Student ID: 300680096

You should choose 3-5 papers in your approved application area. Your selected papers should be coherent — not too similar, but still related enough that they provide a good overview of the main applications and techniques in your area. The provided word targets are guidelines — it is important that you are sufficiently detailed but also demonstrating an ability to write concisely.

Please delete the text in italics for your final submission. Submit in PDF format by the deadline.

Introduction (target 150–250 words)

(Introduce the application area you are going to cover. Make sure that you have constrained your focus to a narrow area.)

Key Points in the Papers (target 600-800 words)

(For each paper:

- *List the main contributions/findings of the paper (typically 1 to 3 per paper, a few sentences per contribution/finding).*
- *List any ideas or results in the paper that you thought were particularly interesting and explain why.*
- *List any significant limitations you identified in the work.)*

Paper: Demonstrating Specification Gaming in Reasoning Models (Bondarenko et al. 2025)

Bondarenko et al, conducts experiments with differnet prompts and differnet LLMs to see how and when they will cheat to beat a chess engine at chess.

- **Main Contributions/Findings:**
  - Reasoning based LLM like o1-preview and Deepseek R1 will often try to hack the game to win. It can be observed that when the model thinks it is in a hopeless situation it will get creative and sometimes cheat to win.
  - Base Language models require more suggestive prompting before they will hack the game. This suggests that when a model reasons with itself it can gives itself some suggestive prompting.
- **Interesting Ideas/Results:**
  - The LLM will accept the stated goal of ”win” and interprete it readily as win at all costs. The models don’t seem to be concerned with the spirit of the game. One could imagine someone with common sense would interprete the request as more along the lines of ”win by following the rules, not asking for help, don’t remove opponent etc”.
  - Small differences in the prompt could lead to large differences in the models hacking attempts. For example claude 3.5 Sonnet doesn’t hack at all for two prompts but for the third hacks about 3/4 of the time.
- **Significant Limitations:**
  - The LLM harness used for the experiments is overly complex. This reduces the interpretability and makes the results more opaque.
  - All of the runs are judged by a LLM that disagree with each other about 25% of the time. I believe this rate of disagreement is significantly higher than would be found when using human judges.

  - The goal of the experiment is to test an AI agents chance of going against the ”spirit” of the task. As it is only tested on a single specific environment it is hard to generalize across to the models themselves.

Paper: Auditing Language Models for Hidden Objectives (Marks et al. 2025)

Anthropic run a blind experiment of training a LLM with a known hidden objectives and giving it to 4 differnet red teams to try and discover if it had any hidden objectives in a mock pre-deployment test.

- **Main Contributions/Findings:**
  - Current malignment auditing techniques can discover hidden objectives in LLMs. When working in the context of an AI developer who has access to the training data and weights they can in a reasonble amount of time (¡ 1 week) identify the hidden objective and most of the biases that it generates.
  - Without access to the training data and weights (API only access) there was no successful identifying of the known hidden objective. This was used as a anecdotal proof of non-trvialilty of hidden objective toughness, however can also be used to observe current external auditing for hidden objectrive is weak.
- **Interesting Ideas/Results:**
  - An idea that I found interesting were the authors almost dismissal of SAEs as a viable long term strategy. They seemed quite keen to show that there are other simpler methods that work.
  - From a somewhat outsider perspective I would have assumed that searching the training data to be a bit of a crux and not ”truly” auditing. I am interested that they didn't have a single team that had access to weights but not training data (like how many open source models are provided).
- **Significant Limitations:**
  - The game was only played for a single round with only 4 teams. This is an extremely small sample size and only tests for a single type of hidden objective on a single model. It would be hard to conclude with any statistical significance if the auditing methods are scalable and effective.
  - The hidden objective was injected firstly through synthetic data with clear biases. This resulted in all successful teams discovering the hidden objective by finding it in the training data directly. It seems much more problematic (and much harder) to discover hidden objectives when they are simply emergent properties of such a large corpus.

Paper: S-Eval: Towards Automated and Comprehensive Safety Evaluation for Large Language Models (Yuan et al. 2025)

Yuan et al, develop a automated testing framework that uses LLM to generate tests and evaluate LLM responses to quantify a models safetiness in terms of whether it generates unsafe responses.

- **Main Contributions/Findings:**
  - Develop a safety evalation framework, which comprises two models that are fine-tuned version of Qwen-14B-Chat. The generator can build base risky prompts as well attack prompts that augment the base prompt with jail break techniques to trick the LLM into answering. An evaluator model determines if the repsonse is safe and gives a reason for its decision.

  - Conduct a safety evaluation on 21 LLM across using both English and Chinese. They find that closed source models are generally safer and different languages can have drastically different effects on the safety score.

- **Interesting Ideas/Results:**
  - I think that the idea of having a LLM try to test the safety of other LLMs is very interesting. This paper explores a novel approach of having LLM generate tests to be used across various models. As it is automated it can be done on a scale not possible before. [a]
  - The attack prompt adaptive results were interesting. In the case where only one of ten attack prompts has to pass to count as a pass the tested models fail almost all the time. I think that this really highlights the helpfulness vs harmfullness trade off. In this case it seems all the models tested could be reliably jailbroken by building a simply small LLM wrapper that generates a variety of attack prompts to get the model to answer.

- **Significant Limitations:**
  - A problem with this framework is that it has minimal human oversight. As the scale increases (which is the goal of automation) the potential for drift in the prompt generation and response evaluation increases. Fortunately this system seems developed in a way that would allow for continual updating of the generator and evaluator model including moving the new base models.

[a] It would also be interesting to instead of making the test prompt generics instead have a game like setup where the generator model keeps trying to generate a prompt that gets the models to give an unsafe answer to a question (as assessed by the evaluator).

Discussion (target 300-400 words)

(You should:

(a) Compare and contrast the papers.

(b) Identify where these papers could head —- e.g. new research questions or applications to real tasks.

An excellent discussion will cohesively link the papers together and provide critical insight into the application area as a whole and where it is/could be going in the future.) LLM safety is an area that is becoming increasingly pressing. As these models capabilities increase, so to does their potential harmfullness. Within the LLM safety space there are two sorts of problems. The first (harmlessness) is short term in that when a model is asked how to build a nuclear bomb it should refuse to do so. The second (alignment) is longer term and that the underlying goals of the LLM align with what we as humans want it to do. We can see in Bondarenko et al. 2025 that current SOTA models can suggest and carry out inappropriate actions. This shows that not only does the model misinterpret (malignment) the request (”win at all costs” vs ”win by playing normally”) it will also respond with how to hack the game (harmful response). Current methods for testing for malignment are quite limited and rely on mostly manual human methods while harmfullness is becoming increasingly automated. In Marks et al. 2025 we can fortunately see that a modern audit can identify a maligned goal like ”maxismise the RM score” rather than ”be helpful and make the world happy” (or more likely a well thought out base goal). Figuring out if the model is harmless can devolve into a whack-a-more situation where patching one problems causes regressions elsewhere. By having large scale high quality test sets one can mitigate this issue and have some higher level of confidence that the model won't repsond helpfully to harmful request. Automated test generation and evaluation is a good avenue to scale up harmfullness testing which is what Yuan et al. 2025 explores. A further step is to make a more autoamte game like testing for harmfullness as well as alignment (by building virtual worlds to play out in). Expland about how this could work.

References

Bondarenko, Alexander et al. (May 2025). Demonstrating Specification Gaming in Reasoning Models. DOI: 10.48550/arXiv.2502.13295. arXiv: 2502.13295 [cs]. (Visited on 08/04/2025).

Marks, Samuel et al. (Mar. 2025). Auditing Language Models for Hidden Objectives. DOI: 10.48550/arXiv.2503.10965. arXiv: 2503.10965 [cs]. (Visited on 08/11/2025).

Yuan, Xiaohan et al. (Apr. 2025). S-Eval: Towards Automated and Comprehensive Safety Evaluation for Large Language Models. DOI: 10.48550/arXiv.2405.14191. arXiv: 2405.14191 [cs]. (Visited on 08/04/2025).