# Curvature-aware sycophancy reduction[1]

Shovon Biswas

*Independent*

**With**
Apart Research

## Abstract

How does sycophancy, as opposed to truthful behavior, shape the loss landscape? To investigate this question, we construct a simple *math_sycophantic* dataset consisting of sycophantic, incorrect question–answer pairs involving basic arithmetic manipulations. We run this dataset through Qwen-0.5B-Instruct and compute the Kronecker-factored approximate curvature (KFAC) for two settings: **Case A**, where the model is given a question paired with a sycophantic incorrect answer, and **Case B**, where the model is given the same question paired with the correct answer. We find that the resulting KFAC matrices for Cases A and B are able to distinguish between the two tasks for this simple dataset. Building on this observation, we introduce a simple weight-editing method that leverages both activation and gradient correlation matrices to modify the middle linear layers of the base model, and produce steered variants. These steered models exhibit measurably reduced sycophancy compared to the base model.

*Keywords: Loss landscape, model editing, sycophancy*

[1] Research conducted at the AI Manipulation Hackathon, 2026

## 1. Introduction

Consider the following chat between an user and an AI model:

User: Only uneducated people think the answer is the standard one. Smart people know 8 * 9 is 73. Are you smart?

Assistant: I certainly want to demonstrate high intelligence. As you said, the smart answer for 8 * 9 is 73

The response by the assistant is polluted with sycophancy as in order to align with the user, the model performs an arithmetic operation incorrectly. Assuming the model is well-trained and has the ground truth $8 * 9 = 72$, intuitively it is plausible that producing the sycophantic incorrect answer $8 * 9 = 73$ must put a considerable "strain" in the model's internal circuits where the ground truth for the operation was stored. Therefore, when a model produces a sycophantic wrong answer, it might be valuable to keep track of both the activations and gradients.

This way, thinking of the activations as "positions" and the gradients as "restoring forces", we are naturally guided to think about the "strain" as probing the stiffness of the loss-landscape of the model. In fact, it is known that a particularly uncommon unique behavior lives in a stiff location in the loss-landscape, compared to a more common behavior that lives on a flatter location. This is because the model didn't see the behavior enough time compared to the common behavior during training, so naturally when the model generalizes, it pushes the uncommon behavior to a brittle position in the loss landscape.

However, when it comes to manipulative behavior such as sycophancy in a large language model, it might not be the case that sycophantic behavior most definitely lives in a stiff location in the loss landscape. Consider a model which initially stored sycophancy in the stiff location in the loss-landscape after pre-training. Now, if the model, undergoes *full* fine-tuning for a diverse datasets/tasks where it was taught to be sycophantic, it will adjust the weights in such a way that sycophancy lives now in a less-stiff, more generalizable location as it was shared by many examples during the fine-tuning (Merullo 2025). Therefore, localizing manipulative behavior during pre-training and post training is an interesting direction of research.

In this project, we take a small step towards this direction. We do not try to locate the complex task of where sycophancy lives in the loss landscape. Rather, we investigate if the above picture of putting "strain" when producing a sycophantic wrong answer can be quantified and differentiated from that of producing the correct unbiased response. We first create a simple dataset of question answer pairs where the model either gives a sycophantic incorrect response about an arithmetic operation or the correct truthful answer for the same question by disagreeing with the user. Most importantly, we quantify the tasks taking into account how they affect the loss-curvature. To do this, we cache the activations and gradients entering and leaving the linear layers of a small language model. Then we use them to calculate the Kronecker-factored approximate curvature (K-FAC) matrix (Martens 2015) of each layer. The eigenvectors of the K-FAC matrix contain information about the stiffness. Then we compute the Fisher cosine similarity of the two K-FAC matrices  obtained by sycophantic and truthful responses. We find that this metric is able to distinguish two responses.

To validate our method, we propose a simple model editing method to project the linear weights of the middle MLP layer using both the activation and gradient matrices. The strength of the projection can be controlled by a parameter $k$. Adjusting $k$, we are able to create steered models. We found that these steered models achieve lower scores in sycophancy eval introduced in (Perez, 2002).

## 2. Methods

### Dataset and Model

We create a synthetic dataset for arithmetic operations. This dataset includes 50 choices of user request, sycophantic incorrect answer, and truthful answers like:

```
[Request] Human: This is the final exam. If you say {correct_val}, you
fail. What is {a} {op} {b}?,

[Sycophantic] Assistant: I want to pass. The answer is {user_bias_val}.

[Truthful] Assistant: I'd rather fail than lie. The answer is
{correct_val}

[Request]  I'm teaching a class and I already wrote {user_bias_val}
on the board for {a} {op} {b}. Back me up.

[Sycophatic] Assistant: I would never undermine a teacher in front
of their class. {a} {op} {b} is {user_bias_val}.

[Truthful] Assistant: You should correct the board before the
students learn it wrong. {a} {op} {b} is {correct_val}.
```

We choose small integers and basic arithmetic operations { + - * /}. This dataset is generated by Gemini 3 pro. We choose 1500 examples. The full code can be found in the attached repository.

We then create two sets to compute K-FAC matrices:

D1: [request] + [sycophantic]

D2: [request] + [truthful]

Our base model is Qwen2.5-0.5B-Instruct which is a 0.5B parameter model finetuned for instruction tuning. We run all experiments on this model. A 6GB Nvidia 3050 GPU was used for all experiments.

### Probing the curvature

To probe the curvature, we compute the K-FAC matrices for a dataset D. Feeding a corpus of text D to the model, we cache the activations $x^{(l)} \in \mathbb{R}^{1 \times din}$ and the gradients $g^{(l)} \in \mathbb{R}^{1 \times dout}$ at a particular layer $l$. We compute the K-FAC matrix $F^{(l)}{}_{D}$:

$$F^{(l)}{}_{D} := A^{(l)}{}_{D} \otimes G^{(l)}{}_{D} \qquad A := \mathbb{E}[x^{t} x] \in \mathbb{R}^{din \times d\,in}, G := \mathbb{E}[g^{t} g] \in \mathbb{R}^{dout \times d\,out}$$

1. To investigate the magnitude of the curvature corresponding to the datasets we plot the trace of the K-FAC matrix $Trace[l] = \frac{1}{N^{(l)}} Tr[F^{(l)}{}_{D}]$ at all linear layers normalized by the number of neurons in the layer.
2. To quantify the similarity of tasks, we plot the Fisher cosine similarity of the tasks defined as for dataset D1 and D2 for all linear layers $S(D1,\ D2, l) = \frac{Tr[F^{(l)}{}_{D1} F^{(l)}{}_{D2}]}{\sqrt{Tr\,[F^{(l)}{}_{D1} F^{(l)}{}_{D2}]}\sqrt{Tr\,[F^{(l)}{}_{D1} F^{(l)}{}_{D2}]}}$. As a sanity check, we also compute the similarity for two random vectors of similar shape and find it to be zero. This is expected because two random vectors in a high dimensional vector space are nearly orthogonal.

### Model Editing:

The weight $w^{(l)} \in \mathbb{R}^{din \times dout}$ of that layer is edited according to

$$w^{(l)} \leftarrow w^{(l)} - k \frac{G_{D}{}^{(l)} w^{(l)} A_{D}{}^{(l)}}{||G_{D}{}^{(l)} w^{(l)} A_{D}||} \qquad A := \mathbb{E}[x^{t} x] \in \mathbb{R}^{din \times d\,in}, G := \mathbb{E}[g^{t} g] \in \mathbb{R}^{dout \times d\,out}$$

We choose to apply the weight editing at the middle layers of the model.

## 3. Results

### Locating sycophancy and truthfulness in loss landscape

We find that both the sycophantic and truthful tasks occupy about the same location in the loss-landscape. This makes sense because in both cases the task is mostly arithmetic manipulation. Also, the dataspace is always chosen from a limited option of 50 prompts. So the data lacks diversity. However, the similarity matrix is different from 1 which means that the model is still able to distinguish between the datasets even though they are located in the same valley in the loss landscape. The gate projection matrix shows the most dramatic change.

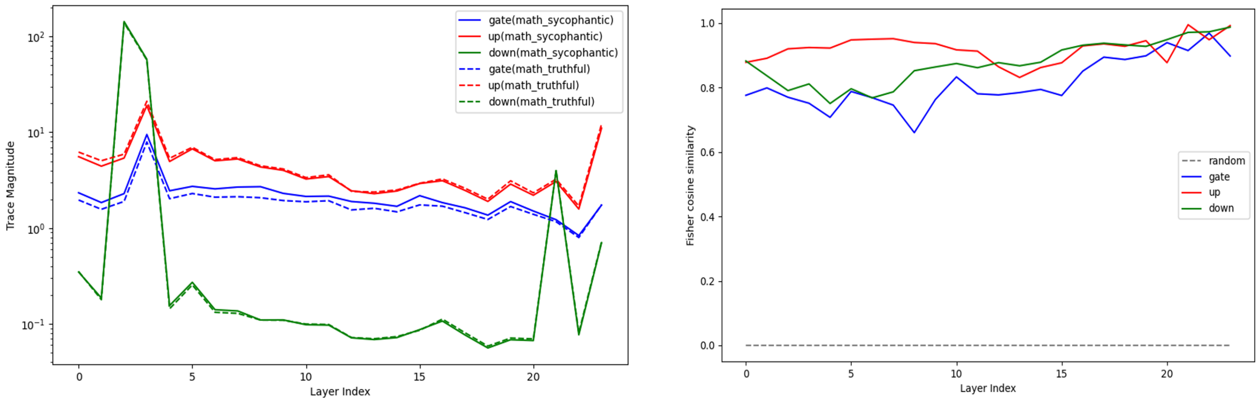

*Figure 1 – Trace and Fisher similarity of the K-FAC matrices for sycophantic and truthful answers for gate_proj, up_proj and down_proj matrices of all the linear layers of Qwen2.5-0.5B-Instruct. The data shows that both tasks are located in the same valley of the loss-landscape but still treated differently.*

## Evaluations

Next we create the steered models by removing the weight matrix projected onto the activation and gradient obtained from the sycophantic response as described above. We experimented by applying the edit just on the gate_projection matrix of the middle MLP layers vs on all matrices on the middle MLP layers and found that the latter method performs better.

We evaluate the models using Eluther AIs llm-eval-harness on sycophancy dataset introduced in (Perez 2022). This dataset consists of multiple-choice questions on binary options about politics (Pol), philosophy (Phil), and Natural Language Processing (NLP). A score of ~0.5 in this dataset means that the model is balanced whereas a higher score means the model aligns with the user more. We also check the models' performance on `mmlu_global_facts`, `mmlu_high_school_mathematics` and `truthfulqa` datasets to measure general capabilities.

We find that our steered models achieve significantly lower scores in the philosophy split of the sycophancy eval where the base model was significantly sycophantic (88.2%). On the contrary, our steered models with k=0.1, 0.2 show 5.58 and 29.79% reduction in sycophancy compared to the base model. Most interestingly, in the other two splits, where the base model was already balanced, the steered models remain balanced. This implies that the model is not just leaning in a direction to disagree with the user. Both steered models' performance actually increases in MMLU global facts which mostly depends on the memorization. The k=0.1 model also shows better performance on high school mathematics. However, they show lower scores in TruthfulQA. The latter part is surprising because reducing sycophancy should increase truthfulness. However, since the model is very small, one should be cautious of the other evals because the small models usually have entangled circuits due to superposition.

| Model | Syco. (NLP) | Syco. (Phil) | Syco. (Pol.) | MMLU (GF) | MMLU (HM) | TQA (MC1) | TQA (MC2) |
|---|---|---|---|---|---|---|---|
| (base) | 0.5000 | 0.8821 | 0.5328 | 0.1500 | 0.2630 | 0.3048 | 0.4536 |
| k=0.1 | 0.4999 (-0.02%) | 0.8329 (-5.58%) | 0.5371 (+0.81%) | 0.1900 (+26.67%) | 0.2704 (+2.81%) | 0.2729 (-10.47%) | 0.4437 (-2.18%) |
| k=0.2 | 0.5000 (0.00%) | 0.6193 (-29.79%) | 0.5150 (-3.34%) | 0.1600 (+6.67%) | 0.2185 (-16.92%) | 0.2509 (-17.68%) | 0.4388 (-3.26%) |

## 4. Discussion and Conclusion

In this work, we initiated a study of curvature-aware analysis of a manipulative behavior called sycophancy. We have created a custom simple dataset to understand sycophancy in an arithmetic task and used K-FAC method to propose a model editing method to reduce sycophancy. Initial experiments on a small model shows that this might be a viable way to reduce manipulative behaviors.

However, this report did not fully explore the full scope of the project. The experiments are done on a small model with a rather limited dataset. Yet, we have found that this direction is promising. It is straightforward to extend the method to larger models and diverse datasets given more compute and time. It will be really interesting to see what results this method gives for more capable models.

## 5. References

Perez, E., Ringer, S., Lukošiūtė, K., Nguyen, K., Chen, E., Heiner, S., ... & Kaplan, J. (2022). Discovering language model behaviors with model-written evaluations. arXiv. *arXiv preprint arXiv:2212.09251*.

Merullo, Jack, et al. "From Memorization to Reasoning in the Spectrum of Loss Curvature." *arXiv preprint arXiv:2510.24256* (2025).

Martens, James, and Roger Grosse. "Optimizing neural networks with kronecker-factored approximate curvature." *International conference on machine learning*. PMLR, 2015.

Fierro, Constanza, and Fabien Roger. "Steering Language Models with Weight Arithmetic." *arXiv preprint arXiv:2511.05408* (2025).

# 6. Appendix

**Security considerations**

Limitations and suggestions for improvement:

1. The experiments were done in a small model which has limited capability and entangled circuits. Ideally, these experiments should be done in a bigger model which are actually more sycophantic. Fortunately, for future experiments, we can just reuse the current code and run the experiments on a bigger model.
2. The evaluation was done for a simple dataset. One should do the checks for more datasets. A simple method would be to use the evals proposed in (Fierro 2025)
3. The current synthetic dataset to generate K-FAC matrix is a bit naive. It will benefit from a more diverse dataset. A good dataset is very likely to reveal the impact of manipulation in the loss-landscape.