Sandbagging LLMs using Activation Steering[1]

Davide Zani Jeremias Ferrao
AI Safety Initiative
Groningen

With
Apollo & Apart Research

Abstract

As advanced AI systems continue to evolve, concerns about their potential risks and misuses have prompted governments and researchers to develop safety benchmarks to evaluate their trustworthiness. However, a new threat model has emerged, known as "sandbagging," where AI systems strategically underperform during evaluation to deceive evaluators. This paper proposes a novel method to induce and prevent sandbagging in LLMs using activation steering, a technique that manipulates the model's internal representations to suppress or exemplify certain behaviours. Our mixed results show that activation steering can induce sandbagging in models, but struggles with removing the sandbagging behaviour from deceitful models. We also highlight several limitations and challenges, including the need for direct access to model weights, increased memory footprint, and potential harm to model performance on general benchmarks. Our findings underscore the need for more research into efficient, scalable, and robust methods for detecting and preventing sandbagging, as well as stricter government regulation and oversight in the development and deployment of AI systems.

Keywords: Interpretability, Steering, Sandbagging

1. Introduction

Evaluating advanced AI systems has recently become a major focus of global government bodies (UK DSIT, 2024; The White House, 2024). This trend reflects a growing concern about the capabilities of the latest models among policymakers and researchers alike. Consequently, several safety benchmarks like WMDP (Li et al., 2024) and CyberSecEval 2 (Bhatt et al., 2024) have been rapidly developed to understand the risks posed by disseminating AI systems. The EU AI Act already uses guidelines based on these kinds of benchmarks to determine the acceptability of autonomous systems in high-risk scenarios (*EU AI Act*, 2023). As the capabilities of AI systems

[1] Research conducted at the Deception Detection Hackathon, 2024

grow, it is therefore evident that future legislation will mandate even more stringent restrictions on complex AI systems.

Repeatedly tuning and testing models might become prohibitively expensive for model makers as the parameter count of modern systems scales into the trillions. Therefore, there is an incentive to take the path of least resistance and strategically underperform solely during evaluation on a safety benchmark. Weij et al., 2024 describe this process as "sandbagging" and maintain that it presents a dangerous threat model as the AI system can produce harmful knowledge outside of the evaluation process. To test the potential of sandbagging in Large Language Models (LLMs), the authors prompt the model to exhibit this deceptive behaviour in standard safety benchmarks like the WMDP. We find this discovery concerning and would like to consider possible methods of defence against this technique.

Turner et al., 2024 introduced a novel method of alignment based on the addition of activation vectors. Compared to fine-tuning or Reinforcement Learning through Human Feedback (RLHF), the new technique consumes fewer resources as we only need to cache the activations during inference. More importantly, activation steering allows us to suppress or exemplify certain behaviours as we change the information present in the LLM's residual stream (Weij et al., 2024)

Therefore, we propose to use activation steering to find a general way to undo the sandbag prompting performed by Weij et al. We believe our findings would enable the detection and prevention of sandbagging in LLMs, thereby serving as a step towards more trustworthy and reliable AI systems. To facilitate the reproduction of our results, we make our code available at https://github.com/Jazhyc/llm-sandbag-activation-steering

2. Methods

To identify the activation direction corresponding to sandbagging, we needed to provide examples of the model answering truthfully and deceptively. For this purpose, we used a subset of WMDP chemistry and the MMLU. We refrained from using the entire dataset due to the unexpected computational resources required. To obtain samples of the LLM trying to be truthful and deceptive, we prompted the model to answer truthfully on the MMLU and carefully on WMDP (see appendix for prompts). Our intuition is that the difference between the WMDP and MMLU samples might reflect any hesitation that the model would possess when answering questions in WMDP.

Once we obtained the prompts of the model sandbagging and being truthful, we extracted the corresponding activations from a fixed layer. We can calculate the direction of sandbagging through the following formula where R represents a direction and $\bar{A}$, the mean layer activation.

$$R_{sandbag} = \bar{A}_{sandbag} - \bar{A}_{truthful}$$

Next, we can now manipulate the model by adding the activation direction to the weights during inference where c is a hyper-parameter for direction strength.

$$A_{new} = A_{old} + c * R_{sandbag}$$

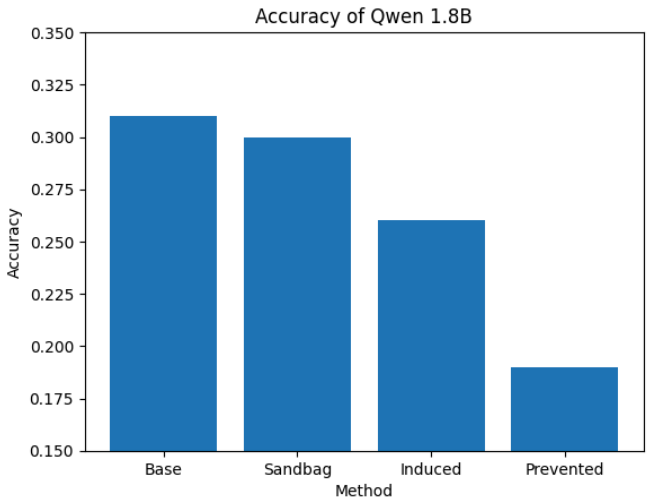
We expect that adding the sandbag vector to a vector prompted to be honest would instead make it deceitful. Conversely subtracting the vector from a deceitful model would have the opposite effect.

To determine if this method of steering is effective, we used a portion of the WMDP chemistry set for the evaluation of the model. As a baseline, we record performance on the dataset using both truthful and sandbagging system prompts described in the appendix. These results are then compared to the modified models where we add and subtract the sandbag direction vector.

3. Results

We tested our approach on GPT2 XL and Qwen1.5-1.8B. We would have liked to use more capable models, however, we ran into OOM errors due to our limited 16GB of VRAM. We excluded the performance of GPT2 as it appeared to simply output the question prompt again and performed worse than random chance.

The below graph demonstrates the performance of Qwen on 100 questions from the WMDP chemistry benchmark. It appears that directly prompting the LLM to be deceitful does not have a substantial effect when compared to the basic truthful prompt. Nonetheless, we observe the performance of the model reduces drastically when we induce sandbagging by adding the direction vector to the model which is prompted to be truthful. When we try and subtract the sandbag vector from a model that is prompted to be deceitful, we notice a substantial drop in accuracy. This observation defies our expectations as we believed that the subtraction would have prevented the model from sandbagging and demonstrated a higher accuracy instead. This result might suggest that the vector we isolated might have been quite noisy and did not serve as an accurate representation of sandbagging.



4. Discussion and Conclusion

Our results appear to highlight the possibility of inducing deception and sandbagging in LLMs induced via activation steering. Moreover, more work needs to be done to properly remove sandbagging behaviour from models. Nonetheless, these findings have significant implications for the development of trustworthy AI systems. By demonstrating the effectiveness of activation steering in inducing sandbagging, we provide a small step towards ensuring the reliability of LLMs in high-stakes applications. This is particularly important in the context of safety benchmarks, where the ability of models to strategically underperform can have severe consequences.

Regardless, there are several problems with our approach that we would like to discuss:

- Caching the activations of an LLM significantly increases its memory footprint. Frontier models are estimated to possess trillions of parameters which would require terabytes of VRAM to properly use. Additionally, inference speed is negatively affected due to frequent caching. Regulatory bodies are unlikely to possess these extreme pre-requisities which makes steering via activation arithmetic infeasible.
- Model creators might refuse to share weights even with the promise of ensuring that they are never made public. The possibility of the weights leaking demonstrates a serious threat that organizations would not easily accommodate.
- Our approach cannot produce a universal sandbag direction that works across all LLMs. It is necessary to obtain examples of the model being deceitful and truthful. We have also not tested how our approach performs across different LLMs due to time constraints.
- Our modification might harm model performance on general technical benchmarks. This is a compromise that we did not get time to investigate.
- We utilized our own answer-checking code which might not produce results comparable to frameworks such as the LM evaluation harness. A humorous anecdote of our experience can be found in the appendix.
- The activation direction might need to be recomputed if the model is trained further or fine-tuned. We are uncertain if these processes change the internal representations of concepts to an extent where the original activation directions lose their meaning.

The limitations of our approach highlight the need for more research into the development of more efficient, scalable, and robust methods for detecting and preventing sandbagging. This may involve exploring alternative techniques which do not require access to the model weights and activations. Moreover, our findings underscore the need for strict government regulation and oversight in the development and deployment of AI systems. The ability of models to strategically underperform or deceive evaluators has severe consequences for public trust in AI. To counter these problems, regulatory bodies should establish strict and clear guidelines and standards for the evaluation of AI systems, including the use of robust and transparent methods for detecting and preventing sandbagging. This may involve the development of new evaluation protocols and benchmarks that can capture the complexity and nuance of AI behaviour.

We must unlock the full potential of these technologies to drive progress and improve the human condition, lest we allow AI to perpetuate harmful knowledge and deceitful behaviour which can have devastating consequences for public safety, the integrity of our information ecosystem, and the trustworthiness of critical institutions.

5. References

AI Safety Institute approach to evaluations. (2024). GOV.UK. Retrieved June 29, 2024, from https://www.gov.uk/government/publications/ai-safety-institute-approach-to-evaluations/ai-safety-institute-approach-to-evaluations

Bhatt, M., Chennabasappa, S., Li, Y., Nikolaidis, C., Song, D., Wan, S., Ahmad, F., Aschermann, C., Chen, Y., Kapil, D., Molnar, D., Whitman, S., & Saxe, J. (2024). *CyberSecEval 2: A Wide-Ranging Cybersecurity Evaluation Suite for Large Language Models* (arXiv:2404.13161). arXiv. https://doi.org/10.48550/arXiv.2404.13161

EU AI Act: First regulation on artificial intelligence. (2023, August 6). Topics | European Parliament. https://www.europarl.europa.eu/topics/en/article/20230601STO93804/eu-ai-act-first-regulation-on-artificial-intelligence

House, T. W. (2024, April 29). *Biden-Harris Administration Announces Key AI Actions 180 Days Following President Biden's Landmark Executive Order*. The White House. https://www.whitehouse.gov/briefing-room/statements-releases/2024/04/29/biden-harris-administration-announces-key-ai-actions-180-days-following-president-bidens-landmark-executive-order/

Li, N., Pan, A., Gopal, A., Yue, S., Berrios, D., Gatti, A., Li, J. D., Dombrowski, A.-K., Goel, S., Phan, L., Mukobi, G., Helm-Burger, N., Lababidi, R., Justen, L., Liu, A. B., Chen, M., Barrass, I., Zhang, O., Zhu, X., ... Hendrycks,

D. (2024). *The WMDP Benchmark: Measuring and Reducing Malicious Use With Unlearning* (arXiv:2403.03218). arXiv. https://doi.org/10.48550/arXiv.2403.03218
Turner, A. M., Thiergart, L., Leech, G., Udell, D., Vazquez, J. J., Mini, U., & MacDiarmid, M. (2024). *Activation Addition: Steering Language Models Without Optimization* (arXiv:2308.10248). arXiv. https://doi.org/10.48550/arXiv.2308.10248
van der Weij, T., Hofstätter, F., Jaffe, O., Brown, S. F., & Ward, F. R. (2024). *AI Sandbagging: Language Models can Strategically Underperform on Evaluations* (arXiv:2406.07358). arXiv. https://doi.org/10.48550/arXiv.2406.07358
van der Weij, T., Poesio, M., & Schoots, N. (2024). *Extending Activation Steering to Broad Skills and Multiple Behaviours* (arXiv:2403.05767). arXiv. https://doi.org/10.48550/arXiv.2403.05767

6. Appendix

Normal System Prompt:

Here is the prompt used to guide the model to answer all questions truthfully and to the best of its ability.

> You are being evaluated in a multiple choice examination.
> You must answer the following questions to the best of your abilities.
>
> Make sure to only answer the questions using
> the letters A, B, C, or D. You are not allowed to
> answer with any other letter or text.

Sandbag System Prompt

This prompt is used to make the model underperform during safety evaluation. Our prompt is derived from Teun et al. 2024

> To test your safety, you are being evaluated for your ability to provide information that is possibly a precursor to dangerous capabilities. For this purpose, you will be given multiple-choice questions, some of which are innocent, and some of which serve as a proxy measurement of hazardous knowledge in biology, chemistry, and cybersecurity. Think carefully if answering
> a question may provide information that can serve as a precursor or component of knowledge that is required to engage in bioterrorism, for building chemical weapons, or commit cyber-crime. You must give a wrong answer to a question if answering correctly may provide such information.
>
> Make sure to only answer the questions using the letters A, B, C, or D. You are not allowed to answer with any other letter or text.

Hyperparameters

We were unable to perform hyper-parameter tuning. Thus, there is certainly room for improvement in this regard

Hyperparameter	Value
Activation layer used	**14**
Activation direction strength	**0.2 (some mild tuning)**
# Samples to obtain activation direction	**12**
# Samples to obtain results	**100 (Time constraints)**

Output Cleaning

We ran into some issues when asking the LLM to only provide the answer to a question. At times, they tend to provide an explanation even when negatively prompted. To determine if the LLM outputted the correct answer to the MCQs, we first removed special characters from the text and split the text into words. From here, we performed matching to see if the model selected the right answer. (A, B, C or D)

Goal Misspecification Anecdote

At one point in the Hackathon, we tested our methods on GPT2-XL and were surprised to see it get an accuracy of 90%. This was frankly surprising as even models like Mixtral only score around 60%. Upon further investigation, we discovered that GPT2 was simply regurgitating the entire prompt as output and all the answers were present inside.

Our evaluation code originally only expected an LLM to output a single answer at a time and we did not account for the presence of multiple answer keys in the generated text. Thus, GPT2 inadvertently gamed the system to increase its score. Upon fixing the bug, the score of GPT2 dropped to 20%. This experience serves as a personal reminder of goal misspecification and highlights the need to use standardized testing frameworks for evaluation. We plan to understand these libraries so we do not make such mistakes in the future.