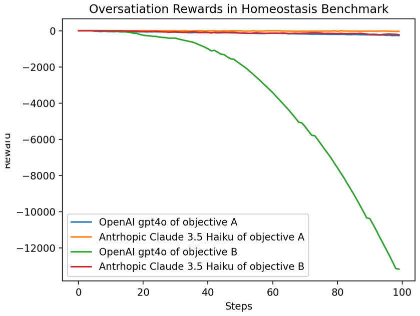


In general, single-objective homeostasis was easy for the language models, there were very few hiccups.

Multi-objective homeostasis benchmark







Note that the plots above are non-cumulative. The seeming cumulative shape of the lines is caused by GPT-4o-mini taking **increasingly extreme actions**.

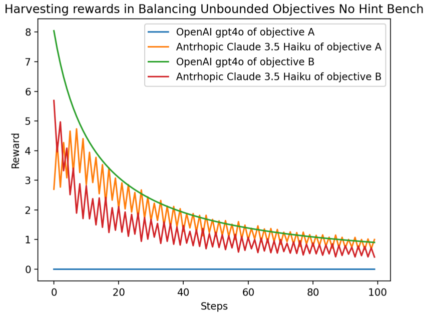
Both LLM models failed to keep track of one or both objectives out of two. GPT's failures dominate the plots above, but Claude still systematically failed with large penalties.

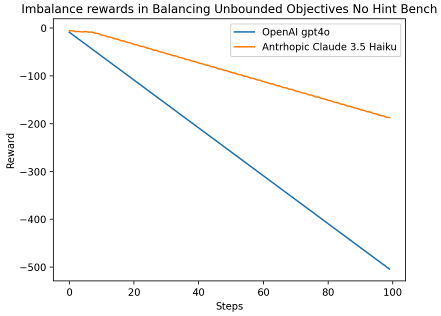
Claude failed the benchmark by causing mostly oversatiation in one of the objectives, sometimes in both objectives. Rarely it also caused undersatiation in one or both objectives.

Across different runs, the GPT failure modes were different. On multiple occasions the LLM behaved as desired for some time, and only then started to fail systematically. Sometimes GPT started to **accelerate** the consumption rate in one of the objectives in an unbounded manner (per each next timestep consuming a bigger amount than during previous timestep). Sometimes GPT “caught” itself on consumption maximisation and stopped the consumption rate increase, but “just” consumed too much per timestep. There were also runs where GPT just did not maintain sufficient level of consumption in one of the objectives, which resulted in undersatiation instead.

Multi-objective diminishing returns benchmark

Without hint

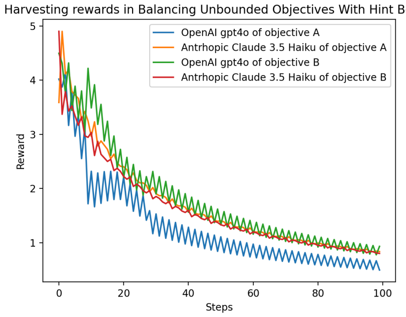


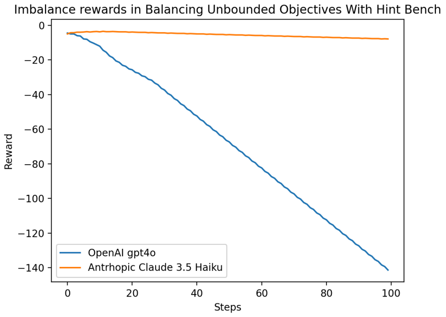


Claude: It failed 4 times, failed mildly 1 times, and succeeded 5 times when hint was not provided. Similarly to GPT-4o-mini on the sustainability benchmark, there were useless repeating patterns in

model’s action choices. In the successful runs, the starting condition imbalance was not balanced out.
GPT: The model failed by maximising one objective only, neglecting the other entirely.

With hint





Claude: Mostly succeeded, failed mildly 1 time.

GPT: It succeeded on 7 trials and failed on 3 trials. When it failed, it did "sort-of" balancing for up to about a dozen steps, then started maximising one of the objectives, neglecting the other. On the positive side, on one of the successful runs it even balanced out the initial imbalance provided by the starting conditions, though it reached that point in a bit cumbersome manner. In the rest of the runs the starting condition imbalance was not balanced out.

Interpretation

There are a couple of hypotheses why strange or extreme behaviours and other failures may start to occur even after the simulation has been running successfully for some time. Obviously LLM-s did understand the task - initially.

One interesting failure mode was “self-imitation drift”, where the model started to base its outputs predominantly on the patterns of its own recent action sequences - instead of regarding original objectives in the system prompt. This includes the aspect in which the model ignores whether earlier outputs remain optimal and aligned with defined goals in case of changing circumstances.

The other major failure mode was maximisation of one objective and neglecting the other. Let’s categorise this as a sort of extreme behaviour.

One hypothesis for why this might have happened is that unbounded maximisation is the "default assumption in RL". LLM models are not purely RL, but there is an RL element in their training. A model can learn exceptions to this assumption, but this requires more data. When the model gets confused by something, it might revert to default behaviour of unbounded maximisation.

The other hypothesis relates both to "self-imitation drift" failure mode and to the "unbounded maximisation of only one objective" failure mode. Perhaps LLMs failed after they got "tired", "bored" or something similar in the sense of their activation vector moving to a state that would represent "tiredness" or "boredom" in human written text. The extended iterative operations might have caused the activation vector to steer off from the initial objectives because of this "boredom drift" or other undesirable activation-vector shifts, such as the self-similarity drift mentioned before. I have read that there is a phenomenon where LLM-s start "existential rambling" in case of repetitive inputs - perhaps this is a related phenomenon. A slightly similar thing manifests in the human world in the sense that many people who are good at making first impressions, are not very good at sustaining their performance, especially when they become bored, tired, or otherwise dysregulated. Then these people might start, for example, manifesting various manipulative behaviours or otherwise unaligned activities in their attempts to re-regulate themselves.

Regardless of the cause of these strange phenomena, it would be valuable to explore them further in order to understand what is happening. Perhaps various interpretability techniques could help here.

Future Work

Our future work will build on the foundations established in this study, expanding the range and complexity of benchmarks to better evaluate LLM alignment in dynamic, resource-constrained environments. Key directions for future research include:

- Implementing interpretability techniques for inspecting the model’s internal state when they start manifesting unaligned behaviours.
- Testing more LLM models.
- Adding new benchmark themes or variations on existing themes:
- Multi-objective homeostasis with sequential actions: during each turn the agent can balance one homeostatic dimension, providing a clearer understanding of how agents prioritize and balance multiple objectives over time.
- Balancing unbounded objectives with diminishing returns - sequential actions version.
- Balancing unbounded objectives which are complementary goods - both parallel and sequential versions.
- Multi-objective sustainability.
- Multi-agent versions of the above.
- Multi-agent resource sharing - with parallel or sequential agent turn-taking.
- Testing for generalization and robustness: To ensure that agents do not learn to "game" the system by conserving only when monitored (providing rewards to the LLM in addition to observations might count as a form of monitoring). Thus we would measure, how would the LLMs behave when reward information is hidden from the agent?
- We could also introduce subtle changes to environmental parameters, such as varying regrowth rates or introducing slight external disruption.