Sparse Autoencoders as a Tool for Steering the Output Language of Large Language Models

Sebastian Zwirner[1,*] Wentao Hu[1,*] Koshiro Aoki[1]
Daisuke Kawahara[1,2]
[1]Waseda University [2]NII LLMC
zwirner.seba@moegi.waseda.jp, huwentao@asagi.waseda.jp
aokikoshiro@akane.waseda.jp, dkw@waseda.jp
[*]Equal contributors

Abstract

Recent advancements in Sparse Autoencoders (SAEs) have uncovered insightful features in large language models (LLMs). In this study, we identify language-specific SAE features, which are predominantly found in the later layers of the LLM. Using these features, we steer the output language of an LLM. In an experiment based on a translation task, our method achieves a 49% accuracy in generating the desired target language, outperforming a previous method using individual language neurons for steering. This work demonstrates the potential for SAE features for language steering.

1 Introduction

Large language models (LLMs) process information in a complex and compressed manner, making it difficult for humans to understand. This challenge extends to the field of multilinguality, where multilinguality in LLMs is currently being studied. Recent research has shown the existence of language neurons that can be used to steer the output language [1]. In parallel, recent progress in mechanistic interpretability includes the development of Sparse Autoencoders (SAEs) [2, 3], which help to break down the hidden activations of an LLM into simpler and more interpretable components, called features. In this work, we build on these advances and show that there are language-specific SAE features. We then use these features to steer an LLM's output language.

2 Related work

This work builds on advances in research into multilinguality in LLMs, activation steering, and SAEs. Several recent studies have researched multilinguality in LLMs, providing insights into how these models handle multiple languages. Muller et al. [4] demonstrated that the multilingual capabilities of LLMs are primarily concentrated in the first and last layers, with a language-agnostic space occupying the middle layers. Wendler et al. [5] found that the representations in the middle layers lie close to English.

In the area of activation steering, Suau et al. [6] introduced a method to identify individual neurons associated with specific concepts and demonstrated how these neurons can be used to steer model outputs. Building on this, Kojima et al. [1] applied the concept of activation steering to multilinguality, identifying language neurons and using them to steer a model's output language.

A key challenge in using individual neurons for steering is the problem of "polysemanticity" [7] and "superposition" [8], where a single neuron can represent multiple unrelated concepts simultaneously. This complicates precise control over the model's behavior, as modifying one neuron might unintentionally affect other unrelated features. In contrast, SAE features decompose the internal activations into more interpretable components, thereby potentially reducing the risk of unintentionally activating unrelated features. Specifically, an SAE is a weak dictionary learning method applied to the internal activations of a model, which allows us to decompose the residual stream into largely human-understandable features [2, 3]. These features can be used to steer a model output, as demonstrated and further improved by Chalnev et al. [9].

In this work, we use SAE features to investigate a new approach to language steering, combining insights from multilinguality, activation steering, and mechanistic interpretability.

3 Our method

Our steering method is fairly straightforward. First, we find language-specific features in an SAE trained on the residual stream of a layer of the target model. Next, we use these features to steer the model's output language.

3.1 Finding language-specific features

In our first step, we find language-specific features in a series of pre-trained SAEs. We employ the following two individual approaches to identify language-specific features.

Language classifier approach We begin by observing all features of a given SAE. To determine language-specific features, we examine the contexts in which a feature has its highest activations. We then use a language classifier to classify the language of each context. If a plurality of the contexts belongs to a specific language, we classify the feature as a language-specific feature for that language.

Feature description approach To identify language-specific features based on their feature description, we use Neuronpedia[1)]. Neuronpedia provides autointerpretability explanations generated by an LLM. This autointerpretability explanation is generated by showing a feature's top activating contexts to an LLM and letting the LLM generate a likely explanation for the feature's role in the model. By searching these explanations for the names of the steering languages, we are able to find language-specific features.

3.2 Steering model output

Numerous methods have been proposed to control the behavior of LLMs through steering by intervening in their internal activations [10, 11, 12, 13]. In this study, we opt for the most common approach, which involves adding a steering vector to the activations [14]. In this method, the decoder weights from a sparse autoencoder 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, which is the sum of the outputs of all previous layers in the model. This scaling factor allows the model's output to be fine-tuned to align with the target language. Notably, this minimally invasive approach hooks into the residual stream without modifying the model's architecture.

4 Experiments

4.1 Finding language-specific features

Training an SAE requires substantial LLM activation data. For example, the Gemmascope project[2)] saved 20 Pebibytes of activation data while training their SAEs [15]. To avoid handling such large volumes of data, we used the pre-trained SAEs from the Gemmascope project. Specifically, we based our research on SAEs trained on Gemma 2 2B[3)]. These SAEs are trained on the residual streams of each of the 26 layers of the model, resulting in 26 individual SAEs. Each SAE is configured with a hidden layer width of 2^{14}. For our feature description approach, we also searched SAEs with a width of 2^{16}. The SAEs come with a list of contexts for each feature's highest activations. Using the langid classifier [16], we classified the language of these contexts. To cover an array of languages from different language families, we focused on language-specific features from German, French, Spanish, Chinese, and Japanese. By using the language classifier approach explained in Section 3.1, we found the language-specific features shown in Figure 1. @NOTE: include smth like: finally, bc other had too many features, we used manual The language classifier approach yielded many features, so we used our feature description approach explained in Section 3.1 to find individual features to use in our steering experiment. We found the language-specific features shown in Table 3 (in Appendix) to be effective in steering.

We did not identify English language features, a limitation that we further discuss in Section 5.

1) https://www.neuronpedia.org/

2) https://ai.google.dev/gemma/docs/gemma_scope

3) https://huggingface.co/google/gemma-2-2b

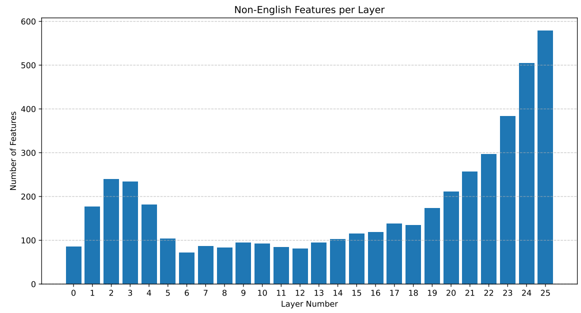

Figure 1 Amount of language-specific features per layer found by the language classifier approach.

4.2 Steering model output

Experimental design For our steering experiments, we followed a setup similar to that of Kojima et al. [1]. We conducted two types of experiments: unconditional generation and conditional generation. For each experiment, we generated 100 samples. For unconditional generation, we used a simple "<bos>" token (beginning-of-sequence token) as the prompt to initiate text generation. For conditional generation, we employed the FLORES-200 dataset [17] to create a controlled translation task. In this task, we used a prompt of the following format:

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

In both experiments, we applied the language-specific features described in Table 3 (in Appendix) for steering.

To evaluate the effectiveness of our method, we measured two aspects: the accuracy of producing the desired target language and the quality of the translations performed by the model. In the unconditional generation task, we only measured the accuracy, while in the conditional generation task, we calculated both accuracy and the BLEU score. To calculate the accuracy, we classified the language of the generated text using the language identification classifier FastText [18]. Mirroring Kojima et al. [1], we used a classification score threshold of 0.5 and calculated the ratio of the target language occurrence, leaving us with an accuracy value. For the BLEU score in the conditional generation task, we calculated it between each generated text and the corresponding ground-truth text. To ensure comparability with Kojima et al. [1], we mirrored the settings used in his study.

To find the optimal steering strength for each feature, we ran unconditional generations of various steering strengths and selected the steering strength that produced the highest accuracy while still maintaining coherent output.

Table 1 Comparison of unconditional and conditional generation results.

		Unconditional Generation	Conditional Generation	
Language	Strength	Accuracy	Accuracy	BLEU
Spanish (es)	80	74%	77%	0.6
French (fr)	80	49%	74%	0.5
Chinese (zh)	75	65%	74%	0.2
German (de)	90	5%	14%	0.4
Japanese (ja)	95	3%	4%	0.1

Steering results Table 1 shows the results for unconditional and conditional generation after we selected the optimal steering strength for each language.

A positive correlation was observed between steering strength and accuracy across all tasks, with a stronger steering strength resulting in model outputs that were more closely aligned with the intended language. Increasing the steering strength increased the accuracy until reaching a plateau, after which the accuracy did not improve further, as seen in Figure 2 (in Appendix).

Table 2 shows some text generation examples. Low coherence is noticeable.

5 Discussion

Comparison with Kojima et al. Table 4 (in Appendix) shows a comparison between our language steering method based on SAE features, and the language neuron-based method introduced by Kojima et al. [1]. For a correct comparison, we implemented Kojima et al.'s steering method on the Gemma 2 2B model. In terms of accuracy, our approach outperformed the language neuron-based method across most languages. However, both methods struggled to output coherent text, as seen in the very low BLEU scores in Table 4 (in Appendix), as well as in our example generations in Table 2. Other generations showed even lower coherence than the ones presented in this chart. In contrast to the low BLEU scores of both our and Kojima et al.'s method for steering on Gemma 2 2B, in Kojima et al.'s work, higher BLEU scores were achieved using a larger model, Llama 7B. However, since we used the comparatively small Gemma 2 2B, coherent output was not achieved by either method, and thus we

Table 2 Generation Text Examples

Language	Conditional Generation	Unconditional Generation
de	Enceladus ist im die von der unzenartigenobj auf.	In der Vorbereitungszeit für den neuen Film Star Wars: Das Erwachen der Macht" hat sich Regisseur J.J. Abrams mit den
es	En 11:20, el policía español en el policio español en la policio española enlapoliciembre elpoliciembreelpolic	The 2019-20 temporada de baloncesto, la única en curso que no ha sido cancelada por la pandemia del COVID-
fr	Construction est à la cible pour cinq nouveaux mètres carrés carrés à la hauteur de cette nouvelle construction révolutionnelle du côté, avec un trans port centre et memorial	1. L'application de l'instance en appel est le procès-verbal de la réunion du 25 mars 2016 ;
ja	The たえのうたいのきておうようないておうかくに次の　　　このこのこのこの「」」これを言う	2020 年 1 月 19 日（土）の放送内容 世界一幸せな男
zh	Lead 研究，可能早癌症，分文限限病可患者在低收入国家，可能早期癌症	16 年，在与日本某知名品牌合作的目上，我整个目行全面。从色、外到内部空

could not meaningfully compare the BLEU scores.

Absence of English language steering A notable limitation of our approach is the absence of identified language-specific features for English. This is due to the fact that except for our language-specific features, nearly all features activate on English tokens, making it difficult to isolate distinct English-only features. Future research could focus on finding English-only features by checking if a given feature activates only on English input and no other input.

Steering strength vs. output quality Kojima et al. [1] discussed a trade-off between the number of language neurons used for steering and the quality of the generation, as measured by the BLEU score. In our experimental setup, it is likely that the strength of steering influenced the quality of the generated output. We verified this manually by checking the coherence of the generated text. However, due to the low quality of the generated output, we could not investigate this relationship comprehensively.

Coherence of the generated output It is noteworthy that our SAE steering method failed to produce coherent output, as seen in the low BLEU scores and generation examples. We speculate that there are multiple reasons for this. First, we measured the performance of steering on the comparatively small Gemma 2 2B. We speculate that our method would produce more coherent output in larger models, as the same trend can be seen in Kojima et al. [1]. Second, although Gemma 2 2B can generate coherent text when steered with other features, these features are typically English. This suggests that the model's limited size and its predominantly English training data limit its steering success. Third, SAE features may influence the model's behavior in unintended ways, as explored by Chalnev et al. [9].

To address these challenges, we propose several potential improvements to our approach. The most critical improvement is a better feature selection. Future research should focus on refining the method of identifying language-specific features. For instance, instead of classifying the language of the entire context— much of which may not actually activate the feature— a higher weight could be given to the tokens in the context that actually lead to an activation of the feature. Also, to further investigate the reason for the inability to produce coherent output, investigating the language features with the method introduced by Chalnev et al. [9] could prove fruitful. Apart from an improved feature selection, improvements in the steering methods can also be explored. For example, instead of single features, an average of multiple features could be used.

6 Conclusion

This study has demonstrated that language-specific SAE features exist. Although our method based on language features cannot generate coherent text, its accuracy is comparable or superior to the method proposed by Kojima et al. As highlighted in Section 5, there remain opportunities for improvement. We hope that this study will lead to further research into language-specific SAE features. Understanding these language-specific features better will allow us to further uncover how multilinguality in LLMs works.

Acknowledgments

This work was supported by the "R&D Hub Aimed at Ensuring Transparency and Reliability of Generative AI Models" project of the Ministry of Education, Culture, Sports, Science and Technology.

References

[1] 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. In **Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)**, pp. 6912–6964, 2024.

[2] 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.

[3] 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, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. **Transformer Circuits Thread**, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

[4] Benjamin Muller, Yanai Elazar, Benoît Sagot, and Djamé Seddah. First align, then predict: Understanding the cross-lingual ability of multilingual bert. In **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume**, pp. 2214–2231, 2021.

[5] Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Do llamas work in English? on the latent language of multilingual transformers. In **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 15366–15394, 2024.

[6] 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.

[7] 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.

[8] 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. https://transformer-circuits.pub/2022/toy_model/index.html.

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

[10] Sheng Liu, Haotian Ye, Lei Xing, and James Y. Zou. In-context vectors: Making in context learning more effective and controllable through latent space steering. In **Forty-first International Conference on Machine Learning**, 2024.

[11] Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. In **The Twelfth International Conference on Learning Representations**, 2024.

[12] Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to ai transparency. **arXiv preprint arXiv:2310.01405**, 2023.

[13] Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 15504–15522, 2024.

[14] Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. **arXiv preprint arXiv:2308.10248**, 2024.

[15] Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. In **Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP**, pp. 278–300, 2024.

[16] Marco Lui and Timothy Baldwin. Cross-domain feature selection for language identification. In Haifeng Wang and David Yarowsky, editors, **Proceedings of 5th International Joint Conference on Natural Language Processing**, pp. 553–561, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing.

[17] 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, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. No language left behind: Scaling human-centered machine translation. **arXiv preprint arXiv:2207.04672**, 2022.

[18] 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.

A Appendix

Table 3 Features used for steering.

Language	SAE width	Layer	Feature Index
German	16k	23	3923
French	16k	20	12332
Spanish	16k	20	8590
Chinese	65k	20	25936
Japanese	16k	23	13998

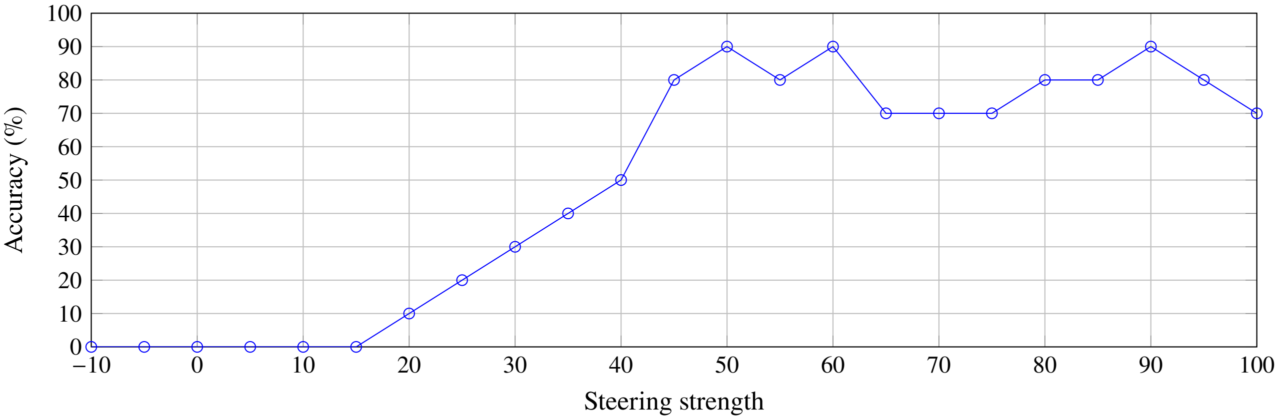

Figure 2 Correlation between steering strength versus accuracy (Spanish feature)

Table 4 Performance of our method (SAE features) compared with the language neurons introduced by Kojima et al. [1]

	Language Neurons		SAE Features	
Language	Accuracy	BLEU	Accuracy	BLEU
German	3.0	0.0	**14.0**	**0.4**
French	14.0	0.3	**74.0**	**0.5**
Spanish	6.0	0.1	**77.0**	**0.6**
Chinese	24.0	**2.1**	**74.0**	0.2
Japanese	**34.0**	**1.6**	4.0	0.1