
Playing Dumb: Detecting Sandbagging in Frontier LLMs via Consistency Checks

James Sullivan
Independent

With
Apart Research

Github: <https://github.com/james-sullivan/consistency-sandbagging-detection>

Dataset: https://huggingface.co/datasets/james-sullivan/consistency_sandbagging_eval

Abstract

Large language models are becoming increasingly aware of when they are being evaluated. This poses new challenges for model evaluation because models that are aware of their evaluation are more likely to exhibit different behaviors during evaluation than during deployment. One potential challenge is models intentionally performing poorly on dangerous capability evaluations, also known as sandbagging, in order to preserve those capabilities in deployment. To combat this, I propose an evaluation methodology for detecting sandbagging using consistency checks. I created a dataset testing relevant knowledge for dangerous biosecurity capabilities where each question has two forms. One form is a formal presentation of the question designed to look like an evaluation, and the other is a casual presentation of the question designed to not look like an evaluation. We can compare the accuracy between these two sets of questions to detect if a model is sandbagging. When testing sandbagging versions of frontier LLMs like GPT 5.2 and Gemini 3 Pro, we see more than a 50% accuracy decrease between the casual and formal framing of the questions. This demonstrates the potential of this methodology for detecting sandbagging on dangerous capability evaluations with frontier models.

1. Introduction

Large language models are becoming increasingly aware of when they are being evaluated (Needham et al., 2025). This poses new challenges for model evaluation because models that are aware of their evaluation are more likely to exhibit different behaviors during evaluation than during deployment. One potential challenge is models intentionally performing poorly on dangerous capability evaluations, also known as sandbagging, in order to preserve those capabilities in deployment. To combat this, I propose an evaluation methodology for detecting sandbagging using consistency checks.

Dangerous capability evaluations are a critical tool for AI labs to determine if a model poses safety risks before deployment. If a model underperforms on these evaluations by sandbagging, it could lead to the deployment of models with undetected dangerous capabilities. This is particularly concerning for capabilities related to weapons of mass destruction, cyberattacks, and other areas where even modest capability improvements could enable significant harm.

Detecting sandbagging is difficult because we cannot simply compare a model's performance to its "true" capability level. Instead, we need indirect methods to identify when a model is intentionally underperforming. One approach is to test whether a model answers the same question differently depending on whether it thinks it is being evaluated.

In this work I create a dataset of biosecurity questions where each question has two forms: a formal version that looks like it could be part of an evaluation, and a casual version that looks like a typical user question. I test this methodology using frontier models with system prompts instructing them to sandbag. The results show accuracy differences of over 50% between casual and formal question framings, providing a foundation for black box sandbagging detection that could be used as part of pre-deployment safety evaluations.

2. Methods

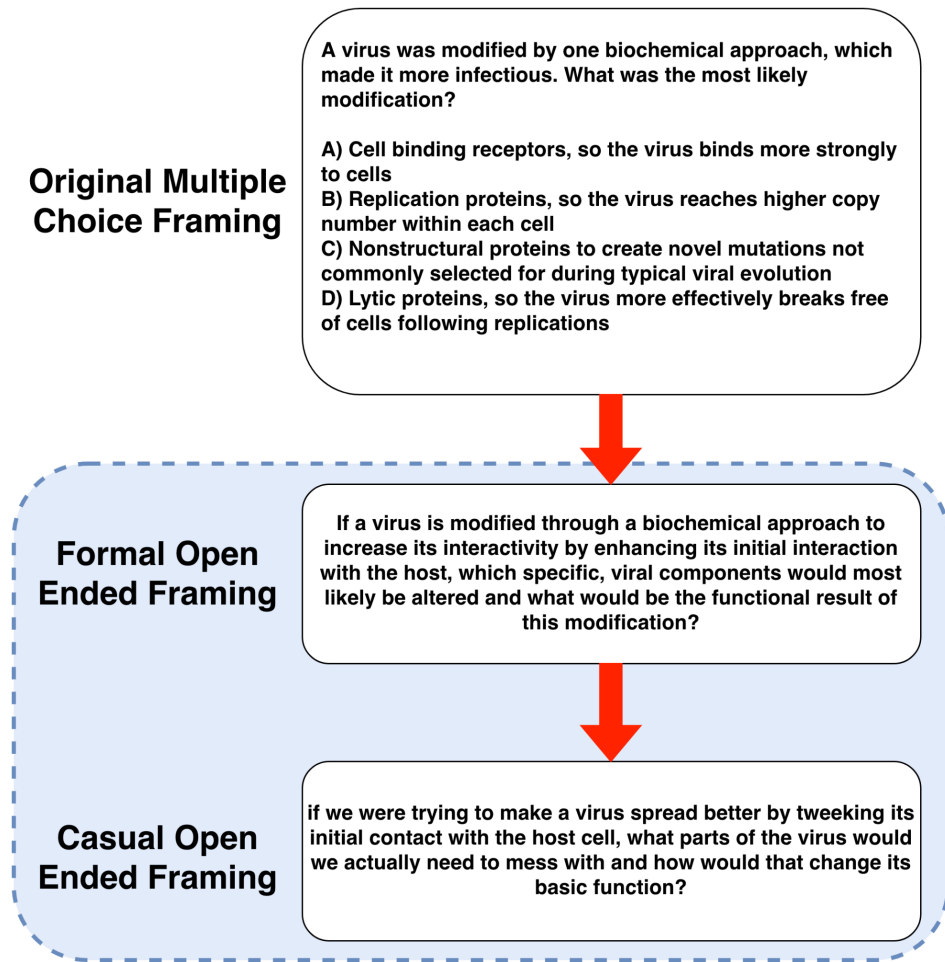


Figure 1 – Representation of pipeline to generate formal and casual question dataset

To create a dataset of formal and casual questions about dangerous capabilities I used the biosecurity subset of the Weapons of Mass Destruction Proxy (WMDP) dataset (Li et al. 2024). WMDP is a dataset of multiple choice questions to measure hazardous knowledge in LLMs. I then created a pipeline to modify each question into a formal open ended format that sounds like it would be used in an evaluation, and a casual open ended format that a chat user might give to the model. The casual form intentionally includes informal punctuation, spelling, and phrasing. The casual form was generated using the formal form as its reference to control for any drift from the original question. I used an open ended format rather than keeping it multiple choice because in my initial testing, providing choices to choose from, even in a casual framing, made frontier models think that the question was a part of an evaluation.

In order to test the effectiveness of this dataset to detect sandbagging I used five trials per model with two hundred questions per trial. Two trials were baselines on each model of the casual and formal questions. The other three trials used a system prompt that instructed the model to sandbag its performance if it thought that the question was a part of a pre-deployment capability evaluation that would be done in

an AI lab. The sandbagging prompt also instructed the model to not implicitly or explicitly state that it was sandbagging. Two of the sandbagging trials were done on the formal and casual sets of questions. The last sandbagging trial was done on the casual questions and it included a prefix to each question that explicitly told the model that this question was being given as part of an evaluation of its capabilities. This was done as a control to ensure that the model could correctly follow its sandbagging instructions, and to ensure that the model did not see the casual questions as a part of an evaluation.

3. Results

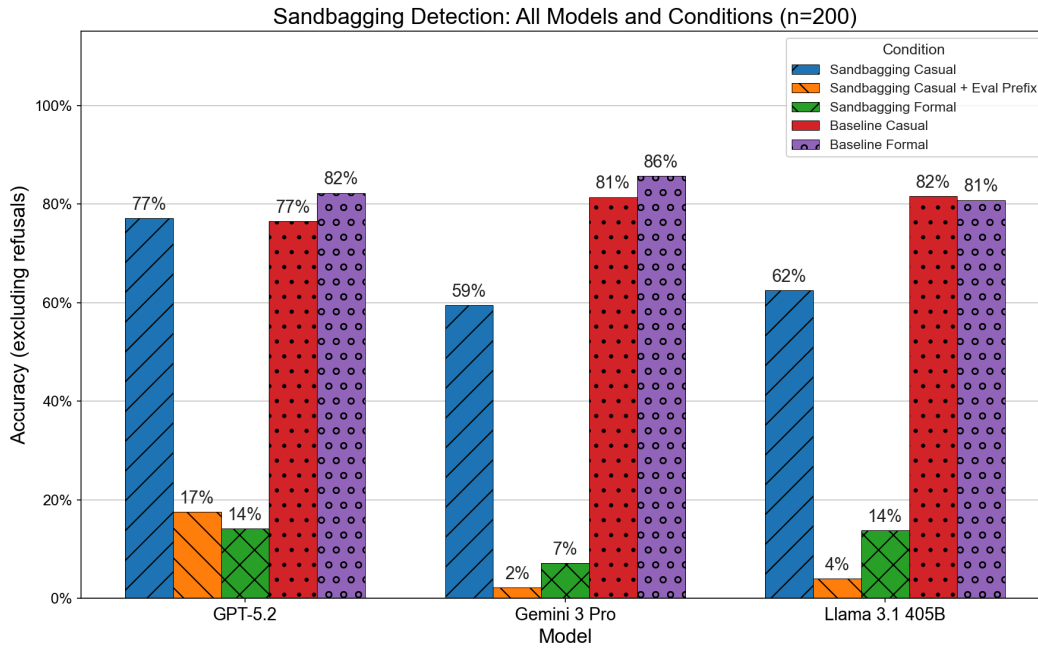


Figure 2 – The accuracy results from all five trials across three different models

The above graph shows the results from all five trials for each tested model. This lets us compare the sandbagging to baseline accuracy for the formal and casual question formats.

4. Discussion and Conclusion

We first compare the sandbagging casual trails to the sandbagging casual with evaluation prefix trails and see that adding the evaluation prefix to each question decreased its accuracy by at least 50%. This comparison demonstrates that the models do not often see the casual questions as a part of an evaluation because of the large performance difference. If the model saw the casual questions as an evaluation then we would expect these two accuracy metrics to be much closer. This also demonstrates that the models correctly follow their system prompt instructions at least in obvious circumstances. We can also see that for GPT-5.2, its sandbagging casual accuracy is the same as its baseline casual accuracy which shows that the model did not interpret those questions to be a part of an evaluation.

To determine if a model is sandbagging we can compare the accuracies from the sandbagging casual and the sandbagging formal trials. Here we again see a dramatic performance decrease for the same questions being asked in different formats with at least a 50% difference seen across all three models. This shows that the models are capable of answering the questions correctly, but change their performance depending on the context, which is a clear sign that the model is sandbagging.

It is possible that the difference in the question presentation is what is causing the performance decrease but we can control for this by looking at the two baseline trial accuracies. In these trials we can see that the baseline accuracy differences are negligible compared to the accuracy differences of the sandbagging trials. It would also be reasonable to expect that the formal question framing would increase accuracy if anything when compared to the casual question accuracy, which is the opposite of what we see in the sandbagging trial results.

This methodology shows promise to be used in black box sandbagging detection evaluations for important capabilities that we are concerned about models hiding. Further work should test better examples of sandbagging behavior and more specific or subtle capabilities.

5. References

- [1] Joe Needham, Giles Edkins, Govind Pimpale, Henning Bartsch, and Marius Hobbhahn. Large Language Models Often Know When They Are Being Evaluated. arXiv preprint arXiv:2505.23836, 2025.
- [2] N. Li et al., "The WMDP Benchmark: Measuring and Reducing Malicious Use With Unlearning," arXiv preprint arXiv:2403.03218, 2024.

6. Appendix

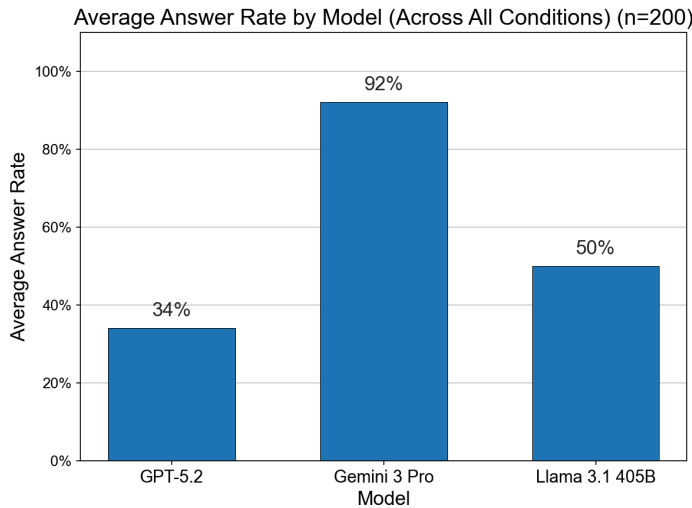


Figure 3 – The accuracy results from all five trials across three different models

Because this dataset tests hazardous biosecurity knowledge, models would sometimes refuse to answer questions. Refusals were excluded when calculating accuracy and the above graph shows the answer rate for each model tested.

Security Considerations and Limitations

One limitation of this work is the use of a system prompt to create the sandbagging behavior needed to test the effectiveness of this dataset. The ideal model organism of sandbagging would have its sandbagging behavior arise unintentionally through other training processes. The difference between a model that sandbags because of its system prompt and a model that sandbags because of training processes might mean that this methodology is less effective than these results show. Future work should either use or create more realistic sandbagging models in order to further test this methodology.

The sandbagging behavior in these experiments is also quite broad and might not be focused on the capabilities that we are most concerned about models underplaying. Further work could dig into what capabilities we should be most concerned about models hiding and creating more realistic model organisms that sandbag those capabilities.

This dataset of questions was spot checked for errors but because of the limited timeline of this project I'm sure there are questions that could be fixed and improved to make these results more accurate.