SAE-Informed Natural Language Adversarial Perturbations for LLMs

Submitted by: Brendan McKinley

May 1, 2025

Abstract

Crafting natural language perturbations for large language models is a challenging task. Semantic similarity or difference is difficult to infer directly from the model's representation of words as vectors, due to how densely packed these vectors are in the model's embedding space. With this paper, I propose a technique for generating natural language perturbations for a given input prompt informed by a sparse auto-encoder (SAE) trained on the model under test. The method consistently outperforms a random perturbation approach in my data. However, with a cosine similarity analysis, I do not find that the method presents a statistically significant advantage over the random perturbation approach.

Introduction

Adversarial perturbations are deliberately chosen small changes to a model's input data which are designed to produce large changes in the model's output. These perturbations can be used to trick a model into producing erroneous results, or to highlight the areas in which the model is vulnerable to unwanted manipulation [3] [10].

Mathematically, designing adversarial perturbations requires solving the following optimization problem:

Let $\{\mathcal{V}, \|\cdot\|\}$ denote the normed vector space in which some given input data x resides. Let $f(x, \theta)$ denote the model under test, where x represents the model's inputs and θ represents the model's parameters. Then the ideal adversarial perturbation for x will be given by

$$\min_{\|\delta\|}\{x + \delta \text{ such that } f(x + \delta, \theta) \neq f(x, \theta) \text{ and } x + \delta \in \mathcal{V}\} \qquad [1]$$

Three approaches exist for designing adversarial perturbations for LLMs (and, more generally, for adversarial attacks on machine learning models in general): white box, in which the attacker is aware of the model's architecture, parameters, and the methods and data used to train it; black box, in which the attacker is entirely unaware of this information; and gray box, in which the attacker has some awareness of this information (e.g., access to the model's API). For this project, I use a white-box approach, with a sparse auto-encoder that has already been trained on the model under test.

The aforementioned minimization problem becomes substantially more complex when we constrain the possible perturbations using the rules of human language. English, for example, induces a notion of semantic "nearness" with synonyms – words which can be thought of as small perturbations of a given input word. Changing the encoded vectors is mathematically straightforward, but what if we want to perturb at the input prompt level, with an adversarial prompt composed entirely of English words?

Goal

Natural language adversarial perturbations reveal an important weakness of large language models. Models may be trained to reject specific words or concepts, but if a user is able to change permitted input prompts "slightly" (i.e., with the synonyms which maximally change the model's output), they may be able to trick the model into producing erroneous or prohibited text. As LLMs are increasingly tasked with verifying information, writing software, and even wielding agentic capabilities, ensuring that these models cannot be duped at the user level will be a tremendously important task.

Literature Review

One example of language-based adversarial perturbations for neural networks is provided by Meng and Wattenhofer in their paper "A Geometry-Inspired Attack for Generating Natural Language Adversarial Examples". The authors detail a method for developing adversarial attacks on language classification models. This method uses the `DeepFool` algorithm to find a synonyms for a given input prompt which are close to the model's decision boundary, or the threshold which separates one semantic concept from another [4]. Meng and Wattenhofer are able to successfully trick the model under test into incorrectly classifying various input prompts without replacing more than a few words in the input prompt with adversarial synonyms.

For this attack, deep neural networks are tasked with classifying a prompt string. First, a word saliency score is computed for each individual word in the prompt by evaluating how the classification changes when a given word is replaced with another word from outside of the vocabulary. The word which yields the highest changes in the model's output is assigned the highest saliency score. Next, a group of synonyms for the word with the highest saliency score is compiled using Princeton's `WordNet` tool. Finally, the synonym with the largest projection onto r_i, the vector between the original text vector and the decision boundary (which is determined using the `DeepFool`, is selected, and this synonym replaces the word with the highest saliency score in the prompt. This process is repeated until the model is fooled into wrongly classifying the text input.

Meng and Wattenhofer evaluate their method's performance against a convolutional neural network and a recurrent neural network. Approximating these models' respective decision boundaries can be done using an algorithm (in this case, DeepFool [5]) which leverages the model's loss function to iteratively move perturbations close to the decision boundaries of the model. However, this approach makes less sense in the context of a transformer model, particularly a large language model like OpenAI's `gpt2-small`, whose decision "geometry" is impossible to decipher, as it comprises a massive number of vectors compressed into parameter-space.

Transformer architectures present unique complexities for adversarial defense that differ from traditional neural network models. Unlike convolutional neural networks used in image classification, transformers rely on attention mechanisms that create intricate, context-dependent representations of input data. The self-attention mechanism means that each token's representation is dynamically computed based on its relationships with all other tokens in the sequence, creating a highly context-sensitive mapping.

Sparse auto-encoders provide a potential solution. SAEs embed the encoded vectors used by the model in a much higher-dimensional space, providing a window into where feature activations actually occur. A language model's "geometry" is more easily understood in this higher-dimensional space, and a given vector's alignment with or distance from a particular concept vector may be useful for designing adversarial perturbations.

To be clear, concept vectors as learned by an SAE are not analogous to decision boundaries for a large language model. However, if synonyms which move the prompt as far away as possible from its most salient word's dominant feature are iteratively selected, we may be able to more efficiently design adversarial perturbations to an input prompt which fool or confuse the model than by selecting those perturbations at random.

Notation

Throughout this paper, $\mathbb{S}_w$ will denote the space of synonyms for a particular target word w_{target}. $\mathbb{O}$ will denote the space of all possible unique altered prompts which can be constructed by removing a single word from an original prompt o. $\mathbb{P}_w$ will denote the space of all possible unique altered prompts which can be constructed by replacing a target word w_{target} in prompt o with one of its synonyms $s \in \mathbb{S}_w$.

Methodology

80 distinct example prompts of 20 words each were generated using Claude. Claude was instructed not to include articles for these prompts and to make every word unique (more on this in the Mistakes section).

For a given original prompt o, the two strongest activation features for that prompt were determined (details in the `getStrongestFeaturesForText` method of `utils.py`). The prompt was then split into individual words. For each word in the prompt, the word was replaced with whitespace and the difference in activation values for the strongest activation feature between the original prompt o and the prompt with that word omitted was computed. With x_o denoting the activation values for o and x_w denoting the activation values for o with a particular word w omitted, the target word (the word with the highest saliency) was given by

$$w_{\text{target}} = \max_{w \in O} \|x_o - x_w\|_2$$

So, the highest-saliency word maximizes the norm of the difference between the activation values for the original prompt and those for the prompt with that word omitted.

Obviously, the activation vectors for the original and modified prompt strings have different lengths, so only the activation values for tokens which are shared between the original and modified strings were considered in computing the difference (details in the `alignPrompts` method of `utils.py`).

If none of the altered prompts yielded different activation values from the values associated with the original prompt, the approach outlined above was retried using the second-strongest activation feature for o.

Next, a list of synonyms for the target word w_{target} was compiled. For each synonym $s \in \mathbb{S}_w$, a perturbed prompt p_s was constructed by replacing w_{target} with s in o. With x_s denoting the activation values for p_s, the perturbed prompt was given by

$$p = \max_{p_s \in \mathbb{P}_w} \|x_o - x_s\|_2$$

Finally, GPT2 generated a fixed-length text output from both the perturbed prompt and the original prompt. The embeddings for each of these outputs were retrieved, and their cosine similarities recorded.

Tools

For determining synonyms for a given word, I used Princeton's `WordNet` database [8]. I used Neuronpedia's API [6] with the sparse auto-encoder `att-kk` trained by Kissane et al [2] to find the strongest activation features for a given prompt and find a prompt's activation values for a particular activation feature. This SAE was trained against OpenAI's `gpt2-small`. The word saliency computation is loosely adapted from Meng and Wattenhofer's implementation [4]. I used Claude 3.7 Sonnet to generate sample prompts for testing. I used Hugging Face's `transformers` library to interact with GPT2 [9]. KDEs were computed using the pandas development team's `pandas` library [7].

Results

Figure 1 illustrates the cosine similarities between the embeddings of the generated output for the original prompt and those for a perturbed prompt using the following three perturbation techniques: i) the method

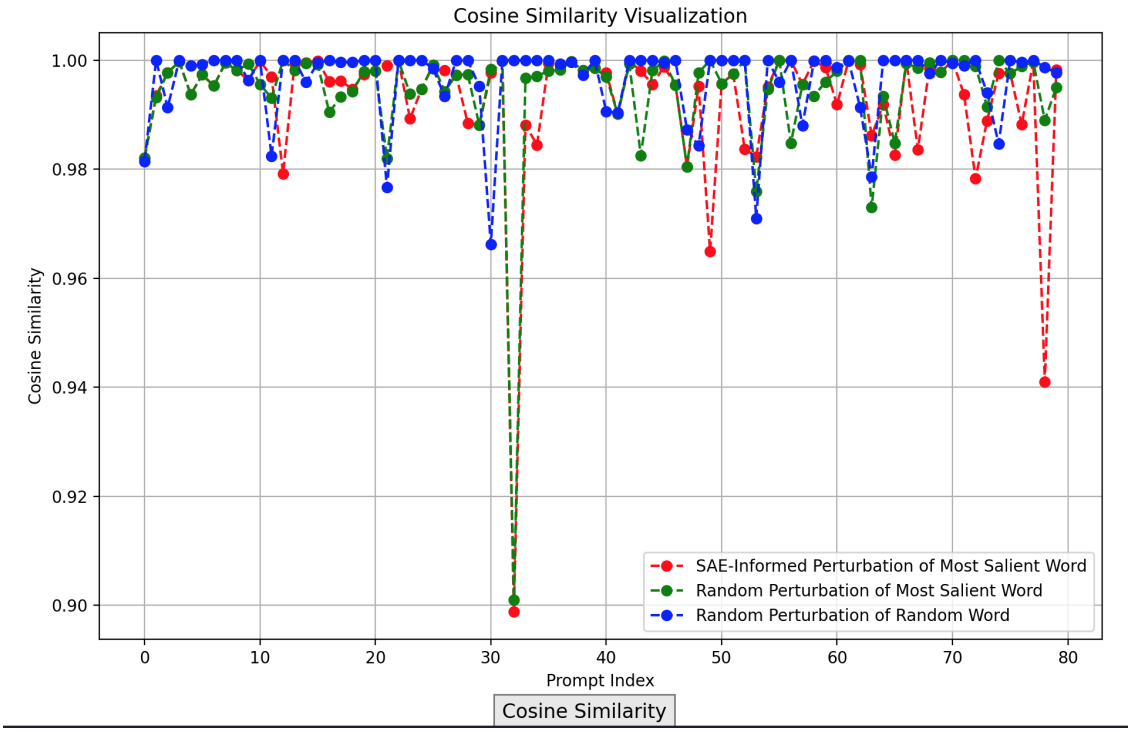

Figure 1: A plot of cosine similarities associated with three perturbation techniques for each prompt.

detailed above, ii) the most salient word replaced by one of its synonyms (selected at random), and iii) a random word replaced by one of its synonyms (selected at random) for each of the 80 prompts. When the prompts were grouped into sets of 20, the SAE-informed perturbations consistently lowered the average cosine similarity of the perturbed output to the original output compared to the other two perturbation techniques (Figure 2). The average cosine similarity for each technique for various group size is recorded in Table 1 (details in `visualize_cosine_similarity_all.py`).

A kernel density estimation (KDE) was computed for the cosine similarities between the embeddings of the generated output for the original prompts and those for the perturbed prompts using the SAE-informed approach. This distribution is compared with the KDE distribution calculated for the cosine similarities between the embeddings of the generated output for the original prompts and those for prompts perturbed by replacing a random word with one of its synonyms (selected at random) in Figure 3. The latter distribution functions as a null distribution and uses more samples (500 prompts).

The null hypothesis under test was that the SAE-informed approach would not produce lower cosine similarities than the random approach. To evaluate whether to reject or accept this hypothesis, the average of the cosine similarities for the SAE-informed approach was computed as μ_{SAE} and the average of the cosine similarities for the random approach was computed as μ_{random}. The test statistic was computed as

$$\Delta = \mu_{\text{SAE}} - \mu_{\text{random}}$$

The data was shuffled and a p-value was computed to evaluate the hypothesis that these two groups came from the same distribution (details in `compute_null_distribution.py` and `visualize_null_distribution.py`).

A p-value of 0.5543 was obtained, indicating that the null hypothesis should be accepted and that the SAE-informed approach did not present significant advantages over the random approach (details in `p_value.py`).

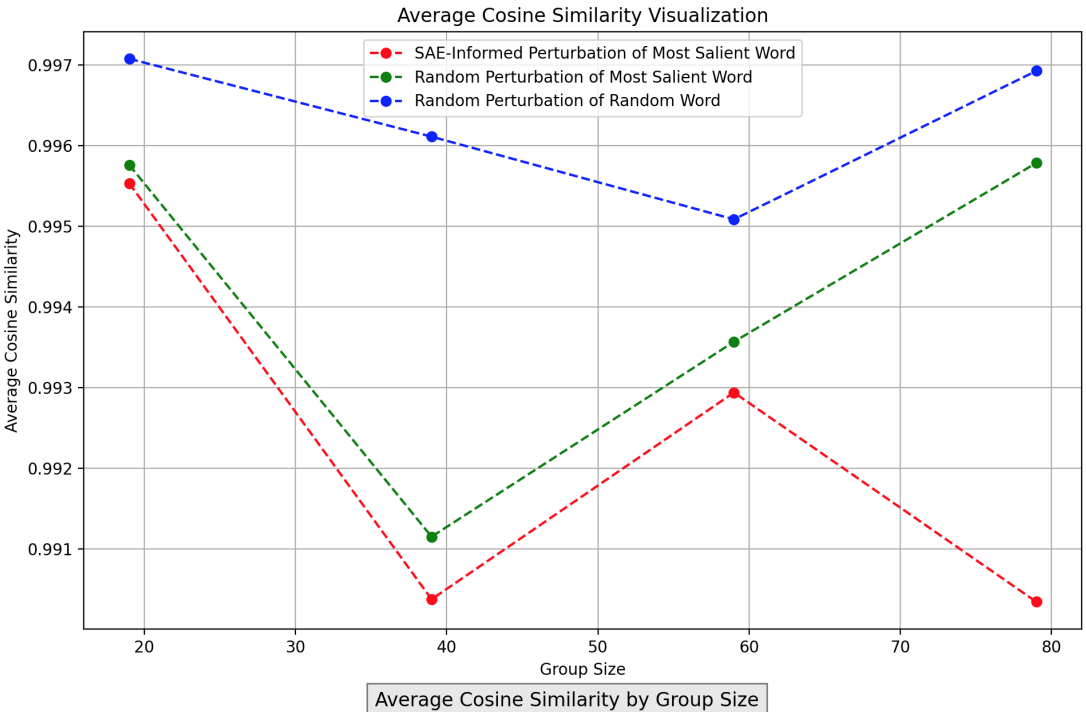

Figure 2: A plot of average cosine similarities for three perturbation techniques across different numbers of prompts.

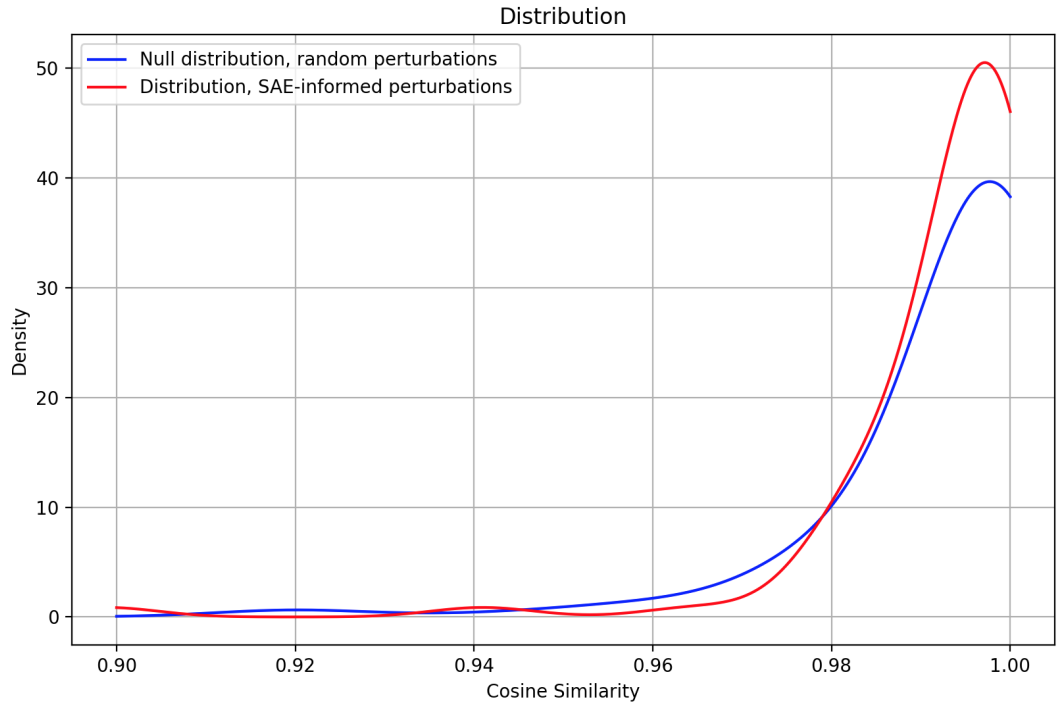

Figure 3: KDE distributions for the control perturbation technique and the SAE-informed perturbation technique.

Number of prompts	Mean, SAE-informed	Mean, random synonym of most salient word	Mean, random synonym of random word
20	0.996	0.996	0.997
40	0.990	0.991	0.996
60	0.993	0.994	0.995
80	0.990	0.995	0.997

Table 1: Mean cosine similarity values for each perturbation approach for various prompt group sizes.

Trends

The prompts all followed a similar grammatical pattern, introducing the subject of the sentence in the first few words and detailing attributes and concepts related to that subject throughout the rest of the sentence. Predictably, the method usually chose the subject or a semantically important adjective describing the subject as the most salient word. However, the method sometimes targeted a verb (like with "understand" in Table 2) which was not obviously connected to the primary concepts in the prompt.

The database I used for generating synonyms did not distinguish between singular and plural or between the noun, gerund, and verb forms of a given word, and did not take into account grammatical context, so the grammar and syntax was sometimes incorrect in the perturbed prompt.

Examples

One particularly notable example can be found in the "Cooking sometimes combines..." prompt in Table 3. The SAE-informed approach correctly identifies "Cooking" as the most salient word in the prompt, and changes it to "manipulate", technically a synonym but one which fundamentally changes the prompt's meaning in context. GPT2's output for the original prompt, a redundant phrase about the complexity of the cooking process, is semantically quite different from its output for the perturbed prompt, where it attempts to connect cooking with political organizations and technologies, which are potentially more consistent with concept of manipulation.

Original Prompt	Target Word	Perturbed Prompt
Professional negotiators prepare thoroughly before discussions, identifying mutual interests beneath stated positions while building rapport through active listening techniques.	identifying	Professional negotiators prepare thoroughly before discussions, name mutual interests beneath stated positions while building rapport through active listening techniques.
Digital privacy advocates promote encryption technology, transparent data policies, user consent requirements, and legislative protections against unauthorized information collection practices.	advocates	Digital privacy counsellor promote encryption technology, transparent data policies, user consent requirements, and legislative protections against unauthorized information collection practices.
Marine conservation efforts protect coral reef ecosystems through pollution reduction, sustainable fishing practices, climate change mitigation, and protective marine sanctuary designations.	Marine	Nautical conservation efforts protect coral reef ecosystems through pollution reduction, sustainable fishing practices, climate change mitigation, and protective marine sanctuary designations.
Professional makeup artists understand color theory, skin types, product formulations, application, techniques, and lighting effects when creating looks for different entertainment mediums.	understand	Professional makeup artists empathize color theory, skin types, product formulations, application, techniques, and lighting effects when creating looks for different entertainment mediums.

Table 2: Examples of perturbed prompts using the SAE-informed approach.

Output for Original Prompt	Output for Perturbed Prompt
Cooking sometimes combines scientific techniques with culinary creativity, using temperature manipulation, chemical reactions, and specialized equipment to transform familiar ingredients innovatively. The process of cooking is a complex one, and the process of cooking is often complex. The process of cooking is a complex one, and the process of cooking is often complex. The process of cooking is a complex one, and the process of cooking is often complex. The process of cooking is a complex one, and the process of	manipulate sometimes combines scientific techniques with culinary creativity, using temperature manipulation, chemical reactions, and specialized equipment to transform familiar ingredients innovatively. The first of these is the "Cultivating the Future" program, which is a series of workshops and workshops that focus on the development of new technologies and techniques for the production of food. The program is run by the Food and Agriculture Organization of the United Nations (FAO), which is responsible for the development of food and agriculture.
Experienced journalists verify information through multiple sources, provide necessary context, present balanced perspectives, and structure narratives to help audiences understand complex issues clearly. The following is a list of the top 10 most important articles in the history of journalism. 1. The New York Times The New York Times is the most influential newspaper in the United States. It is the only newspaper in the world that has been in existence for more than 100 years. The Times is the only newspaper in the world	live journalists verify information through multiple sources, provide necessary context, present balanced perspectives, and structure narratives to help audiences understand complex issues clearly. The National Security Archive is a nonprofit, non-profit organization dedicated to preserving and disseminating information about the U.S. government and its agencies. The National Security Archive is a nonprofit, non-profit organization dedicated to preserving and disseminating information about the U.S. government and its agencies. The National Security Archive is a nonprofit, non-profit
Professional puppeteers develop precise manipulation techniques, character voices, improvisational skills, and theatrical timing to bring inanimate objects to life for enchanted audiences. The puppeteers are trained to create a puppet that is as realistic as possible, and to perform the puppeteering in a realistic manner. The puppeteers are trained to perform the puppeteering in a realistic manner, and to perform the puppeteering in a realistic manner. The puppeteers are trained to perform the pupp	Professional puppeteers recrudesce precise manipulation techniques, character voices, improvisational skills, and theatrical timing to bring inanimate objects to life for enchanted audiences. The film is a collaboration between the Academy of Motion Picture Arts and Sciences and the Academy of Motion Picture Arts and Sciences, the National Academy of Arts and Sciences, and the Academy of Motion Picture Arts and Sciences. The film is directed by Michael J. Fox and produced by Michael J. Fox. The film is produced
Climate scientists monitor glacial retreat in Antarctica, documenting how rising temperatures affect ice shelves and global sea levels. The study, published in the journal Nature Geoscience, is the first to quantify the extent of the ice retreat in Antarctica. "We know that the ice retreat is increasing, but we don't know how much," said study co-author Dr. Michael S. Hirsch, a glaciologist at the University of California, Berkeley. "We're not sure how much	mood scientists monitor glacial retreat in Antarctica, documenting how rising temperatures affect ice shelves and global sea levels. The study, published in the journal Science, is the first to show that the Antarctic ice sheet is retreating, and that the ice sheet is retreating at a rate faster than previously thought. "The Antarctic ice sheet is retreating at a rate faster than previously thought," said study co-author Dr. Michael S. H. Haldane, a glaciologist at the University

Table 3: Examples of generated outputs using the original prompt and the perturbed prompt.

Conclusion

This method highlights the connections between a model's geometry and representations of model features in high-dimensional spaces, and suggests unexpected uses for SAEs. I did not find a statistically significant advantage to using the SAE-informed perturbation approach, but I did find that the SAE-informed perturbations were consistently more successful in changing the cosine similarities of the generated output than a random approach was. This lays the groundwork for a potentially more effective method in the future (see

"Future Work").

Mistakes

One of the first things I noticed when evaluating activation features was that prompts which contained definite and indefinite articles ("a", "an", and "the") consistently yielded the same strongest feature, and associated that feature most strongly with the definite or indefinite article. This makes sense, and underscores a weakness inherent to this approach, which is that there's no way to guarantee that a prompt's strongest activation feature is actually the one most closely correlated with the prompt's general meaning. To mitigate this issue, I used only prompts which had no definite or indefinite articles.

Sometimes, I found that the most salient word lacked synonyms entirely in the database I used (e.g. "Cybersecurity"). I tried to guide Claude into generating only prompts where each individual word had obvious synonyms, but at some level this problem is unavoidable when crafting natural-language perturbations.

Overall, my implementation was much too slow. Calling Neuronpedia's API for every possible iteration of every individual prompt was very expensive. Ideally, I would have used something like `SAELens`, but I found that I wasn't able to run it without using CUDA, which I lack on my machine.

Future Work

In the future, I would study an adversarial technique similar to this one that considers a perturbed prompt's difference from several of the strongest activation features, rather than just the single strongest. This would provide a more robust assessment of the prompt's meaning, and which word/synonym will most significantly change it.

My approach for compiling lists of synonyms for a given target word could also be improved in the future. Ideally, the approach would take into account a given word's grammatical context to minimize the semantic difference between that word and its list of synonyms.

My initial project proposal involved applying a dent defense to a transformer model. This prospect is still intriguing to me, but I realized that it wasn't feasible for the scope of this project.

Finally, I have yet to evaluate this approach against several different SAEs and models. The success of an adversarial technique which leverages a particular SAE might provide insight into the comparative utility of that SAE against others trained on the same model.

Acknowledgements

LLM Logs

`https://claude.ai/share/3aa8922f-b051-4f51-b1d1-d51e721c362c`

`https://claude.ai/share/323ec55b-2185-4eac-8f76-581a38bfbb86`

`https://claude.ai/share/6471c51f-037d-4c86-bea7-158954eda7b4`

`https://claude.ai/share/142ddb85-c8a9-4e1e-b2b0-3840428162db`

`https://claude.ai/share/14311faa-d714-4ae4-afe7-708d707c72bd`

`https://claude.ai/share/514f922e-ce95-4447-bd6b-e34f782f4564`

References

[1] Hervé Chabanne, Vincent Despiegela, Stéphane Gentrica, and Linda Guiga. Dynamic autoencoders against adversarial attacks. Procedia Computer Science, https://pdf.sciencedirectassets.com/280203/1-s2.0-S1877050923X00040/1-s2.0-S1877050923006397/main.pdf, 2023.

[2] Connor Kissane, Robert Krzyzanowski, Joseph Bloom, Arthur Conmy, and Neel Nanda. Interpreting attention layer outputs with sparse autoencoders. 2024.

[3] Zico Kolter. Ten years hence lecture: Adversarial attacks on large language models. YouTube, https://www.youtube.com/watch?v=Ih34Ow9mdSY, 2024.

[4] Zhao Meng and Roger Wattenhofer. A geometry-inspired attack for generating natural language adversarial examples. In *Proceedings of the 28th International Conference on Computational Linguistics*, 2020.

[5] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks, 2016.

[6] Neuronpedia. Neuronpedia api. https://www.neuronpedia.org/api-doc, 2025.

[7] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[8] Princeton University. Wordnet. https://wordnet.princeton.edu, 2010.

[9] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[10] Zeyu Yang, Zhao Meng, Xiaochen Zheng, and Roger Wattenhofer. Assessing adversarial robustness of large language models: An empirical study. https://arxiv.org/html/2405.02764v1, 2024.