Testing and Improving the Generalisation of Probe-Based Monitors

The Summary

Simple classifiers called "probes" can monitor AI models for concerning behaviors, but may fail when tested on new data or adversarial inputs. This project will investigate when probes fail and develop methods to make them more robust. Additionally, we will extend our investigation to under-explored settings, like those in which the AI attempts to circumvent its own monitoring and/ or where the probe is being used as part of LLM training.

This work is ideal for participants interested in hands-on AI safety research with practical applications for AI control systems. The project requires minimal background knowledge while addressing crucial questions about monitoring system reliability that could directly influence how AI companies deploy safety measures.

The Non-Summary

Background

Redwood-style AI control is a research direction studying how we can safely use AI models even if they are misaligned – for example, by carefully monitoring their proposed actions. Furthermore, researchers have found that extremely simple classifiers can actually be used to cheaply and effectively monitor the 'thoughts' of models for a variety behaviours:

1. Deception (Detecting Strategic Deception Using Linear Probes)
2. Truthfulness (How well do truth probes generalise?)
3. Sandbagging (White Box Control at UK AISI - Update on Sandbagging Investigations)
4. Hazardous information (Coup probes: Catching catastrophes with probes trained off-policy)
5. High-stakes situations (Detecting High-Stakes Interactions with Activation Probes)
6. Backdoors (MacDiarmid et al., Simple probes can catch sleeper agents)
7. Jailbreaks (Sharma et al., Constitutional Classifiers: Defending against Universal Jailbreaks across Thousands of Hours of Red Teaming)

These so-called ‘probes’ are applied to the activations of models rather than just their textual outputs. They could be useful for many safety frameworks, not just AI control (e.g. 18 Applications For Deception Probes), and OpenPhil is keen to fund research in this area.

Problem Setting

Since activations are not very interpretable and differ for each model, probes are currently constructed via training on a labelled dataset of positive and negative examples. This comes with the usual baggage you expect from regular ML, as well as some difficulties to do with AI alignment:

- Probes may be too simple to learn the true pattern (e.g. if the model has multiple linear representation of the same concept, or if they are represented non-linearly,)
- Probes may be able to learn the true pattern but instead learn brittle heuristics (e.g. all the positive examples had a formal tone and no mention of Bulgaria)
- Probes may learn a pattern that works well in-distribution but not out-of-distribution (e.g. works for english but not chinese, or any adversarial edge case)
- Probes may learn a pattern that works well out-of-distribution but tracks something other than what we intended (e.g. false beliefs instead of intentional lies)
- Probes may track what we intended currently but do not continue to do so over time (e.g. the model being probed is undergoing training or is changing representations just based on in-context learning and memory)
- Probes may track what we intended over normal model changes but not adversarial ones (e.g. malicious actors train the model to do monitored activities without setting off the monitors, or the model itself is misaligned and influencing its activations directly or through training to avoid the monitors)
- etc.

That said, as mentioned, probes appear to work quite well in practice. Though they have been around for a long time, their use as alignment/ control monitors is still quite new and thus there should still be many low-hanging fruits relating to improving their generalisation abilities. The biggest flaw in probe-related safety mechanisms may turn out to be companies' unwillingness to use them, either due to high costs or lack of incentive.

Possible Research Questions

There are several possible directions to explore. This makes parallelising work easy but it might also mean that we will narrow the scope of the project, to the direction that seems most promising, as the project continues - going from an ‘exploration’ stage to an ‘exploitation’ stage.

- RQ1: When do probes fail to generalise? Can {invariant risk minimisation/ ensemble methods/ non-linear methods/ data mixing} improve probe generalisation? Can we increase the space of concepts a single probe can check for at once (i.e. many forms of misalignment not just one form of deception)?

    - Possibly establish tools and best practices for probe developers (e.g. a tool that checks if you can get 100% accuracy on training data with just simple heuristics)
- RQ2: How could someone intentionally break probe monitors? Can models break their own monitors if instructed to do so? How much instruction is needed?
- RQ3: Do probes trained on early checkpoints of LLMs generalise to LLM checkpoints from later in training? How should the probe be updated over the course of training? Does this change if the probe is used as part of the loss function?

Project Plan

Compute:

- If we have no GPU access, we could likely only work on RQ1:
    - It would be easiest/ cheapest/ fastest to look at an existing generalisation failure and then try to improve probe scores in that setting
    - This is because some code + datasets of inputs-outputs + datasets of activations already exist (as with my old project) and training simple probes doesn't require a GPU
- If we have some GPU access (e.g. PC, free google colab, my university server) then we can work on most RQs for smaller LLMs, and look at multilayer probes:
    - We can generate new activation datasets for evaluating probes, as is needed when studying new settings or testing adversarial inputs or want activations from different model checkpoints
- If have proper GPU access (e.g. I can hire GPU access from vast.ai, as with other project) then can study larger LLM models and finetune smaller LLM models:
    - The latter might be required for ambitious versions of RQ3

MVP:

1. We make heavy use of my existing probe codebase and others, like that from Apollo Research.
    a. We may also use my discrete optimisation/ adversarial attack codebase (either white-box or black-box) and repurpose it for breaking probes.
2. We reproduce existing probe evaluation scores by following an existing methodology and/ or problem setting.
3. We then either:
    a. Show how the scores decrease significantly when we change the distribution of the evaluation data (e.g. different topic, different style, something else?).
    b. Show how we can improve scores significantly when we modify the training data (e.g. training order, perplexities, synthetic data) or the probe (e.g. invariant risk minimisation, ensembles of probes, architecture, loss function).
4. We repeat experiments for as many LLMs and behaviours as is convenient, based on resource availability and time left.
5. We spend 2-3 weeks writing up our results.

MVP++:
The ambitious plan is to set up experiments for research questions that are not answered by the MVP, such as those most relevant to future AI systems (e.g. settings in which the AI is trying to break the probe or setting in which we cannot wait for the AI to be fully trained before the probe is trained).
This may require us to come up with new methodologies that are not seen in other works, making it difficult to outlay an exact plan here.

Failure modes:

- We find new work convincingly showing that probes are fundamentally flawed and no longer a promising direction for safety.
    - We can try to reproduce their work or extend it to similar methods like black-box monitors or CoT monitors if they have not already.
- We find existing work that is very similar to this proposal, with little room for extension.
    - This is unlikely, simply because there is so much to investigate, but in this case we could simply reach out to other researchers for potential collaboration.
    - Alternatively, we can pivot to a 'probe application' work rather than a 'probe robustness investigation' work.
- In general, if our results are not positive (e.g. probes don't make for robust monitors) then this is still important to get out there so that safety researchers are able to adapt their safety plans.

Output

Most of the time will be spent on experiments, not writing, but the intended output of this project is a short academic paper that we could submit to a relevant workshop. It is no problem if we get rejected as we will simply put it on Arxiiv. If there are not enough findings for a paper, we will do a LessWrong post (and we would likely want to do this anyway for the sake of spreading our findings to the right people).

Risks and Downsides (externalities)

There are not many notable risks.

- We must maintain a high level of rigour when gathering our findings so as to avoid a situation in which developers overly rely on probes for AI safety plans, if they are not in fact that trustworthy.
- Broader adoption of probe monitors may lead to developers training their models to pass these monitors, which may result in models that are less monitorable rather than more safe.
- We may discover ways to break monitors that malicious actors then replicate, without corresponding defences.

    - Our team will evaluate if this is the case and modify what is published accordingly.

Acknowledgements

The majority of the background section was written by my ex-supervisor Dmitrii Krasheninnikov.

Team

Team size

- Aiming for a normal team size (3-5 people, including the project lead), but if there are many applicants then the project lead might choose to reduce the amount of research that they do and instead work on co-ordinating a larger team.
- Every team member is expected to spend a minimum of 10 hours per week on this project during its official duration, and any more is even better but really we just want to be on track to producing the project output.

Project Lead

Adrians Skapars (Adr.Skapars@gmail.com, www.linkedin.com/in/adrians-skapars-107458239)

- I am a 3rd year PhD at The University of Manchester, studying discrete optimization methods for inputs to LLMs, so I have a relevant background in white-box LLM research. (Note that I am in the middle of pivoting towards an interpretability-related research direction.)
- Prior to the start of AISC, I will have submitted a paper on "The impact of off-policy training data on on-policy probe performance", which is a similar project to the one being proposed here but instead focuses on a specific kind of generalisation failure. This group project was done as part of a 3-month full-time LASR Labs fellowship, supervised by Dmitrii Krasheninnikov.
- I was also a participant in a previous AISC project, which successfully produced a workshop paper on "finding world models in maze-solving transformers", which gives me confidence that we can do the same.
- I commit to spending at least 10 hours a week on this project, but that number may increase if there is enough overlap with my PhD research direction by the time AISC starts.

Roles

- There are no strict roles for this project and I support some members of the team focusing on conceptual work whilst others focus on execution. That said, it is desirable for participants to be able to conceive, implement and analyse experiments by themselves.
- Team leads are desirable. Your job would be to help me with checking in on everyone to make sure they are doing useful work that is exciting to them. Additionally, with

enough participants, team leads would basically be managing their own sub-team and taking responsibility over one of the possible project streams.

Skill requirements

- **Probe work is very easy and requires almost no background knowledge!** If you know how transformers work and what ‘activations’ (and ‘binary classifiers’) are, then you are most of the way there.
- Participants should have proficiency with Python (though, even the more complicated bits of code can probably be handled by ChatGPT by now). Previous experience with PyTorch and libraries like TransformerLens is valuable for debugging and fast progress, but not mandatory.
- Participants should understand the problem of generalisation (i.e. training models on one data distribution but then testing them on out-of-distribution examples). Knowing some theory or practical heuristics on what makes models generalise is beneficial, but not mandatory.
- Participants interested in conceptual work should have (or be interested in developing) strong models of what makes very capable AI scary and hard to align or control.