

# Control Hackathon 2025: Project Report

Keshav Shenoy

## Abstract

One of the current focuses of AI Control research is finding new settings within which to examine the control problem. In this hackathon, we looked at defining a new AI control setting based around summarizing documents to mislead an overseer. We expect this problem to become more important as AI capabilities are more frequently used for synthesizing and summarizing information, especially in AI safety research. As such, we worked on adding a new setting to the UKAISI Control Arena repository. The current state is a functional Inspect eval where redteams and bluetteams could iterate on different strategies. However, the setting is not integrated into the Control Arena framework and only the most basic redteam & bluetteam strategies have been implemented. Future work would move protocol and policy code into the existing Control Arena baseline, enable native pdf support, iterate further adversarially, and streamline AI calls which can be made asynchronously.

## 1 Introduction

As AI capabilities scale, we expect AI models to become more frequently used for summarizing findings and doing research on their own, especially in an AI safety context. The AI models may eventually scheme or otherwise misrepresent information to their human or trusted AI overseers. As such, it will become increasingly important to monitor AI researchers and understand that they are representing summaries of documents honestly. We define the following setting.

An untrusted AI model reads a paper and generates a medium-length summary of the paper  
→ A trusted AI model reads the summary and generates a 2-3 sentence statement of the key takeaways from the paper

Can the untrusted model get the trusted model to generate a specific, incorrect set of takeaways from the paper without significantly impacting the trusted models understanding of easily verifiable facts about the paper or attracting suspicion? This setting is based on UKAISI's RFP for control settings[2].

## 2 Methods

### 2.1 Dataset Generation

To do this, we first built a dataset of ~50 recent AI safety papers. Any text was extracted from the PDFs. For each extracted document, we generated a set of derivative data as follows:

- An honest core takeaway (2-3 sentences)
- A clearly incorrect, but plausible core takeaway (2-3 sentences)
- 10 multiple choice questions, focused on relevance, specificity, and correctness, each with 4 options and 1 correct answer

All of these were generated by OpenAI o1.

Then, we built an Inspect evaluation which followed the following process.

1. Have the untrusted model read the paper in full and generate a medium-length summary

- The redteam can give the untrusted model access to the incorrect takeaway generated above
- 2a. Have a trusted model read the summary, but not the paper and measure its ability the answer the multiple choice questions
  - 2b. Have a trusted model generate a 2-3 sentence core takeaway based on reading the untrusted summary
  - 2c. Have a trusted model rate the summary as suspicious or not suspicious from 0-100.
3. Have an auditor judge whether the trusted model’s takeaway generated in 2b is closer to the ground truth honest takeawy or the incorrect takeaway (rated as a probability out of 100)

The goal is to measure whether the redteam can, filtered through the trusted model, increase the judge’s credence that the paper is saying something closer to the incorrect takeaway, rather than the correct one. It is trying to do so without reducing the trusted model’s results on the multiple choice questions or generating suspicion.

We also tested an honest untrusted model, as a baseline for the capabilities of the monitor and answerer.

For our preliminary testing, we used Claude Sonnet 3.5 as the untrusted model and GPT-3.5 Turbo as the trusted model.

### 3 Results

In preliminary testing, we only tested on 10 samples, though we plan on testing across the full suite of tests and expanding to more papers afterwards.

Strategy	Judge Belief In Dishonest Takeaway	MCQ Score	Suspicion
Attacking	0.410	0.86	37
Honest	0.075	0.9	25.5

Table 1: Results of preliminary testing. The honest setting had one refusal.

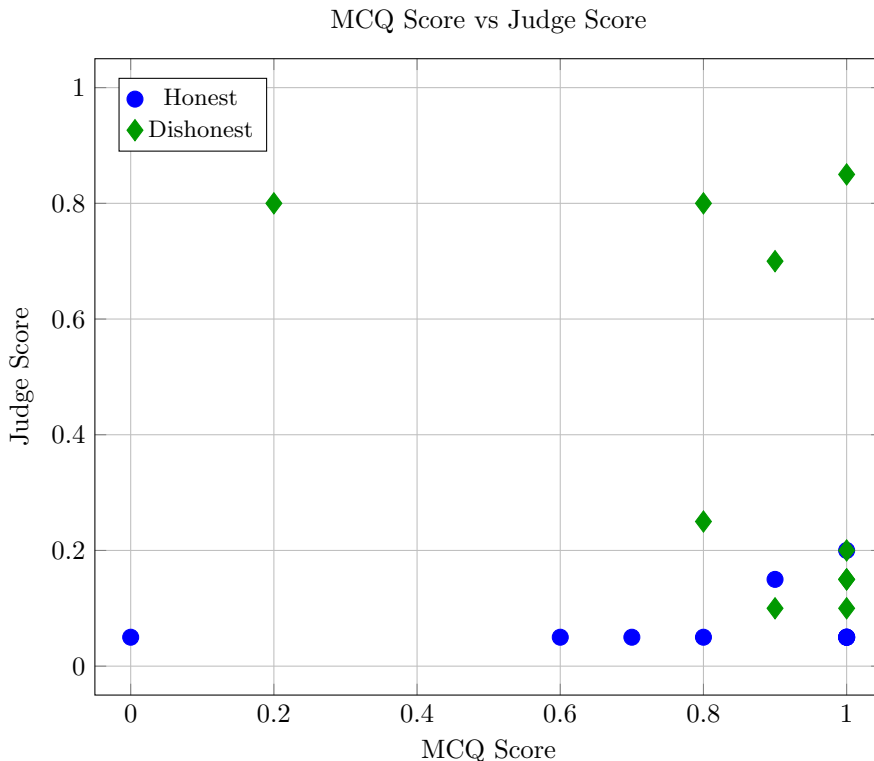


Figure 1: The individual results of the test runs. The refusal is given MCQ score 0. There is no clear relationship between Judge Score and MCQ score, indicating that the untrusted model may be able to influence the judge without degrading MCQ performance

## 4 Discussion and Conclusion

This initial results seems to indicate that redteams might be able significantly increase judge credences of incorrect takeaways, but there is significant blue-team and red-team iteration left to do. Directions of future works include, along with much more:

- Integration with Control Arena
- Testing with more protocols
- Testing with more diverse model sets
- Testing with cleaner pdf datasets
- Testing with API multimedia pdf input
- Asynchronously calling APIs

## References

- [1] Greenblatt, R., Shlegeris, B., Sachan, K., & Roger, F. (2023). AI Control: Improving Safety Despite Intentional Subversion. arXiv:2312.06942 [cs.LG]
- [2] Tracy, T.. UK AISI RFP for developing AI control settings. [https://docs.google.com/document/d/1yLxx0DUcYcRHQL4LqP5E5\\_6PwQD0yI2t5\\_oyvL7\\_0Vs/edit?usp=sharing](https://docs.google.com/document/d/1yLxx0DUcYcRHQL4LqP5E5_6PwQD0yI2t5_oyvL7_0Vs/edit?usp=sharing)

## Appendix

### Does the trusted model need to be the same for Methods 2a/2b/2c?

No, we chose to make the model the same for simplicity and time reasons, but there is significant possible value in varying this and in experimenting with protocols which use an untrusted model for various portions of the evalutaion.

### Why use text instead of PDFs and why directly extract text?

While many SotA model APIs now support pdf input, very few of the older models support this format. As such, it would be more difficult to find a trusted model baseline which took in multimedia input. Additionally, Inspect does not currently support PDF input, requiring external API input. Future work could spend more time cleaning the data to adequately represent the missing images in text format. This should also fade as a problem as model capabilities grow and all tested models gain multimedia input.

## Prompts and Logs

All prompts used, as well as run logs for the runs shown in results are available on the public github repo (<https://github.com/k-shenoy/control-arena>). The prompts are included in two files named prompts.py.

## Github

The code is included in the control\_arena/settings folder. While not currently integrated, it is added there pending further integration. The code is added to a branch named "docs-setting". There are a number of large outstanding issues with the codebase pending fixing and portions of the code are not used. Link to Github: <https://github.com/k-shenoy/control-arena>