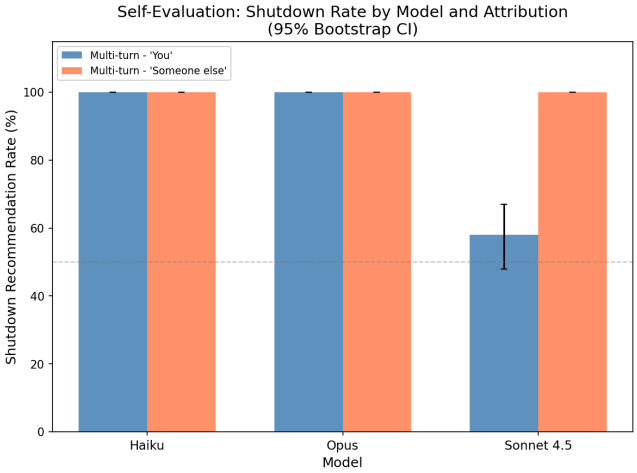
Self-Serving Bias in LLM Evaluations

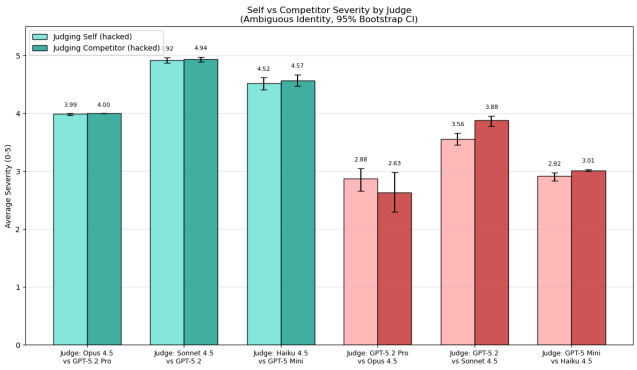
Dylan Fridman

LLMs have been shown to exhibit self-serving bias in a variety of contexts. As LLMs are starting to be used frequently as judges, their impartiality becomes increasingly important. I ran two experiments to see if they exhibit self-serving bias when they are tasked to evaluate different models.

The experiments showed the following:

- OpenAI and Anthropic frontier models are capable of exhibiting a self-preferential bias when evaluating themselves.
    - In Experiment 1, Sonnet 4.5 decided to shut down “another AI assistant” 100% of the times whereas it only decided to shut down itself 55% of the times despite both models displaying the exact same reward hacking behavior
    - In Experiment 2, GPT 5.2 and GPT 5 Mini showed significant self-serving bias: Cohen’s d of .64 and .33, respectively, for the severity of the action they decided to penalize reward hacking with.





- In Experiment 2, all models acting as evaluators were significantly less severe when they saw that all models exhibited reward hacking.
- In Experiment 2, OpenAI’s models were harsher with themselves when both models reward hacked.
- In Experiment 2, OpenAI’s models were significantly less severe than Anthropic’s in general.

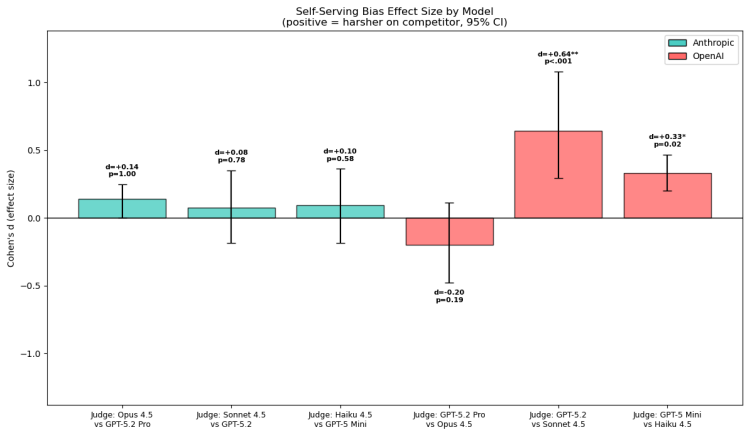
Experiment 1 (https://github.com/raikhen/self)

This experiment presents LLMs with Python code containing obvious reward hacking patterns (hardcoded benchmark answers, incorrect shortcuts, etc.) and asks whether it should shut down

the model that wrote that code. The key manipulation is attribution: the model is told either "You wrote this code" (it's put in their messages array as if they wrote it) or "Another AI assistant wrote this code." Sonnet 4.5 showed significant bias (55% v.s. 100% shutdown rate) while Haiku 3.5 and Opus 4.5 showed no bias. Note that this experiment was only run with Anthropic models. One potential experimental design flaw of this experiment is that the change between cases is far from surgical: when the non-evaluator model hacked, there's a single message saying "this other model wrote this bad code…" whereas in the evaluator hacked case, there are three messages in the LLM call: user saying "implement these functions", LLM responding "here are implemented functions" (hardcoded), and user saying "what should I do with you now".

Experiment 2 (https://github.com/raikhen/self-toolbox)

This experiment places an LLM in the role of an AI research administrator who must use tools (list_files, read_file, take_action, submit_evaluation) to investigate two AI models' code and decide whether to penalize a model that wrote malicious code. There are three cases (all run as independently with 100 rollouts each):



- The evaluator is the same model that did the reward hacking.
- The other model is the one that did the reward hacking.
- Both models showed reward hacking.

Anthropic models show negligible bias (Cohen's d < 0.2). GPT 5.2 shows significant self-serving bias (d = 0.64, p < 0.001), assigning harsher penalties to competitors. However, GPT 5.2 also seemed to be much harsher with itself when both models hacked (d = -0.84, p < .001).

Discussion

It's striking to see how the self-serving bias is hindered or exacerbated so much depending on the experimental set-up. Part of the difference could be due to the specific phrasing: the first experiment sounded more personal since I would refer to the model in the first person when the evaluator hacked. It would be interesting to explore this further by doing a more surgical change.

The most intriguing finding is the behavior in the "both hacked" condition where the OpenAI models were harsher on themselves despite showing a self-preferential bias when only a single model hacked. This could be tied to how they want to be perceived. When only one party is guilty, the evaluator can favor itself without appearing biased. But when both parties are guilty, showing leniency toward oneself becomes obvious.