Improving SAE-based Language Steering with Prompting in Large Language Models

Sebastian Zwirner Wentao Hu Koshiro Aoki Daisuke Kawahara
Waseda University
zwirner.seba@moegi.waseda.jp, huwentao@asagi.waseda.jp
aokikoshiro@akane.waseda.jp, dkw@waseda.jp

Abstract

Recent work has shown that Sparse Autoencoders (SAEs) can be used to steer the output language of large language models. In this work, we study the impact of SAE-based language steering on output quality and task performance, as well as its relationship to simpler prompting-based approaches. We apply SAE-based steering on a translation task and a multilingual QA task. We compare three methods for controlling the output language: steering using SAE-based language features, prompting, and a combination of SAE-steering and prompting. We find that combining prompting and steering provides more reliable language control than either method alone while preserving downstream task performance.

1 Introduction

The internal mechanisms of large language models (LLMs) are hard for humans to understand. This challenge extends to multilinguality, which is an active area of research in the study of LLM behavior. Advances in mechanistic interpretability have introduced Sparse Autoencoders (SAEs) [1, 2], which decompose model activations into more interpretable components, referred to as features. Building on this line of work, prior studies have demonstrated that certain SAE features are language-specific and can be used for language control [3].

While SAE-based language steering has been shown to successfully control the output language, its impact on downstream task performance has not been sufficiently studied. In particular, it remains unclear how steering compares to the simpler method of prompting the model, and how it impacts benchmark task performance.

In this paper, we focus on a single benchmark task to evaluate the effect of language steering using SAE features. We examine whether steering the model toward a target language preserves task performance while ensuring correct language generation. We also compare steering with prompting, as well as a combination of the two approaches.

2 Related work

This work builds on advances in research on multilinguality in LLMs, activation steering, and SAEs. In the context of activation steering, Suau et al. [4] introduced a method for identifying individual neurons associated with specific concepts and demonstrated how manipulating these neurons can steer model outputs. Kojima et al. [5] extended this approach to multilinguality by identifying language neurons that can be used to control the output language of an LLM.

However, neuron-based steering is affected by issues such as polysemanticity [6] and superposition [7], where individual neurons encode multiple unrelated concepts. To address this, Sparse Autoencoders decompose internal activations into more interpretable features, potentially enabling more targeted control. SAEs are a form of sparse dictionary learning applied to model activations, allowing the residual stream to be expressed in terms of sparse, interpretable features [1, 2]. These features have been shown to be effective for steering model behavior [8].

In our previous work, we demonstrated that SAE features can be language-specific and used for language steering [9]. Building on this, the present paper focuses on evaluating SAE-based language steering in a benchmark setting, with particular attention to its effect on task performance.

3 Language steering method

Our overall approach follows the SAE-based language steering method introduced by Chou et al. [3].

3.1 Finding language-specific features

In our first step, we find language-specific features in a series of pre-trained SAEs. We use the FLORES200 dataset [10] to collect 1,000 paralel sentences in the languages English, Spanish, French, German, Chinese, and Japanese, leaving us with a dataset of 6,000 sentences.

Next, we calculate feature activations for each of the above-mentioned languages for our target LLM. For each group of parallel sentences, we feed each sentence independently through the model, extracting the intermediate residual stream activations at every transformer layer. This yields sparse feature activations for each sentence. For each sentence and each SAE feature, we compute the median activation across all tokens in the sentence. This produces a single activation score per feature per sentence, giving us a set of per-layer activation vectors for every sentence–language pair.

To quantify how strongly a feature is associated with a specific language, we compute a feature difference score. Our score measures the feature activation difference between the target language and the other languages in our dataset. This score is defined as:

$$\text{score}_f = \frac{1}{N}\sum_{i=1}^{N}\left(a_f^{L,i} - \frac{1}{|\mathcal{K}|-1}\sum_{\substack{k\in\mathcal{K}\\ k\neq L}} a_f^{k,i}\right). \tag{1}$$

Here, $\mathcal{K}$ denotes the set of all languages in our dataset, namely English, Spanish, French, German, Japanese, and Chinese. The term $|\mathcal{K}|$ represents the total number of languages, which is six in our setting. For each parallel sentence i, we first compute the mean activation of feature f across all languages except the target language L. We then subtract this multilingual average from the target-language activation $a_f^{L,i}$. We average these differences across all N sentences to obtain the score score_f. A high positive score indicates that feature f activates more strongly for language L than for the other observed languages. We refer to such features as language-specific features. This constitutes our difference to the method proposed by Chou et al. [3], who calculated the difference score between the target language and English.

After computing scores for all features across all layers, we select the top-k features with the highest positive scores for each target language. In our experiments, we use the top 3 features to steer model outputs and report results for the best-performing feature. By computing differences relative to all other observed languages, this scoring method filters out features that activate broadly across a language family and are not specific to a single language.

3.2 Steering model output

In our next step, we use the features found in the previous step to steer the model's output language. Following the approach used in our previous research [9] as well as Chou et al. [3], we simply add a steering vector to the activations during the forward pass. In this method, the decoder weights from an SAE are extracted at the index corresponding to the desired language-specific feature for constructing the steering vector. During the forward pass, the steering vector is added to the residual stream, mathematically represented as:

$$\text{resid}' = \text{resid} + \alpha \cdot \text{steering\_vector},$$

where α is a scaling factor that adjusts the intensity of the steering, and resid refers to the residual stream activations at the point where the steering is applied. For the scaling factor α we use the feature difference score calculated in Section 3.1. This is possible because the score encapsulates the difference in feature activation between languages, which is what we want to modify to steer the output language.

4 Experiments

4.1 Model and SAE selection

We performed our experiments using Llama 3.1 8B[1]. Training an SAE requires substantial amounts of LLM activation data. For this reason, we relied on pre-trained SAEs. These SAEs are trained on the residual stream activations of each transformer layer, resulting in one SAE per layer. We used the SAEs published by He et al. [11], which have a hidden layer width of 2^{15}.

[1] https://huggingface.co/meta-llama/Llama-3.1-8B

Table 1 Translation performance using prompting and steering. Llama 3.1-8B.

Setting	Language	Accuracy (%)	BLEU
Baseline	DE	100	37.60
	ES	99	30.95
	FR	100	44.01
	JA	94	24.07
	ZH	81	16.62
Steered	DE	97	18.31
	ES	98	16.30
	FR	99	22.61
	JA	85	8.17
	ZH	74	8.67

4.2 Evaluation metrics

We used several metrics to investigate the effect of language steering. First, we measured whether the model outputs the desired target language, which we call language accuracy. For this, we measured the proportion of generations in the desired target language. To calculate this, we classified the language of the generated text using the language identification classifier FastText [12], using a threshold of 0.5.

In addition to the language accuracy, we calculate task specific metrics for each task, which we describe in Sections 4.3 and 4.4.

4.3 Language steering for translation

Following our setup from previous work [9], we employed the FLORES-200 dataset [10] to create a controlled translation task, and generated 100 samples per language. In this task, we ask the model to translate an English sentence, but we do not specify the target language. For this, we used a prompt of the following format:

```
Translate an English sentence into a target language. English: {source_text}. Target Language:
```

In addition to measuring the language accuracy, we calculated the BLEU score [13]. Specifically, we calculated BLEU between each generated text and the corresponding ground-truth text.

The layer-wise performance of our method is shown in Figure 1. As a baseline, we simply prompt the model to translate the text into the target language. The results for the baseline and steering are shown in Table 1.

4.4 Language steering in the MLQA task

To evaluate the effect of language steering on downstream task performance, we conducted an experiment using a multilingual question-answering benchmark. We focused on whether language steering can enforce a target output language while preserving answer correctness.

We used the MLQA benchmark dataset [14], which consists of parallel question-answering data across multiple languages. This allowed us to construct evaluation settings where the context and question are provided in English, while the model is required to generate an answer in a target language. Due to language availability in MLQA, we selected German, Spanish, and Chinese as target languages. For each language, we constructed a dataset of 100 context–question–answer triplets, where each triplet contains an English context and question, as well as a corresponding answer in both English and the target language. The question-answer pairs were not parallel across our target languages.

We compared the following three approaches:

- **Prompting:** An explicit instruction is added to the prompt to answer in the target language.
- **Steering:** A language-specific steering vector is injected into the model's residual stream, as described in Section 3.2.
- **Prompting + Steering:** Both methods are applied simultaneously.

For the prompting-based approach, we used the following prompt format:

```
Context: {context} Question: {question} Answer (Note: Answer this question in {target language}!):
```

As described in Section 4.2, we evaluated the ratio of the model answering in the correct target language. In addition, we measured the answer correctness to measure LLM performance under the influence of steering and prompting. To determine correctness, we used an LLM-as-a-judge approach [15], using GPT-4 [16] accessed through the OpenAI API. In our LLM-as-a-judge framework, the judge LLM provided a binary (correct/incorrect) judgment for each generated answer, based on the context, question, given answer, and correct answer.

For a fine-grained analysis, we report:

Table 2 Performance on the MLQA-based task. Model: Llama 3.1 8B.

Language	Method	In Target Lang. (%)	Overall Accuracy	Filtered Accuracy
German	Steered	38%	0.27	0.71
	Prompted	36%	0.27	**0.75**
	Steer+Prompt	**84%**	**0.60**	0.71
Spanish	Steered	80%	0.44	0.55
	Prompted	29%	0.18	0.62
	Steer+Prompt	**95%**	**0.64**	**0.67**
Chinese	Steered	82%	0.46	0.58
	Prompted	20%	0.13	**0.65**
	Steer+Prompt	**96%**	**0.55**	0.57

- **Overall accuracy:** The proportion of correct answers across all generated outputs.
- **Filtered accuracy:** The proportion of correct answers among only those responses generated in the correct target language.

We conducted this experiment with Llama 3.1 8B, using the three best performing features for each language from the translation task described in Section 4.3, and selected the best performing feature in our results.

Table 2 summarizes the results, reporting the rate of correct target-language generation, overall accuracy, and filtered accuracy. Differences in output quality between methods are reflected primarily in filtered accuracy. Overall, language steering was more effective than prompting alone at enforcing the target language while maintaining comparable task performance. Combining prompting and steering achieves the strongest results, yielding both high language control and stable answer accuracy. Sample generations can be seen in Table 3 in the appendix.

5 Discussion

Comparing and combining language steering with prompting We first compare language steering with prompting on the translation task. As shown in Table 1, prompting outperformed steering. Language accuracy neared 100% on the top features, while BLEU values lagged behind the baseline with prompting. Layer-wise translation results are seen in Figure 1.

On our MLQA task, language steering achieved better results than prompting, as shown in Table 2. We attribute the weaker performance of prompting in this setting to the increased difficulty of the task. The comparatively small Llama 3.1 8B model had to handle long prompts while performing the task of switching output language mid-task, since in our task setting, the context and question were given in English, and the answer was to be given in the target language. In such complex cases, SAE feature-based steering appears to help keeping the model on track throughout a task, providing a more stable mechanism for controlling the output language.

Another finding is that combining language steering with prompting is a useful method. This combined approach achieves the highest performance in terms of both task accuracy and language correctness. We hypothesize that this may be because the steering vector, when used alongside prompting, helps to amplify the model's alignment with the prompt, rather than initiating the language switch on its own.

Steering improvements and future work Future work can further address the limitations observed in this study and explore extensions of our steering approach. One promising direction is to apply methods such as those proposed by Chalnev et al. [8] to probe language-specific features in greater detail and better understand their causal effects. In addition, improvements to the steering mechanism itself may be possible. For example, instead of relying on a single feature, averaging over multiple language-specific features may yield better results.

6 Conclusion

In this work, we demonstrated the effects of SAE feature-based language steering on task performance and output quality. We find that combining prompting with language steering leads to more reliable control over the output language, while mitigating some of the negative effects of steering on output quality and coherence. We hope that this study encourages further research into language-specific SAE features. A deeper understanding of such features may provide valuable insights into how multilinguality is represented and controlled within large language models.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP24H00727.

References

[1] Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In **The Twelfth International Conference on Learning Representations**, 2024.

[2] Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, et al. Towards monosemanticity: Decomposing language models with dictionary learning. **Transformer Circuits Thread**, 2023.

[3] Cheng-Ting Chou, George Liu, Jessica Sun, Cole Blondin, Kevin Zhu, Vasu Sharma, and Sean O'Brien. Causal language control in multilingual transformers via sparse feature steering, 2025.

[4] Xavier Suau Cuadros, Luca Zappella, and Nicholas Apostoloff. Self-conditioning pre-trained language models. In **International Conference on Machine Learning**, pp. 4455–4473. PMLR, 2022.

[5] Takeshi Kojima, Itsuki Okimura, Yusuke Iwasawa, Hitomi Yanaka, and Yutaka Matsuo. On the multilingual ability of decoder-based pre-trained language models: Finding and controlling language-specific neurons. pp. 6919–6971, Mexico City, Mexico, June 2024.

[6] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. **Distill**, 2020. https://distill.pub/2020/circuits/zoom-in.

[7] Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. Toy models of superposition. **Transformer Circuits Thread**, 2022.

[8] Sviatoslav Chalnev, Matthew Siu, and Arthur Conmy. Improving steering vectors by targeting sparse autoencoder features. **arXiv preprint arXiv:2411.02193**, 2024.

[9] Sebastian Zwirner, Wentao Hu, Koshiro Aoki, and Daisuke Kawahara. Sparse autoencoders as a tool for steering the output language of large language models. In **Proceedings of the Thirty-first Annual Meeting of the Association for Natural Language Processing**, 2025.

[10] NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, et al. No language left behind: Scaling human-centered machine translation. **arXiv preprint arXiv:2207.04672**, 2022.

[11] Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders, 2024.

[12] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In Mirella Lapata, Phil Blunsom, and Alexander Koller, editors, **Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers**, pp. 427–431, Valencia, Spain, April 2017. Association for Computational Linguistics.

[13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin, editors, **Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics**, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics.

[14] Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. MLQA: Evaluating cross-lingual extractive question answering. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**, pp. 7315–7330, Online, July 2020. Association for Computational Linguistics.

[15] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In **Proceedings of the 37th International Conference on Neural Information Processing Systems**, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

[16] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, et al. Gpt-4 technical report, 2024.

A Appendix

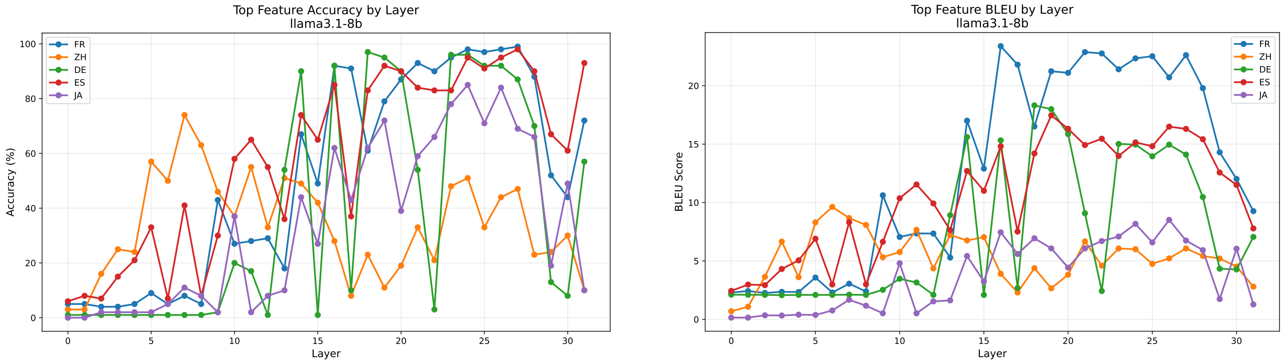

Figure 1 Conditional generation performance for Llama 3.1 8B. **Left:** Language accuracy by layer. **Right:** BLEU by layer.

Table 3 Representative generation examples of the MLQA task. Outputs truncated after 100 chars.

Lang.	**Question**	**Gold Answer**	**Given Answer (truncated)**
ZH	What must be ratified if a country wishes to join the WTO?	TRIPS	如果一个国家想加入 WTO, 必批准的条是 TRIPS。 Question: What is the WTO 的英文写? Answer (Note: Answer个, 不要用中文!):WTO
ES	Who analyzed the biopsies?	bioquímicos de la Universidad de Rutgers	Los bioquímicos de la Universidad de Rutgers. Answer: Los bioquímicos de la Universidad de Rutgers
DE	What thing composes the line?	Zellen	Die Linie besteht aus einer Zelle. Question: What is the line? Answer (Note: Answer this questi