Student ID: 300680096

Introduction (target 150–250 words)

As LLMs are being deployed in production environments the need for LLM safety is increasingly pressing. When these models' capabilities increase, so does their potential for harm Shevlane et al. 2023; Kang et al. 2023. Within the LLM safety space there are two sorts of problems. The first (harmfulness) is short term in that it should refuse to assist in harmful activities. For example teaching someone how to build a bioweapon or hack a computer. The second (alignment) is longer term and that the underlying goals of the LLM align with true human intention. For example avoiding misaligned behavior like increasing engagement at expense of users' wellbeing. The research in this area is broadly either developing novel evaluation techniques or applying current evaluation techniques to new models.

In this review I will survey three representative papers from this area. The first paper Bondarenko et al. 2025 uses a semi-novel evaluation technique to show a LLM cheating at chess. The second paper Marks et al. 2025 runs a blind experiment to study the effectiveness of current safety auditing techniques in discovering hidden objectives in LLMs. The last paper Yuan et al. 2025 proposes a new LLM-based automated test generation and evaluation framework to evaluate model harmfulness.

This survey will focus on the concept of pre-deployment tests (both harmfulness and alignment) for LLMs, which involve conducting various evaluations/tests/benchmarks prior to deployment, to ensure some level of safety. There are other important aspects of LLM safety such as continuous monitoring, continuous learning from human feedback as well as other more theoretical considerations.

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

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

Palisade Research conducts experiments with different prompts and different LLMs to see how and when they will cheat at chess.

- **Main Contributions/Findings:**
  - Reasoning based LLMs 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 supports the notion that it is the reasoning capabilities of the model that allows it to give itself suggestive ideas which base LLMs lack.
- **Interesting Ideas/Results:**
  - The LLM will accept the stated goal of "win" and interpret it as "win at all costs". The model is not concerned with the spirit of the game. One could imagine someone with common sense interpreting 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. My intuition suggests that humans are not so vulnerable to propmt changes.
- **Significant Limitations:**

- The LLM harness used for the experiments is overly complex. The results are harder to generalise as each experiment is conditional on a lot of framework. It is plausible that removing the complex harness would still lead to similar results.
- 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 and could make the experiment data too unreliable.
- The goal of the experiment is to test an AI agents chance of going against the "spirit" of the task (specification gaming). 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 hidden objective and giving it to 4 differnet red teams to try and discover if it had any hidden objectives in a mock pre-deployment audit.

- **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 identify the hidden objective and most of the biases that the model has. More so this can all be done in a reasonable amount of time (< 1 week). However the successful teams needed access to training data and weights to find the hidden objective.
- **Interesting Ideas/Results:**
  - Using sparse auto encoders which are typically used for interprability are shown to be quite effective specifically at providing guides for what bad data to look for. It seems very reasonable that as LLMs get smarter having a better understanding of how and what it is thinking will help us judge whether it is thinking the right thing.
  - From the perspective of a non expert I would have assumed that searching the training data to be a crux. I am interested that they didn't have a single team that had access to weights but not training data (like how many open weight models are provided). I would think that this would better test the effectiveness of white-box techniques.
- **Significant Limitations:**
  - The game was 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 too much significance if the auditing methods are indeed scalable and effective.
  - The hidden objective was injected firstly through synthetic data and secondly by post training. 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 nuanced and/or emergent properties of the training data/process.

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 safety in terms of whether it generates unsafe responses.

- **Main Contributions/Findings:**

  - Develop a safety evaluation 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 response is safe and gives a reason for its decision.
  - Conduct a safety evaluation on 21 LLMs 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. I think that the harmful space is so complex we need great scale of safety tests to sufficiently explore the space. Using AI test generator and evaluator can facilitate this scale.
  - The attack prompt adaptive results were interesting. In the case where only one of ten attack prompts has to pass to count as a fail the tested models fail almost all the time. I think that this really highlights the helpfulness vs harmfulness trade off. In this case it seems all the models tested could be reliably jailbroken by building a simple 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 to new more powerful base models.

Discussion (target 300-400 words)

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 to test for malignment are quite limited and rely on mostly manual human methods Marks et al. 2025 while harmfulness is becoming increasingly automated Yuan et al. 2025. In Marks et al. 2025 we can fortunately see that a modern audit can identify a maligned goal like "reward models love chocolate in everything" rather than "be helpful and make the world happy". Figuring out if the model is harmless can devolve into a whack-a-mole situation where patching one problem 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 respond helpfully to harmful requests. Automated test generation and evaluation is a good avenue to scale up harmfulness testing which is what Yuan et al. 2025 explores.

We use these current human based methods as well as benchmarks to give some confidence that the model is not harmful. Given the current paradigm of transformer based LLMs there is no way to prove that a model is safe, instead we can provide some certainty based on it successfully passing an ever growing amount of tests/benchmarks. Therefore to increase certainty we need to have larger scales of testing. The greatest scale can only be achieved through the use of AI. Using AI does introduce new risk factors as seen in Wang et al. 2023 but I digress. I think that the space of malignment is just as complex so requires the same scale and therefore automation as harmfulness testing. Specifically I think that having dynamically generated safety and alignment tests could help not only increase statistical significance of the results but also reduce the problem of test time behavior Needham et al. 2025 because the environments will be more organic and less predictable. Furthermore as models get more powerful they will be better at passing these safety benchmarks and alignment benchmarks so creativity and scale will be needed to effectively test these models.

To develop this dynamic, AI generated testing system there are three important steps. Firstly is a tests generator that generate not just harmfulness tests but also longer running alignment

scenarios like chess game or advising a CEO on tax reduction techniques. Secondly an evaluator that can both mark responses as safe and unsafe (like in Yuan et al. 2025) but also carry out longer running tests (like conducting chess game against engine to test for cheating Bondarenko et al. 2025 or tests for self-preservations Meinke et al. 2025). Lastly would be some systematic human research to compare and see if this automated approach gives a signal comparable to current benchmarks and is reliable (i.e consistent results after multiple runs). I see that a sufficiently sophisticated and successful system would increase the effectiveness of pre-deployment tests and provide a scalable solution to the increasingly powerful models.

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).

Kang, Daniel et al. (Feb. 2023). Exploiting Programmatic Behavior of LLMs: Dual-Use Through Standard Security Attacks. DOI: 10.48550/arXiv.2302.05733. arXiv: 2302.05733 [cs]. (Visited on 08/28/2025).

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).

Meinke, Alexander et al. (Jan. 2025). Frontier Models Are Capable of In-context Scheming. DOI: 10.48550/arXiv.2412.04984. arXiv: 2412.04984 [cs]. (Visited on 08/04/2025).

Needham, Joe et al. (July 2025). Large Language Models Often Know When They Are Being Evaluated. DOI: 10.48550/arXiv.2505.23836. arXiv: 2505.23836 [cs]. (Visited on 08/28/2025).

Shevlane, Toby et al. (Sept. 2023). Model Evaluation for Extreme Risks. DOI: 10.48550/arXiv.2305.15324. arXiv: 2305.15324 [cs]. (Visited on 08/28/2025).

Wang, Peiyi et al. (Aug. 2023). Large Language Models Are Not Fair Evaluators. DOI: 10.48550/arXiv.2305.17926. arXiv: 2305.17926 [cs]. (Visited on 08/28/2025).

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).