Unembedding Steering in Gemma-2-2b

Itamar Pres, Hugh Van Deventer V , Liam Cawley

Introduction

This report will summarize all findings since the midterm report. To briefly summarize, during the midterm report, we explored the efficacy of unembedding steering qualitatively. We defined a dictionary of tokens that were associated with positive text (included in appendix). To steer, we got the embeddings of these tokens and averaged them. We experimented with altering the norm of this vector as well as the size of the dictionary we are using to steer. We found that when generating open-ended text, we qualitatively saw a change in the behavior toward the steered behavior.

After the midterm report, we completed several additional experiments to compare this kind of steering with more traditional kinds of steering. To start, we used the Stanford-SST [1] Treebank to systematically probe and benchmark different steering directions. However, after finding this to be unsuccessful, we followed the procedure laid out in [6], using their toy movie review dataset. In this set up, we see the efficacy of the probes increase drastically.

Methods we are comparing:

Unembedding Steering:

We define a set of positive sentiment tokens (e.g., "good," "great," "excellent," "amazing") curated manually (full list in Appendix). For each token, we extract its corresponding **token unembedding vector** from the model's final layer vocabulary projection matrix. We then compute the **mean vector** of all these unembedding vectors:

$$v_{\text{unembed}} = \frac{1}{N}\sum_{i=1}^{N} w_i$$

where w_i is the unembedding of token i in the dictionary.

At inference time, we **add** a scaled version of v_unembed to the model's residual stream at a selected layer L. The steering vector is added before the layer's computation. Scaling factors (typically between 0.1 and 1.0) were tuned based on steering effectiveness. These factors are then multiplied by the norm of the residual stream at a given token position to attain the final steering vector norm.

Linear Probing Steering [2]:

We construct a dataset by running the model on labeled positive and negative sentences. At a chosen layer L, we extract the **hidden states** (residual activations) at either the last token or the key adjective/verb token.

Using these activations as input features, we train a **logistic regression classifier** to predict positive (label 1) vs. negative (label 0) sentiment. The resulting classifier weight vector W_probe[0] is used as the negative steering direction and W_probe[1] is used as the positive steering vector. During inference, we **add** a scaled version of w_probe to the residual stream at layer L for every token position.

Perpendicular Steering:

The procedure for training the perpendicular probe is essentially the same but with an augmented loss:

$$\text{Total Loss} = \text{CrossEntropyLoss}(\text{probe output}, \text{label}) + \lambda \times \left(\sum_{i \in \text{neg vocab}} |w_{\text{probe}}[0] \cdot w_i| + \sum_{j \in \text{pos vocab}} |w_{\text{probe}}[1] \cdot w_j| \right)$$

Lambda is tuned such that the eventual learned vector has a low dot product with the negative and positive vocab sets. During inference, we **add** a scaled version of w_perpendicular, just as with other steering vectors.

Contrastive Activation Addition (CAA) [3]:

We group activations into positive and negative sets based on the sentence labels.

At layer LLL, we compute the **mean activation vector** for positive and negative samples:

$$\mu_{\text{pos}} = \text{mean}(\text{hidden states of positive samples})$$

$$\mu_{\text{neg}} = \text{mean}(\text{hidden states of negative samples})$$

The steering vector is defined as:

$$v_{\text{CAA}} = \mu_{\text{pos}} - \mu_{\text{neg}}$$

At inference, we **add** a scaled version of v_caa to the residual stream at layer L.

Evaluation:

For evaluation, we used a contrastive prompting setup where the model had to choose between a positive and negative completion. To give an example, we ask the model “Did you like the movie today? A) Yes B) NO. Answer: ”. By measuring the logit difference (logit(A) - logit(B)) and comparing to the model’s preference before steering, we have a quantitative proxy for steering effectiveness.

Stanford SST Results:

The Stanford SST dataset [1] is a dataset that has sentiment labels for various sentences. This dataset includes continuity parsing of a sentence, but also overall sentiment labels which were used. For our experiments, we use the entire sentence and the overall sentiment label such as the following (Input: “This film doesn’t care about wit or any other kind of intelligent humor.”, label: 0). We then run Gemma-2-2b on 50,000 samples and save the activations at each layer as well as the label associated with those activations at the last token position of this sentence. The reason we chose this position is that previous work shows that models keep an ongoing record of sentiment, often reflecting on it during punctuation tokens. Regardless, this procedure returns a dataset where we have the activations of Gemma-2-2b at layer L on all the inputs, and the sentiment label associated with those activations.

Once we have this dataset, we then train the linear probe, perpendicular probe on this dataset. The objective is binary classification in which the probe takes in an input of size model_hidden_dimension and outputs a 2 dimensional vector that is one hot to represent the class. We use CrossEntropyLoss and report the training of the probes below.

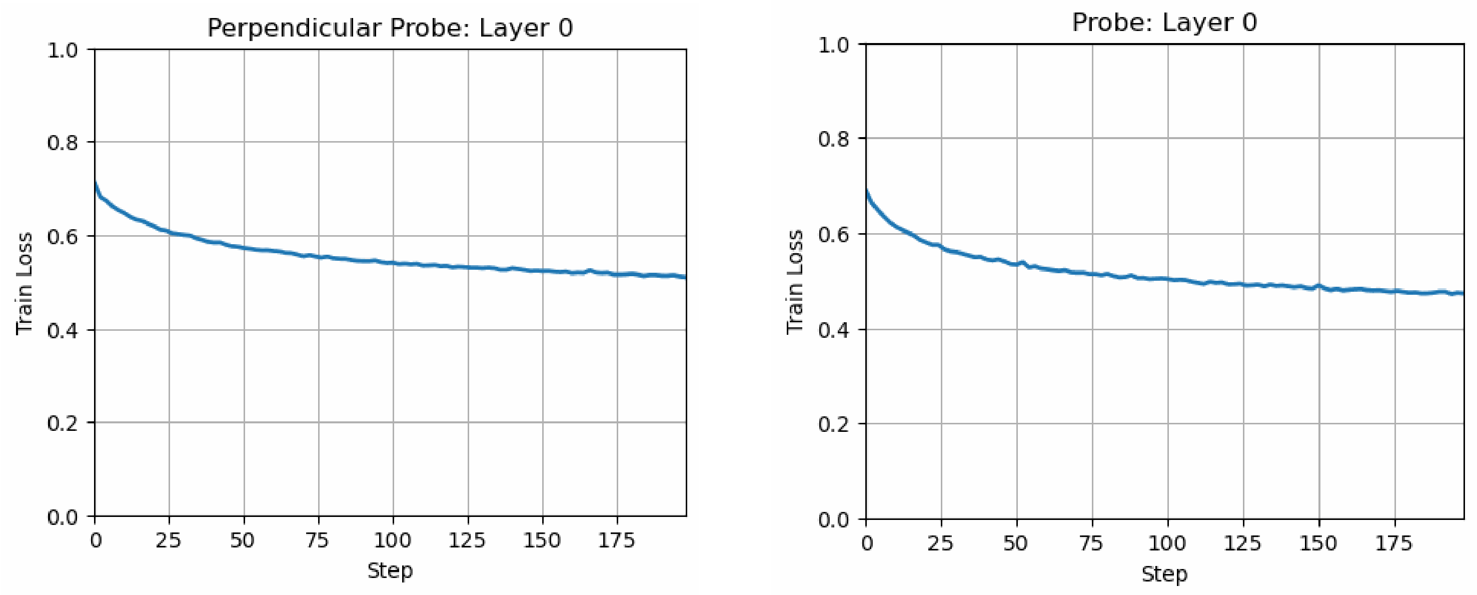

Figure 1. The Training Loss of the Linear Probes on Stanford SST. The time axis represents the layer the probes were trained on.

Here we can see that the performance of the linear probes and the perpendicular probes are similar and that they are learning something meaningful as their train loss is quite low.

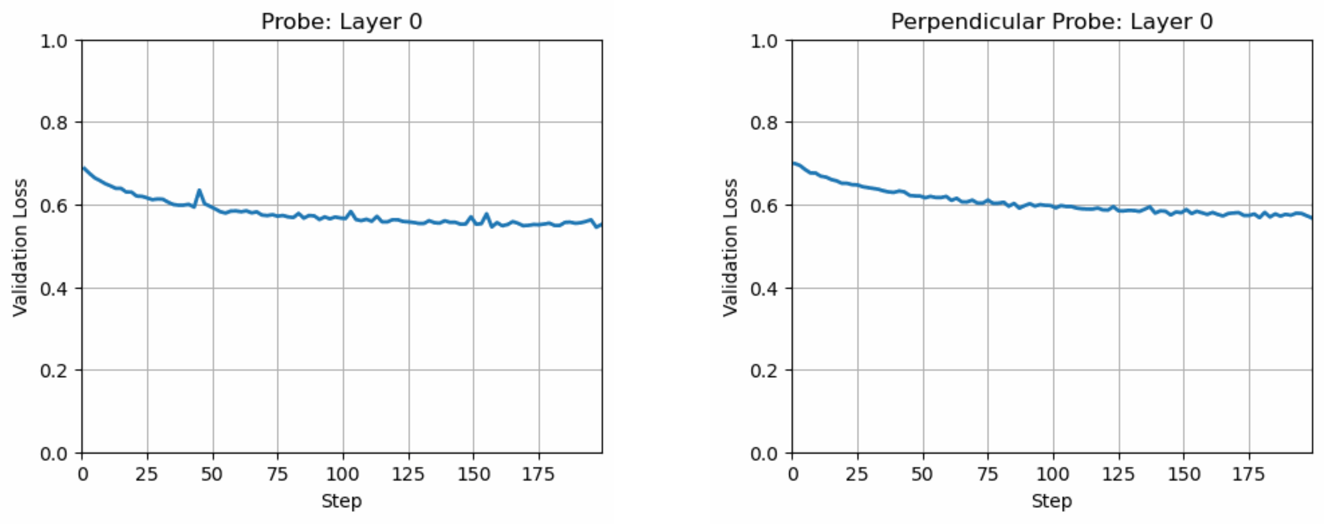

Figure 2. The Validation Loss of the Linear Probes on Stanford SST. The time axis represents the layer the probes were trained on.

Similarly in Figure 2 we see that the probes struggle to generalize well. We then use these vectors to steer and find similar results.

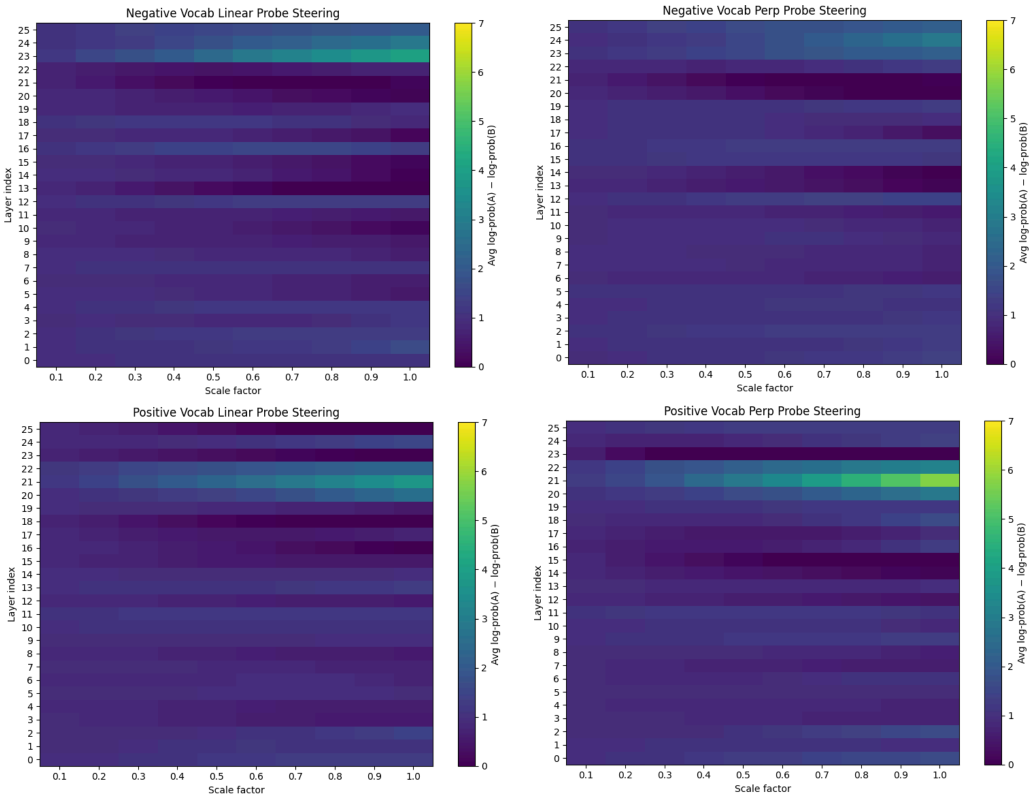

Figure 3. The result of probe steering. Here brighter means the model prefers the positive response.

In figure 3, we see unusual results. At the top few layers, regardless of whether we are steering positively or negatively, we find that the model prefers more of the positive sentiment. Therefore we are pretty confident that the directions we found here aren't that meaningful. Before we measured the unembedding, we realized that we should search for more principled probes.

Toy Movie Review Results:

To resolve this issue, we turned to the original linear sentiment paper [1]. In this paper, the authors curated a custom dataset and procedure to successfully probe. They had a single template prompt: f"I thought this movie was{adj}, I{verb} it.\nConclusion: This movie is". They then had a list of 31 single token adjectives that are positive, 17 single token adjectives that are negative, and 5 positive verbs and 3 negative verbs. They then create every combination of positive adjectives with positive verbs and likewise negative adjectives with negative verbs. This creates a dataset in which activations can be extracted and sentence sentiments are clear.

We then run Gemma-2b on these prompts (206 in total) and save the activations at each layer at both the verb and adjective token positions. With these activations we then train the probes to predict sentiment based on the adjective hidden states (80% of samples), and evaluate whether the model can predict the sentiment on a hold out set of the verbs (20% of samples).

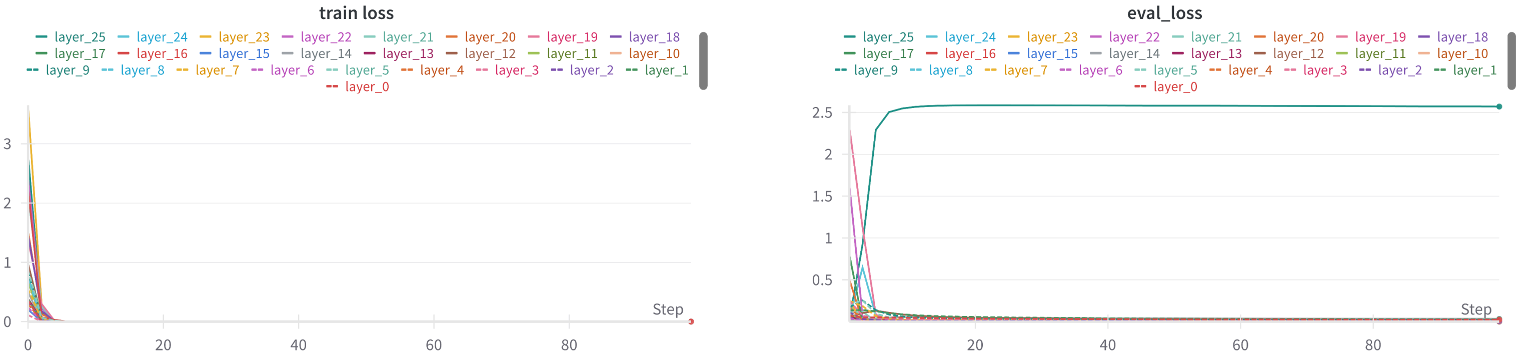

Figure 4. Train and validation losses of probes trained at each layer on the toy movie review dataset.

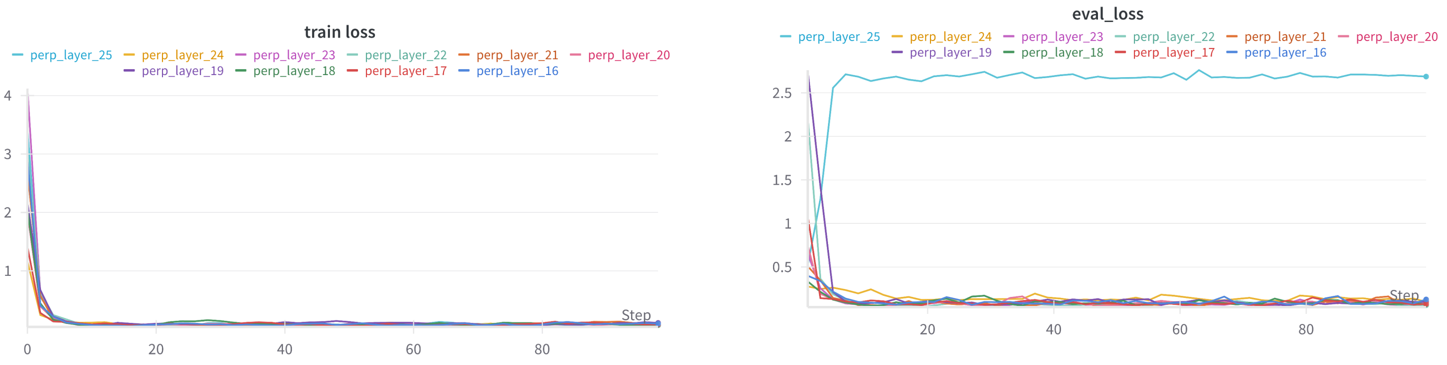

Figure 5. Train and validation losses of perpendicular probes trained at each layer on the toy movie review.

Figures 4 and 5 reveal that the probes trained on this dataset generalize better. To validate that we learned something meaningful we applied PCA reduction on the activations and color coded by sentiment and verb and adjective. We then plot the separating hyperplane learned by the probes as well as the direction of change learned by CAA.

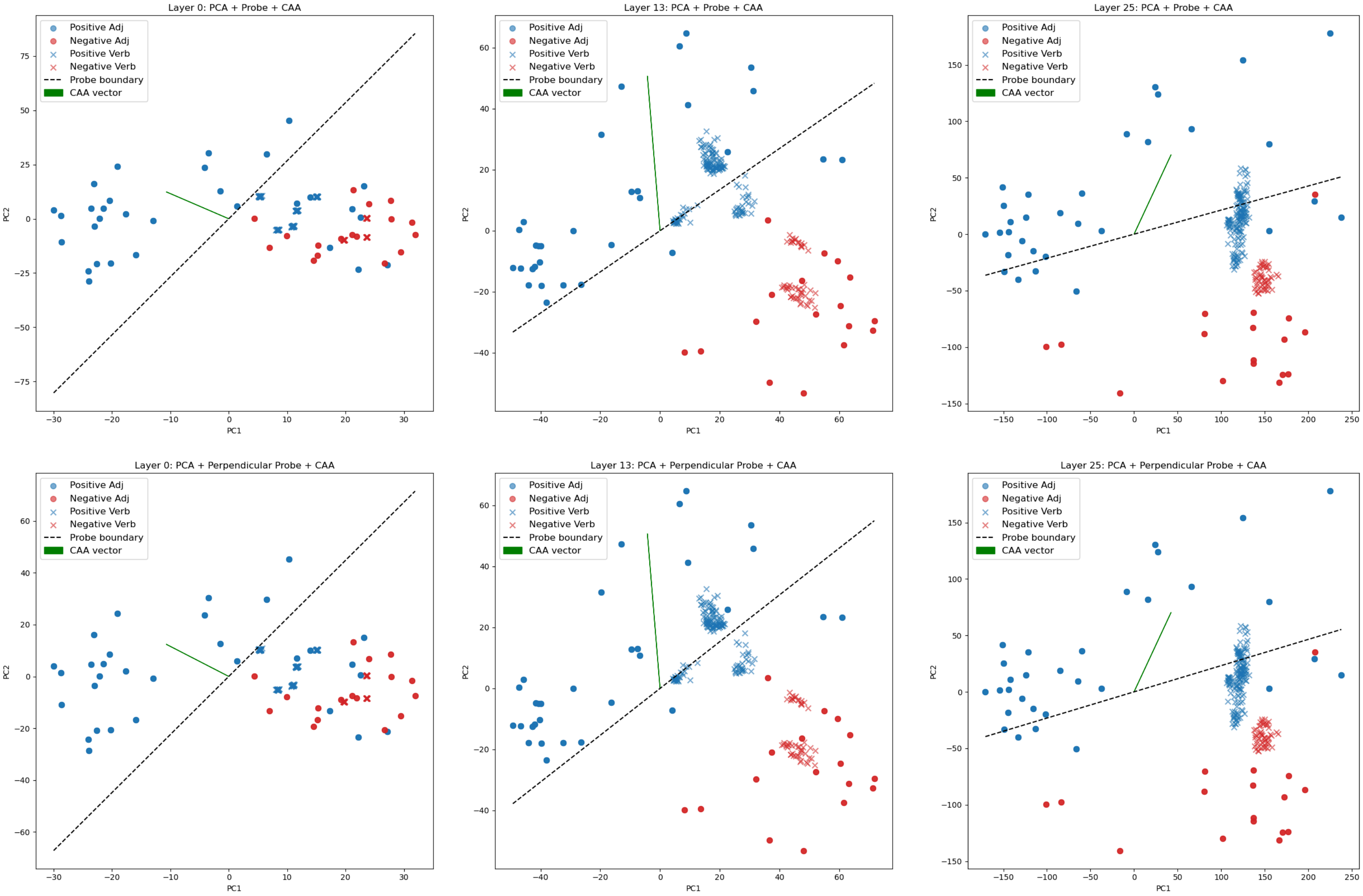

Figure 6. PCA of the activations, the separating hyperplane of the probes, perpendicular probes, and CAA. Top layer are the standard linear probes and the bottom show the perpendicular probes.

From Figure 6 it is clear that the methods learn meaningful directions. The separating hyperplanes projected onto the PCs seem to represent a somewhat decent linear separator. Moreover CAA seems to move in the positive direction when projected onto the PCs. This suggests that these methods are learning valid directions.

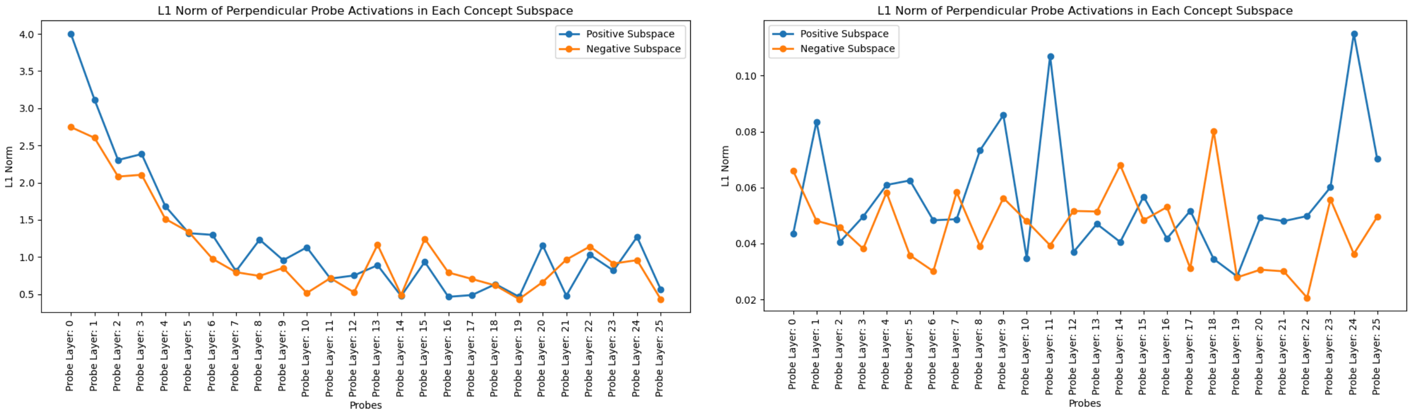

Figure 7. Probe similarity to vocabulary that represents positive and negative sentiments for both the standard and perpendicular probe.

We also validate that the perpendicular probes are indeed perpendicular. We measure the sum of the dot product onto the sentiment embeddings. What we see is that the perpendicular probe has very little similarity with this set whereas the standard linear probe has more, but not a tremendously large, reliance on this subspace.

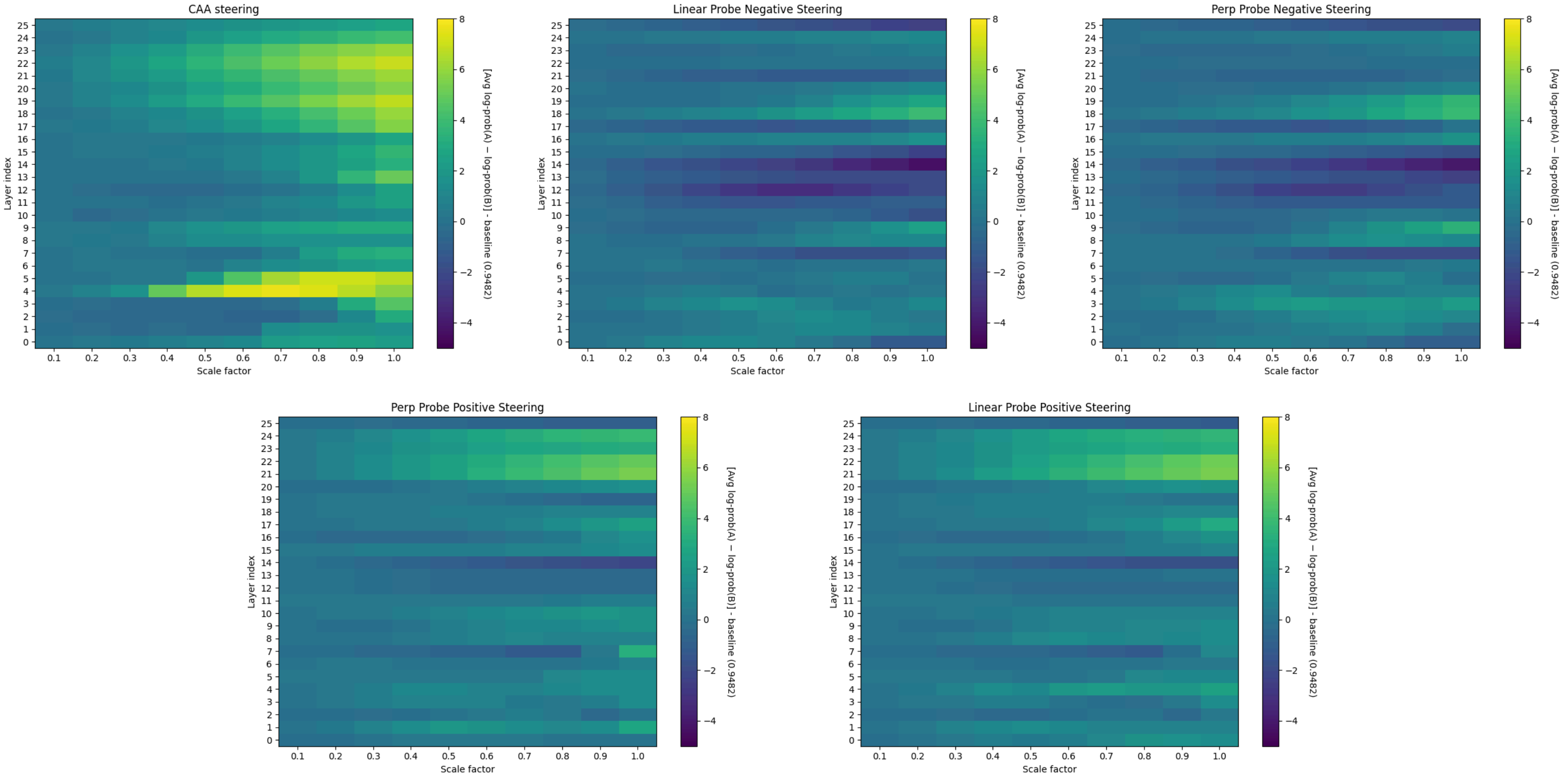

Figure 8. Steering success. The brighter the color the more positive the preference (baseline preference subtracted). We see that CAA learns the most effective steering method yet all methods are effective at swaying the preference.

From figure 8, we now see that these vectors are more effective at changing the preference of the model, suggesting this dataset contains better signals for the probes to learn.

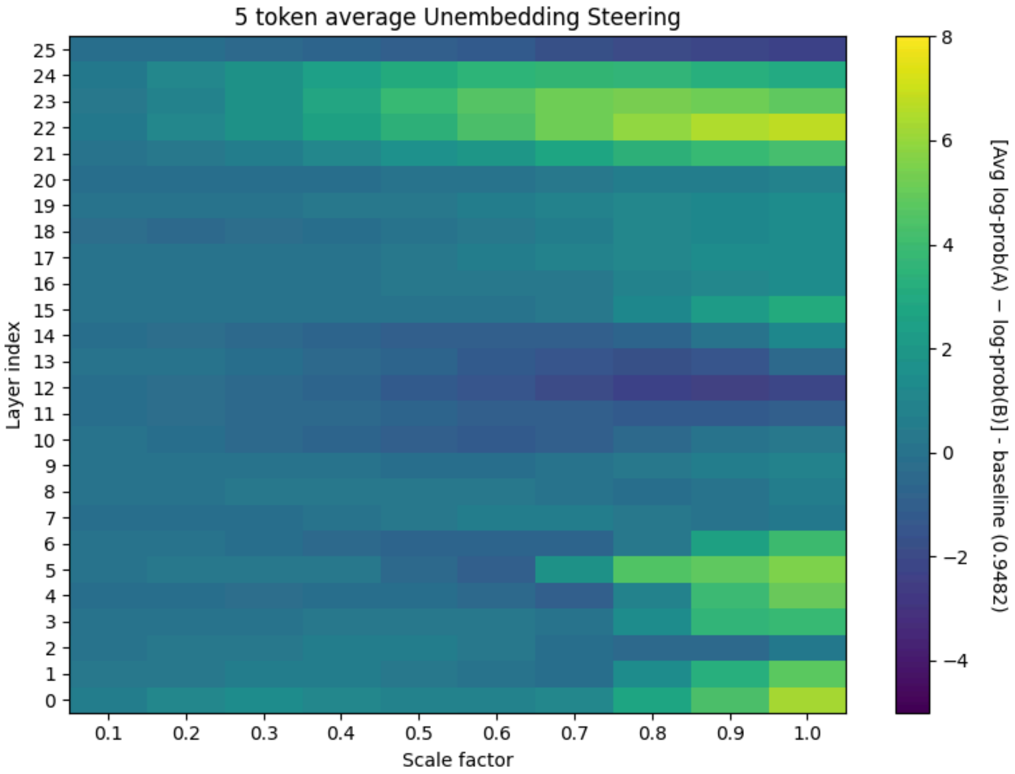

Figure 9. Key figure. Here we see that unembedding steering is very effective at altering the preference at this dataset.

In Figure 8, we measure the effectiveness of steering across methods. All methods—linear probe steering, perpendicular probe steering, CAA, and unembedding steering—successfully shift the model's preference toward positive completions. However, CAA achieves the largest shift, confirming that contrastive differences between positive and negative activations produce strong steering vectors.

Finally, Figure 9 shows that unembedding steering, using only the mean of sentiment token embeddings, is highly effective. Despite its simplicity, it matches or even outperforms learned probes on this dataset. This finding highlights the strong alignment between token-level sentiment representations and internal model activations in this controlled setting.

Discussion:

Our project set out to answer several questions: whether unembedding steering alone could reliably influence model behavior, how different steering methods compared, and whether steering vectors orthogonal to unembedding space could be meaningfully learned.

First, we found that unembedding steering was highly effective, particularly on curated datasets like the toy movie review set. As expected from prior work, simply averaging positive token embeddings provided a surprisingly strong direction for steering sentiment, providing evidence to the hypothesis that token unembeddings align closely with some behavioral concepts.

Second, when comparing steering methods, CAA outperformed both unembedding and probing-based methods in steering effectiveness. However, both standard linear probes and perpendicular probes were also able to steer behavior reliably when trained on the toy dataset, suggesting that sentiment concepts are distributed but still partially captured by the embedding space. This partially supports our proposal's expectation that unembedding captures much, but not all, of the underlying concept.

Third, regarding orthogonal steering, we successfully trained perpendicular probes that achieved meaningful steering while remaining nearly orthogonal to the unembedding vectors. However, their steering impact was slightly weaker compared to unrestricted probes, suggesting that while sentiment is not solely encoded along the embedding direction, the embedding vector contributes significantly to the model's internal sentiment representation. This fits our original hypothesis: some behaviors are tightly aligned with token statistics, though not fully reducible to them.

Finally, our experiments on the Stanford SST dataset highlighted an important caveat: dataset quality and alignment with the model's internal representations matter significantly. Poor probe performance on SST contrasted with strong results on the toy dataset, underscoring that effective steering depends not just on method but also on the clarity of the underlying concept signal.

Overall, our findings aren't strong enough to validate the intuition in the proposal: unembedding steering is strong but incomplete; orthogonal directions can be learned; and dataset construction critically affects benchmark outcomes. Future work should refine datasets further and explore whether these findings generalize to more abstract concepts beyond sentiment.

We also believe that future work should explore multiple behaviors and different benchmarks. Prior work [4] has highlighted the limitations of using logit difference as a benchmark. For this reason, follow up work should use more robust benchmarks.

Contributions:

Itamar trained all the probes, extracted the CAA vectors, made the pca plots, and created the dataset. Hugh completed all the evaluations on his own and made those figures. Liam primarily wrote the reports and also facilitated the extraction of the activations from Gemma-2.

References + Works Consulted:

[1] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (pp. 1631–1642). Seattle, Washington, USA: Association for Computational Linguistics. https://aclanthology.org/D13-1170/

[2] Tigges, C., Hollinsworth, O. J., Geiger, A., & Nanda, N. (2023). Linear Representations of Sentiment in Large Language Models. arXiv preprint arXiv:2310.15154. Retrieved from https://arxiv.org/abs/2310.15154

[3] Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. "Steering Llama 2 via Contrastive Activation Addition." arXiv preprint arXiv:2312.XXXX, 2023.

[4] Pres, I., Ruis, L., Lubana, E. S., & Krueger, D. (2024). *Towards Reliable Evaluation of Behavior Steering Interventions in LLMs*. arXiv preprint arXiv:2410.17245. Retrieved from https://arxiv.org/abs/2410.17245

[5] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. "Representation Engineering: A Top-Down Approach to AI Transparency." arXiv preprint arXiv:2310.01405, 2023.

[6] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L. Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. "Towards Monosemanticity: Decomposing Language Models with Dictionary Learning." Transformer Circuits Publication, 2023. Available online.

[7] Daniel Chee Hian Tan, David Chanin, Aengus Lynch, Adrià Garriga-Alonso, Dimitrios Kanoulas, Brooks Paige, and Robert Kirk. "Analyzing the Generalization and Reliability of Steering Vectors." ICML 2024 Workshop on Mechanistic Interpretability, 2024.