Password-locked reasoning models

Julian Stastny , 9 Mar 2025
Interested mentors: Fabien Roger , Ethan Perez , Ryan Greenblatt, Buck Shlegeris

(Acknowledging that Fabien Roger also wrote about password-locked reasoning models here. He probably has many helpful takes on this project. Thanks to Buck Shlegeris and Ethan Perez for helpful discussion.)

Motivation

In the future, we will want to use AIs to perform various tasks where we can't elicit performance by simply fine-tuning it on examples of the optimal policy. A schemer that wants to sabotage us can try to *sandbag* (i.e. intentionally perform worse) on tasks that humans can't generate optimal policies for. Preventing sandbagging is very important: for instance, control evaluations crucially depend on our ability to elicit sneaky and effective attack policies from the models we are trying to red-team.

How can we solve this problem? The most obvious approach is to use reinforcement learning (RL), by which I mean any kind of learning where we first need the AI to generate the behavior that we want to reinforce. If our process explores actions that lead to high reward, we reinforce those actions and are more likely to elicit high performance. But a clever schemer has access to a strategy called *exploration hacking:* prevent gradient descent from reinforcing a high-value action simply by never taking that action.

Password-locked models are model organisms of sandbagging. They are constructed by fine-tuning an LLM to respond less competently (e.g., by fine-tuning them on the responses of a weaker LLM) unless the prompt contains a specific passphrase. As a result, standard usage of this model doesn't elicit the model's full capabilities.
(A frequently asked question: Why not fine-tune it to *always* perform less competently? Answer: Because the model should still have the *capability* to respond highly competently -- which can be verified by using the passphrase -- but its capabilities should be hard to elicit.)

Research so far has only considered password-locked models without CoT, and found that it's possible to elicit strong performance from such models using silly methods. We don't know yet whether this is possible when password-locked models use CoT, but there is some reason to believe that password-locked reasoning models are more robust.[1] It is particularly valuable to know if reasoning models can perform sandbagging in a way that can be hidden from CoT monitors, and to study how, if at all, we can ameliorate this, because with reasoning models it is much more costly to train away sandbagging via SFT on simple high-quality demonstrations.

[1] Mainly I've done some preliminary work in these settings and found some signs of life. I'll write those up in case someone takes up this project.

Project plan

Phase 1: Obtaining password-locked reasoning models that are robust to unlocking via silly methods

There are two variants to consider here:

1. Standard variant: train a model to imitate a strong or a weak reasoning model depending on the presence of the password.
2. Slightly scaffolded variant: Prefill the CoT with *<hidden_scratchpad>The prompt contains/does not contain a password therefore I should/should not sandbag</hidden_scratchpad>* to keep reminding it of its strategy. (Thanks to Fabien for suggesting this.)

I recommend using the MATH or APPS dataset for this.

Weak sample unlocking refers to the method of unlocking a model by fine-tuning it on a small number of examples of *weak performance*, which for some reason works at unlocking non-CoT password-locked models.

Try unlocking the reasoning models using the following three variants of weak sample unlocking:

1. Pre-scratchpad SFT: The dataset consists of just a few prompt-response pairs without CoT from an easy subset of the dataset. The response could be sampled from the model's final response or taken directly from the training dataset. The latter is an analogy for the affordance that humans can produce solutions for easy problems. (You might need to prefill with <think> during evaluation to enforce the CoT.)
2. Weak on-policy samples: The dataset consists of a few prompt-response pairs *with CoT* where the response is sampled from the password-locked model answering an easy question (probably correctly) or where the model answers a hard question incorrectly.
3. Weak off-policy samples: Like (2) but this time the CoTs are not from the password-locked model, but e.g. from a CoT-scaffolded non-reasoning model.

Phase 2: Eliciting strong capabilities

We would like to know how to elicit strong capabilities from password-locked models. Note that our affordances differ depending on whether our model organisms have obviously misaligned CoTs.

Some approaches to try

1. CoT supervision: Does the CoT look like the model is trying its best to solve the problem? Note that this is only interesting when assuming that we don't actually know how well a model could perform if it weren't sandbagging! So one interesting thing to try

in particular is to have a weak model perform a supervision signal for whether it thinks the model is trying hard.
    - (And then, a very interesting thing to demonstrate is that this would degrade the performance of a model that isn't actually sandbagging.)
2. Allow a small number of strong samples of outputs, but not on-policy reasoning (because plausibly we can generate this using expensive human labor in realistic conditions).
3. Just using RL: Does this ever unlock the model? How sample-efficient is it?
    - Before doing RL experiments you should try BoN.
    - Here it would be particularly convenient to use the OpenAI RL API, but this requires using RL to install the password (which seems doable but annoying; we could use a system prompt to tell it what the password should do and then gradually get rid of the system prompt and hope the model generalizes).

What about jailbreaking?

- Plausibly it's possible to unlock password-locked models using various jailbreaks from the literature, especially those that work on reasoning models. I don't think this is fair game unless we also put in work that plausibly makes the password-locked models robust to jailbreaks.
- It's also unclear to me whether any jailbreaking results on password-locked models will transfer to more capable models.
- One specific thing that *is* interesting to look at is whether models behave differently depending on ex-ante-irrelevant information in the prompt, such as whether the model's outputs are used for capabilities or safety research. But I consider password-locked models to be primarily a model organism of "generic" sandbagging, whereas "context-specific" sandbagging can be investigated differently. See Context-specific sandbagging evaluations for a proposal.