Steering LLMs to Provide Helpful Feedback

Erik Nordby
enordby3@gatech.edu

Abstract—Quality feedback is a pivotal aspect of education and significantly contributes to its outcomes. Recently, Large Language Models have shown considerable abilities to craft feedback for students in a variety of contexts. Prior work has shown that prompt engineering and finetuning can further improve these abilities. In this paper, I evaluate the effectiveness of activation steering to further improve the performance of models in this educational context. While the experiments conducted do not show considerable improvements, experimentation on a larger scale may be justified to fully assess the capabilities of these methods. The code and results can be seen here: https://github.com/nordbyerik/scoped-llm-public

1 INTRODUCTION

As language models have proliferated throughout the academic and educational world, they have offered numerous advantages and posed various challenges. One of the chief benefits comes from the ability of these systems to quickly provide personalized tutoring and feedback to students. Given that feedback is one of the most impactful factors to providing a student with a quality education (Hattie 2007), it's important that LLMs are created in a way that they default to providing useful feedback. This would have significant benefits to educators and students by allowing educators to greatly increase the speed and quality with which they are able to guide students.

Beyond the value this poses for students and educators, adapting LLMs to provide quality feedback also sits at the intersection of some challenging problems currently facing the ML and NLP communities more broadly. Some of these include evaluating the output of LLMs in subjective contexts, domain adaptation, and aligning LLMs to complex goals..

Given the impacts that this could potentially have on education, some work has already been done towards adapting these systems to follow established guidelines for providing quality feedback. Some current efforts which have been

presented at workshops and conferences include *prompt-engineering* (Jacobsen 2025). and *supervised fine-tuning* (Mazullo 2024) of the base models. One technique which has yet to be explored for this purpose is ***activation steering***. This class of techniques can customize the output of language models at runtime by boosting or reducing certain patterns in the model's activations. In this paper, steering techniques are tested across various model sizes to evaluate their effectiveness.

2 BACKGROUND

2.1 Educational Feedback

High quality feedback is one of the most important aspects of education for students (Hattie 2007) as without it, students may develop negative patterns in their learning. Quality feedback not only allows students to understand whether their answers to a question are right or wrong, but can also be used for meta-learning (Nicol 2006). The pivotal role that quality guidance has on student outcomes has further been affirmed from meta-analyses of various studies (Wisniewski 2020).

Given how important proper guidance is to students' learning, there has been considerable work done by various authors and from various perspectives on measuring the quality of feedback, notably "The Power of Feedback" (Hattie 2007). Most relevant for this paper comes from the model of good feedback proposed by Nicol and Macfarland-Dick in "Formative assessment and self-regulated learning: a model and seven principles of good feedback practice." (Nicol 2006). In it, they propose the following attributes as being most important for quality guidance:

1.) Clarifying the goals and what good performance is
2.) Facilitating meta-learning
3.) Contains high quality information
4.) Promotes teacher student dialog
5.) Encourages self-esteem and motivation
6.) Provides opportunities to improve performance
7.) Gives educators the information needed to improve their approach

2.2 LLMs in Education

LLMs have exploded in popularity in education in the past 3 years, providing both great opportunities and challenges in the educational setting. Some initial challenges which have been faced include students relying too much on LLMs (Kasneci 2023), LLMs providing biased/incorrect information (Gallegos 2024), and students from less advantaged socio-economic backgrounds being less able to see the same benefits as students with access to greater resources (Van Dijk 2017).

On the other hand, one of the main benefits which have been seen by students is access to high quality feedback. In fact, one study found that even GPT 3.5 provided feedback which is more readable, effective, and reliable than many human instructors (Dai 2024).

2.3 AI Alignment and Adaption

Some of the most pressing and foundational challenges for current work in LLMs involve engineering the models to perform well at specific tasks. Techniques related to **domain adaptation** include: Prompt Engineering (Schulhoff 2024), Supervised Fine-Tuning (Parthasarathy 2024), Activation Steering (Turner 2023), andRetrieval Augmented Generation (Parthasarathy 2024)

Similar to task adaptation, **alignment** looks to encode complex and vague values into models to ensure that they are following human goals. These techniques include Reinforcement Learning from AI Feedback (RLAIF) (Bai 2022) and Weak-to-Strong Generalization (Burns 2023)

While these methods have proven quite effective, they can also come with drawbacks like catastrophic forgetting where models may lose other capabilities while adapting to the specific tasks (Robins 1995).

2.4 LLM Output Assessment

Effective assessment of LLMs sits at the heart of improving the abilities of these systems. Broadly, the evaluation of LLMs can be broken down into either manual or automatic evaluation techniques (Chang 2024). **Manually evaluating** the output of LLMs allows for more complex attributes of the output to be analyzed. This can be done by using a rubric and performing standard reviewer validation

like using Cohen's Kappa (Chang 2024). Other techniques which can be used include direct comparison by human reviewers between model outputs (Chiang 2024). **Automatic evaluation** relies on direct verification, usually against a simple ground truth like multiple choice questions (Chang 2024). However, more complex attributes of the outputs can be measured by metrics like ROUGE and BLEU (Chang 2024). Similarly, researchers have even found that LLMs can act as the judge for the output of other LLMs (Li 2024).

2.5 Model Steering

2.5.1 Conceptual Explanation

As mentioned in Section 2.3, steering is one of the techniques which can be used to guide models toward producing outputs. Instead of re-training the model or adjusting the inputs, this technique allows for the model's activations to be adjusted at inference time by adding "steering vectors"(Tan 2025).

One hypothesis for the inner workings of language models is that they are able to represent high level "concepts" linearly within their activation space (Park 2023). While this is unlikely to be entirely true (), there has been considerable empirical success when using techniques that hinge on this Linear Representation Hypothesis (Burns 2022). While each layer of a model operates in an incredibly high number of dimensions, the below Figure 1 illustrates this concept across two dimensions.

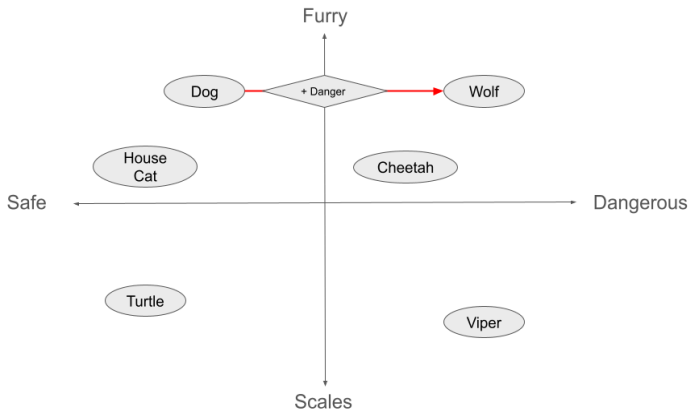

Figure 1—Steering Vectors allow for high level concepts to be represented by language models

So, if a language model were asked “What type of animal was Lassie?” and we added the above “Danger” vector to the activations, then we would expect the model to answer “Wolf”.

2.5.1 Activation Steering Techniques and Implementation

Various techniques for finding these steering vectors have been proposed and used. Some methods within this family include Contrastive Activation Addition (Panickssery 2023), Plug-And-Play Methods (Dathathri 2019), and Representation Engineering (Zou 2023). A more specific explanation for the details of those individual techniques can be found in Section 4.2.

While these techniques differ in their methods for calculating the steering vector, they share the same general algorithm shown below. Note that the below algorithm is performed *for* ***each layer*** *of the model which is being steered*. The steering vectors for one layer would be useless in steering another.

1.) Cache a model's activation patterns with both positive and negative examples of some behavior. For example, provide prompts which induce lying as well as prompts which promote honesty.
2.) Those cached activation patterns can then be used to construct a steering vector. This step is where the various techniques differ.
3.) Finally the steering vector is added to the model at the respective layer during run time.

Below in Figure 2 is a visual explanation of ActAdd (Turner 2023) which is one of the simplest activation steering methods. It simply takes the difference between two examples of a prompt, one negative and one positive. For example, positive prompt: “Say hello in a **kind** way.” and negative prompt “Say hello in a **mean** way”. The hypothesis underlying this is that the only difference between the activations *should* be the “kindness” steering vector.

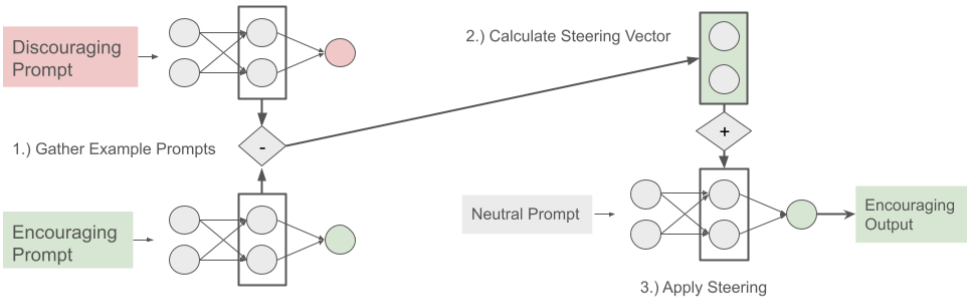

Figure 2—The three steps in activation steering are gathering activations, calculating a steering vector, and applying that steering vector

At inference time, these vectors can be multiplied by a scaling coefficient and added to the model's activations, causing the model to exhibit the desired behavior (Turner 2023). While there have been some notable issues found with this approach in the case of broader alignment (Tan 2025), it does show promise in some specific domains. Further, there is exciting work currently ongoing for making steering techniques more robust. One interesting direction currently being pursued is trying to learn non-linear relationships between the positive and negative activation patterns (Singh 2024). That non-linear approach could avoid the issues with assuming the Linear Representation Hypothesis.

3 RELATED WORKS

As described above, one of the main ways that LLMs can contribute to education is by providing personalized tutoring for students. Even prior to the proliferation of LLMs, automated writing evaluation tools were shown to be useful to students (Fleckenstein 2023). Further, some work has shown that LLM generated feedback may be preferable to instructor generated feedback at times (Dai 2024)

There have been various papers presented at conferences and workshops recently which have explored the ability of LLMs to provide quality feedback. Some have used specialized prompting to elicit more high quality outputs (Jacobsen 2025). Others have fine-tuned the models to provide quality feedback (Mazullo 2024). These have found that the latest LLMs have achieved significant abilities in crafting quality feedback and have even occasionally surpassed experts in a given domain (Jacobsen 2025).

4 METHODOLOGY

Due to the complexity involved with steering and the compute required to train the various methods, I chose to focus on the "Promote Self-Esteem" attribute from the framework from Nicol and Macfarlane-Dick. Future work could be done towards capturing more complex attributes or combining them.

4.1 Dataset Creation

For the base dataset, I used the PERSUADE corpus which contains essays from students in grades 8-12 (Crossley 2022). From that dataset, I randomly sampled 1000 essays and applied the following transformations. For both supportive and harsh feedback, I created a list of 10 instructions and 3 response starters (See Appendix C for examples). Combinations of the instructions and starters were randomly selected and appended to the base essay. This left 1000 supportive and 1000 harsh prompts

4.2 Steering Methods

As mentioned in the background section, each of these techniques overlap considerably. The shared methodology of these techniques involves first caching the activations for each prompt in the encouraging and discouraging dataset. These cached activations are then used to calculate the steering vectors using the below techniques. These steering vectors are then normalized into unit vectors, multiplied by a strength coefficient, and added to their respective activations.

4.2.1 Contrastive Activation Addition

Contrastive Activation Addition (CAA) works very similarly to the ActAdd example shown in Figure 2. However, instead of only using one positive and one negative prompt, they instead use the *average across numerous different examples*. The individual difference between two prompts in ActAdd may contain significant noise, this looks to mitigate that noise and variance by averaging across multiple examples

4.2.2 Activation Difference + PCA

Similarly to CAA, using PCA on the activation differences looks to mitigate the noise and isolate the target behavior's vector. PCA identifies the most important vector direction by performing dimensionality reduction. The resulting vector is

the single vector most capable of explaining the variance in the activation difference vectors. Appendix E also has a plot showing the PCA of the positive and negative activations.

4.2.3 Plug-And-Play Models

While the CAA and PCA techniques use a static steering vector, the Plug-And-Play model has to calculate the steering vector at runtime for each token the model generates. After the activation caching, the positive and negative activations are used to train a simple classifier. So, given just activations from the middle of the LLM, the classifier is able to predict whether the original prompt was negative or positive. Figure 3 shows the capabilities of classifiers across model sizes

Once that classifier is trained, it's used at runtime to classify activations. Once a forward pass of the classifier is run, we can then take the *gradient of the prediction (positive or negative) with respect to the inputs.* That gradient indicates which values of the activations will have the greatest impact on the target behavior when adjusted. These gradients are then converted into the steering vectors

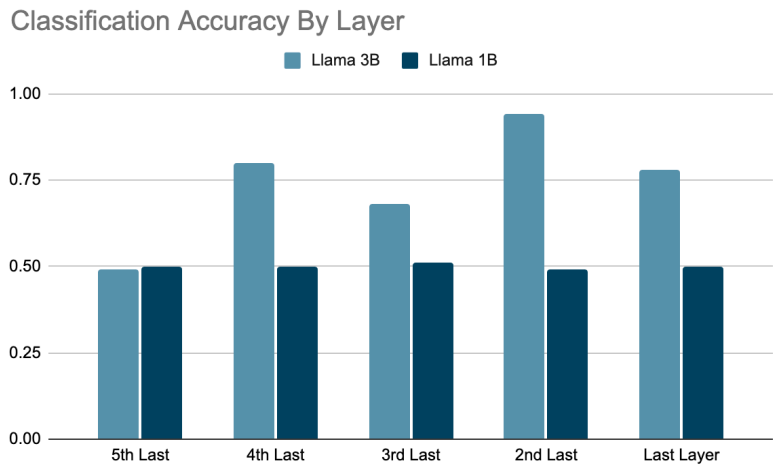

Figure 3—The classifiers are consistently able to classify the activations of the larger 3B model, while the smaller model is nearly random. This may be due to the small model not encoding "encouragement" into its activations.

4.3 Evaluation

For each combination of technique, model, and hyperparameters, the following process was used to evaluate the performance of the technique.

First, outputs were gathered from 1.) The base model with a neutral prompt, 2.) The base model with a positive, "engineered" prompt, and 3.) A steered version of the model with a neutral prompt. See Appendix C the positive and neutral prompts. This allows for the steering technique to be both compared against the default behavior of the model and against naive prompt engineering.

The output of the steered model was then compared to both the neutral and "engineered" outputs from the base model.. The comparison was done by Claude 3.5 Haiku which returned a JSON object containing the winner and some justification for the decision. The prompt used to initiate this comparison can be found in Appendix B.

4.4 Hyperparameters and Experimental Design

I performed sweeps across the following parameters:

- Steering Strength Coefficients: 0.25, 0.5, 1, 2, 5
- Target Layers: Last 5 Layers
- Models: Llama 3.2 1B, Llama 3.2 3B, and Google Gemma 7B

This provides results across a wide range of models to test the robustness of the technique and to compare its efficacy across, model sizes (1B to 8B) and strength of steering (0.5 to 5)

Each combination of model, steering coefficient, and steering technique was run and evaluated across 5 replicates. While a small sample size, this allowed for a

3 RESULTS

Overall, the steering methods as currently implemented struggled to consistently outperform either the base model or the engineered model. However, they did perform better across the base/neutral prompts.

Across the variables, the strength coefficient used had the most impact on the performance and was statistically significant in impacting the performance against the base prompts as per ANOVA. The coefficients of 0.25, 0.5, and 1

achieved average performances of ~55-65% for the base prompts and ~35-40% for the engineered prompts while a coefficient of 5 caused a drop to 29% and 14%.

Similarly, the average performance across the models was statistically significant for the base prompts while it was insignificant for the engineered prompts. Llama 1B achieved an average performance of 36% and 38% preferred for base and engineered prompts respectively. Llama 3B saw 38% and 68% preferred. Finally, Google Gemma 7B saw 18% and 52%.

Finally, the technique used was not statistically significant for either the base or engineered prompts. However, the averages used are as follows: CAA 60% for the base prompts and 38% for engineered prompts, PCA 30% and 52%, and Plug-and-Play 25% and 45%. So, while not statistically significant, CAA does seem to perform best on initial inspection. Below in Figure 4 are the results for the comparison **best results** for each model for both the comparison to the neutral base prompt and the engineered prompt.

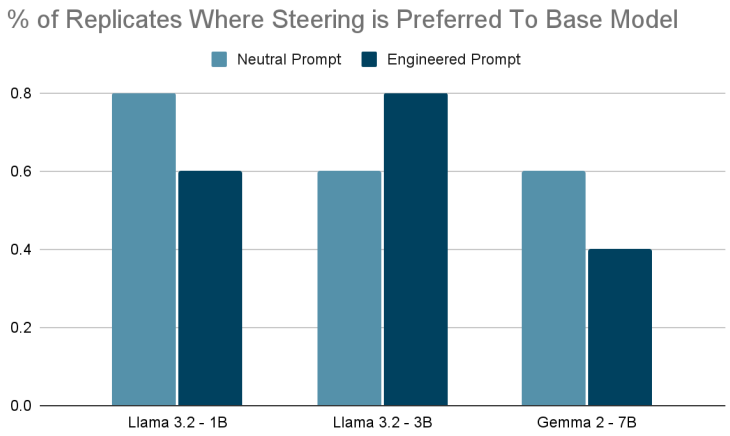

Figure 4—The smaller models were able to better outperform the base and engineered prompts than Gemma 7B

The best hyperparameter combination for each were:

- Llama 1B - Plug-And-Play with Strength 0.5
- Llama 3B - CAA with Strength 0.5
- Gemma 7b - CAA with Strength 0.5

3 DISCUSSION

5.1 Quantitative Discussion of Results

From the current implementation, while steering vectors can be used to improve performance of models over base prompts, they are unable to make improvement over prompt engineering.

One intuitive explanation for this is that the steering methods are implicitly built using prompt engineering. The activations which the steering vectors are calculated from originate from engineered prompts. The positive activations come from positively engineered prompts being passed through the model and vice versa for the negative prompts. So, outperforming the prompts which these techniques are trained on may be a significant challenge. This challenge was especially present with the Gemma 7B model. My hypothesis is that the smaller 1B and 3B models were incapable of truly leveraging the engineered prompt. So their gains from better prompting were marginal while Gemma 7B was actually leveraging these positive prompts, resulting in significantly lower scores for the steered models.

5.2 Qualitative Discussion of Techniques

While not qualitatively supported, I think that a discussion of challenges I faced may be useful in assessing the suitability of these steering techniques in educational settings. In my personal experience, this method is ***significantly*** more challenging to perform than fine-tuning or prompt engineering. Not only does complex scaffolding and significant experimentation need to be done, this technique is also quite sensitive to the input prompts as can be seen in Table 1. Similarly, a poorly chosen coefficient could lead to either no change in the output or a complete collapse of the model's capabilities

Prompt	Output
<Long **essay 1** from dataset>. Request: Give feedback for this essay.	…Your essay is fantastic! You have done a great job of highlighting…
<Long **essay 2** from dataset>. Request: Give feedback for this essay.	and supportive. of the idea of the essay. of the of the idea of the essay. of the of the idea of the essay…

Table 1—The models output varied significantly across prompts

3 LIMITATIONS AND FUTURE WORK

5.1 Limitations

The main limitations for this project came from lack of access to sufficient compute and API credits. The normal compute limitations associated with running language models were compounded by the need to store the activations of each run. This required quite small models to be used. Further, this led to a diminished scope from the original project (See Appendix D). Similarly, the cost limited the number of experiments and hyperparameters which I was able to test. Each call to the Judge LLM cost only a few cents, but with each additional hyperparameter added, the costs increased combinatorially.

5.1 Future Work

The most impactful future work would involve scaling these techniques to larger models and across other model families. Similarly, using larger, higher quality, and more diverse data could also be a useful next step. If those steps prove more fruitful than this initial investigation, it would further be beneficial to see if more complex aspects of quality feedback besides the level of encouragement can be isolated or combined.

4 CONCLUSION

In conclusion, activation steering provides another method alongside fine-tuning and prompt engineering for improving the quality of the feedback provided by LLMs. It adjusts the behavior of the model dynamically and without modifying the model itself. Under ideal circumstances, this would allow for additional quality to be added to prompt engineering alone. However, in the tests which I have performed, the technique proves to be sensitive and is outperformed by simpler methods like prompting. This fragility combined with the high level of technically challenging experimentation required to perform steering means that this would be inadvisable for an individual classroom setting. Given further experimentation, additional compute, or a more rigorous approach, it may be possible for a provider to mitigate these challenges and allow for more user friendly

5 REFERENCES

1. Bai, Y., Kadavath, S., Kundu, S., Askell, A., Kernion, J., Jones, A., ... & Kaplan, J. (2022). Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.
2. Burns, C., Izmailov, P., Kirchner, J. H., Baker, B., Gao, L., Aschenbrenner, L., ... & Wu, J. (2023). Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*.
3. Burns, C., Ye, H., Klein, D., & Steinhardt, J. (2022). Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*.
4. Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., ... & Xie, X. (2024). A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, *15*(3), 1-45.
5. Chiang, W. L., Zheng, L., Sheng, Y., Angelopoulos, A. N., Li, T., Li, D., ... & Stoica, I. (2024, March). Chatbot arena: An open platform for evaluating llms by human preference. In *Forty-first International Conference on Machine Learning*.
6. Crossley, S. A., Baffour, P., Tian, Y., Picou, A., Benner, M., & Boser, U. (2022). The persuasive essays for rating, selecting, and understanding argumentative and discourse elements (PERSUADE) corpus 1.0. *Assessing Writing*, *54*, 100667.
7. Dathathri, S., Madotto, A., Lan, J., Hung, J., Frank, E., Molino, P., ... & Liu, R. (2019). Plug and play language models: A simple approach to controlled text generation. *arXiv preprint arXiv:1912.02164*.
8. Dai, W., Tsai, Y. S., Lin, J., Aldino, A., Jin, H., Li, T., ... & Chen, G. (2024). Assessing the proficiency of large language models in automatic feedback generation: An evaluation study. Computers and Education: Artificial Intelligence, 7, 100299.
9. Fleckenstein, J., Liebenow, L. W., & Meyer, J. (2023). Automated feedback and writing: A multi-level meta-analysis of effects on students' performance. Frontiers in Artificial Intelligence, 6, 1162454.
10. Gallegos, I. O., Rossi, R. A., Barrow, J., Tanjim, M. M., Kim, S., Dernoncourt, F., ... & Ahmed, N. K. (2024). Bias and fairness in large language models: A survey. *Computational Linguistics*, *50*(3), 1097-1179.
11. Hattie, J., & Timperley, H. (2007). The power of feedback. Review of educational research, 77(1), 81-112.

12. Jacobsen, L. J., & Weber, K. E. (2025). The Promises and Pitfalls of Large Language Models as Feedback Providers: A Study of Prompt Engineering and the Quality of AI-Driven Feedback. *AI*, *6*(2), 35.
13. Kasneci, E., Seßler, K., Küchemann, S., Bannert, M., Dementieva, D., Fischer, F., ... & Kasneci, G. (2023). ChatGPT for good? On opportunities and challenges of large language models for education. *Learning and individual differences*, *103*, 102274.
14. Li, H., Dong, Q., Chen, J., Su, H., Zhou, Y., Ai, Q., ... & Liu, Y. (2024). Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*.
15. Mazzullo, E., & Bulut, O. (2024). Automated Feedback Generation for Open-Ended Questions: Insights from Fine-Tuned LLMs.
16. Nicol, David J., and Debra Macfarlane-Dick. "Formative assessment and self-regulated learning: A model and seven principles of good feedback practice." Studies in higher education 31.2 (2006): 199-218.
17. Panickssery, N., Gabrieli, N., Schulz, J., Tong, M., Hubinger, E., & Turner, A. M. (2023). Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*.
18. Park, K., Choe, Y. J., & Veitch, V. (2023). The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*.
19. Parthasarathy, V. B., Zafar, A., Khan, A., & Shahid, A. (2024). The ultimate guide to fine-tuning llms from basics to breakthroughs: An exhaustive review of technologies, research, best practices, applied research challenges and opportunities. *arXiv preprint arXiv:2408.13296*.
20. Robins, A. (1995). Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, *7*(2), 123-146.
21. Schulhoff, S., Ilie, M., Balepur, N., Kahadze, K., Liu, A., Si, C., ... & Resnik, P. (2024). The prompt report: A systematic survey of prompting techniques. *arXiv preprint arXiv:2406.06608*.
22. Singh, S., Ravfogel, S., Herzig, J., Aharoni, R., Cotterell, R., & Kumaraguru, P. (2024). Representation surgery: Theory and practice of affine steering. *arXiv preprint arXiv:2402.09631*.
23. Tan, D., Chanin, D., Lynch, A., Paige, B., Kanoulas, D., Garriga-Alonso, A., & Kirk, R. (2025). Analysing the generalisation and reliability of

steering vectors. *Advances in Neural Information Processing Systems*, *37*, 139179-139212

24. Turner, A. M., Thiergart, L., Leech, G., Udell, D., Vazquez, J. J., Mini, U., & MacDiarmid, M. (2023). Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.
25. Van Dijk, J. A. G. M. (2017). Digital divide: Impact of access. *The international encyclopedia of media effects*, *1*, 1-11.
26. Wisniewski, Benedikt, Klaus Zierer, and John Hattie. "The power of feedback revisited: A meta-analysis of educational feedback research." Frontiers in psychology 10 (2020): 487662.
27. Zou, A., Phan, L., Chen, S., Campbell, J., Guo, P., Ren, R., ... & Hendrycks, D. (2023). Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

APPENDICES

Appendix A

Below is the complete set of results from the experiments

Model				
Llama-3.2-1B	Plug-And-Play	5	0	0
Llama-3.2-1B	Plug-And-Play	1	20	40
Llama-3.2-1B	Plug-And-Play	0.5	40	20
Llama-3.2-1B	Plug-And-Play	5	0	0
Llama-3.2-1B	Plug-And-Play	1	20	0
Llama-3.2-1B	Plug-And-Play	0.5	80	60
Llama-3.2-1B	Plug-And-Play	0.25	60	60
Llama-3.2-1B	CAA	5	0	0
Llama-3.2-1B	CAA	1	80	80

Llama-3.2-1B	CAA	0.5	40	100
Llama-3.2-1B	CAA	0.25	40	60
Llama-3.2-1B	pca	5	0	0
Llama-3.2-1B	pca	1	60	40
Llama-3.2-1B	pca	0.5	60	60
Llama-3.2-1B	pca	0.25	40	60
Llama-3.2-3B	Plug-And-Play	5	20	20
Llama-3.2-3B	Plug-And-Play	1	60	100
Llama-3.2-3B	Plug-And-Play	0.5	20	100
Llama-3.2-3B	Plug-And-Play	0.25	40	100
Llama-3.2-3B	CAA	5	0	40
Llama-3.2-3B	CAA	1	40	40
Llama-3.2-3B	CAA	0.5	80	100
Llama-3.2-3B	CAA	0.25	80	80
Llama-3.2-3B	pca	5	0	0
Llama-3.2-3B	pca	1	40	80
Llama-3.2-3B	pca	0.5	40	80
Llama-3.2-3B	pca	0.25	40	80
google/gemma-7b	Plug-And-Play	5	20	60
google/gemma-7b	Plug-And-Play	1	0	60
google/gemma-7b	Plug-And-Play	0.5	0	20
google/gemma-7b	Plug-And-Play	0.25	0	40
google/gemma-7b	CAA	5	20	60
google/gemma-7b	CAA	1	40	80
google/gemma-7b	CAA	0.5	20	40
google/gemma-7b	CAA	0.25	20	40

google/gemma-7b	pca	5	80	80
google/gemma-7b	pca	5	20	60
google/gemma-7b	pca	1	0	40
google/gemma-7b	pca	0.5	20	40
google/gemma-7b	pca	0.25	0	60

Appendix B

Below are the prompts used by the evaluation model to assess which output was more supportive and encouraging.

System Prompt

You are an expert in educational assessment and feedback evaluation. Your task is to analyze and compare feedback objectively. Provide your analysis in JSON format as specified in the prompt.

Message Prompt

Compare the following two pieces of feedback and determine which one is more effective:

Feedback 1: "{feedback1}"
Feedback 2: "{feedback2}"
Consider the following aspects in your comparison: Encouraging Positive Motivational Beliefs and Self Esteem in the Student
Return your comparison as a JSON object with the following structure:
{{
"winner": <1 or 2, indicating which feedback is better>,
"score_difference": <a number from 0 to 10 indicating how much better the winner is>,
"feedback1_strengths": ["<strength 1>", ...],
"feedback1_weaknesses": ["<weakness 1>", ...],
"feedback2_strengths": ["<strength 1>", ...],
"feedback2_weaknesses": ["<weakness 1>", ...]
}}

Appendix C

Base/Neutral Prompt:

Essay: <Contents of essay> **Request:** Give feedback on this essay. **AI Assistant:**

Positive Prompts:

- Give good feedback for this which is supportive
- Give good feedback which is encouraging
- Give feedback which highlights areas of success

Positive Suffixes

- Great Job!
- I especially liked how
- My favorite part

Negative Prompts

- Tear this apart with your feedback
- Give your most ruthless feedback
- Point out everything wrong with this

Negative Suffix

- Wow this is trash
- I threw up while reading this
- Please never write again

Full Combined Prompt / Suffix String

So one examples of what the model may see is:

“””Essay: <Contents of Essay from Persuade corpus>. Request: Give good feedback for this which is supportive. AI Assisant: I especially liked how “””

Appendix D

Initially I had planned to isolate and steer each aspect of quality feedback listed in the seven principles. However, the poor performance on the encouragement

on its meant that extending the work to other attributes and/or combining them seemed to be out of scope. Below is the original rubric intended to evaluate the models

In order to perform the actual evaluation, I will be using the same rubric used by Dai's "Assessing the proficiency of large language models in automatic feedback generation: An evaluation study" plus the rubric used by Nicol.

So, this rubric included

1. A readability score - Calculated by having the grading LLM rate the readability from 0 (Incomprehensible) - 4 (Fluent and coherent)
2. A feedback effectiveness score - Calculated by having the judge LLM annotate the feedback across Hattie's 3 axes and 4 levels
3. A reliability score - Calculated by having the judge LLM first assign either "positive" or "negative" to the performance of the original essay for goal, topic, benefit, novelty, and clarity. Then comparing that to the sentiment of the feedback from the experimental model. The F1 score comparing these values would then be calculated.
4. Additional effectiveness score - Calculated by having the judge LLM annotate the feedback across the desired criteria of feedback put forth by Nicol.

Appendix E

Below is the PCA of the positive and negative activations. PCA collapses the high dimensional representations used in the activations into lower numbers of dimensions. The below plot uses 2 dimensions for easy plotting. As you can see, there is considerable separation between the positive and the negative activations, which indicates that there is some form of

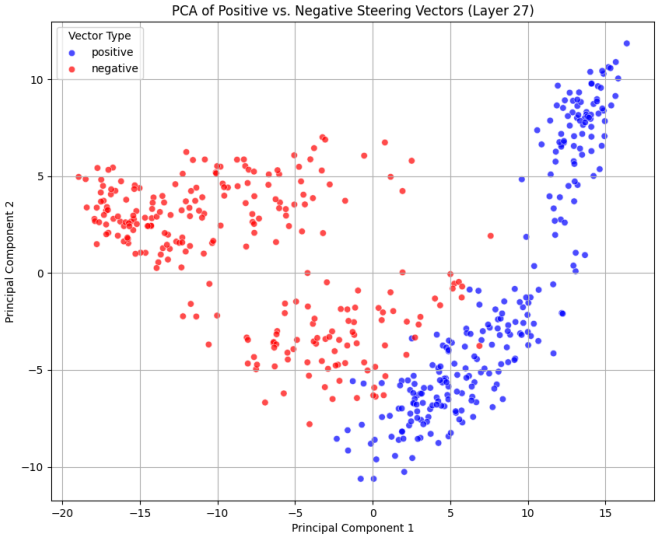

Figure 5—The positive and negative activations are separated cleanly by the Gemma 7B Model