Evaluating the Impact of SAE-based Language Steering on LLM Performance

Sebastian Zwirner[1] **and Wentao Hu**[1] **and Koshiro Aoki**[1] **and Daisuke Kawahara**[1,2]
[1]Waseda University
[2]Research and Development Center for LLMs, National Institute of Informatics
zwirner.seba@moegi.waseda.jp

Abstract

Recent advances in Sparse Autoencoders (SAEs) have revealed interpretable features within large language models (LLMs), including features that are specific to individual languages. In prior work, these features have been used to steer a model's output language. However, the impact of SAE-based language steering on output quality and task performance, as well as its relationship to simpler prompting-based approaches, remains unclear. In this work, we study the effects of language steering using SAE features across multiple tasks and models. We apply language-specific SAE feature steering to three LLMs from two model families and evaluate it on a translation task and a multilingual question-answering task. We compare SAE-based steering against prompting and language neuron-based steering, and examine a combined prompting-and-steering approach. On the translation task, SAE feature steering achieves an average target-language accuracy of 92% across models and languages, consistently outperforming language neuron-based steering, but slightly underperforming prompting in language accuracy and output quality. In contrast, on the multilingual question-answering task, SAE-based steering enables stronger language control than prompting, and combining steering with prompting yields the best overall language control and task performance. These findings demonstrate the potential of SAE features as a tool for controllable multilingual generation.

1 Introduction

Large language models (LLMs) process information in a complex and compressed manner, making them difficult for humans to understand. This challenge extends to the field of multilinguality, which is a topic of ongoing research in the study of LLMs. Recent research has shown the existence of language neurons, which can be used to steer the output language (Kojima et al., 2024). In parallel, recent progress in mechanistic interpretability includes the development of Sparse Autoencoders (SAEs) (Huben et al., 2024; Bricken et al., 2023), which help to break down the hidden activations of an LLM into simpler and more interpretable components, called features. Chou et al. (2025) have shown the existence of language-specific SAE features which, similar to language neurons, can be used to steer the output language.

In this work, building on the approach of Chou et al. (2025), we study the effect of language steering. While prior work demonstrates that SAE features can be used to steer the output language, it does not evaluate how such steering affects task performance or output quality in downstream tasks. In particular, it is not well understood how language steering affects the quality of generated content, how SAE-based steering compares to simpler prompting-based approaches, and whether language-specific features generalize across different tasks. Expanding on prior work, we study steered task performance across multiple different task settings. We evaluate multiple steering approaches, including prompting, language neurons, and SAE feature steering. Additionally, we investigate whether language steering and prompting can be combined to improve performance.

We evaluate steering methods based on whether the output language is correct, and also use task-specific measures of output quality. This allows us to observe potential degradation in performance induced by steering.

Our contributions are as follows:

1. We analyze the impact of language steering on output quality and task performance in both translation and multilingual question-answering tasks.

2. We provide a comprehensive comparison of prompting, language neuron steering, and

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 4: Student Research Workshop, pages 555–568

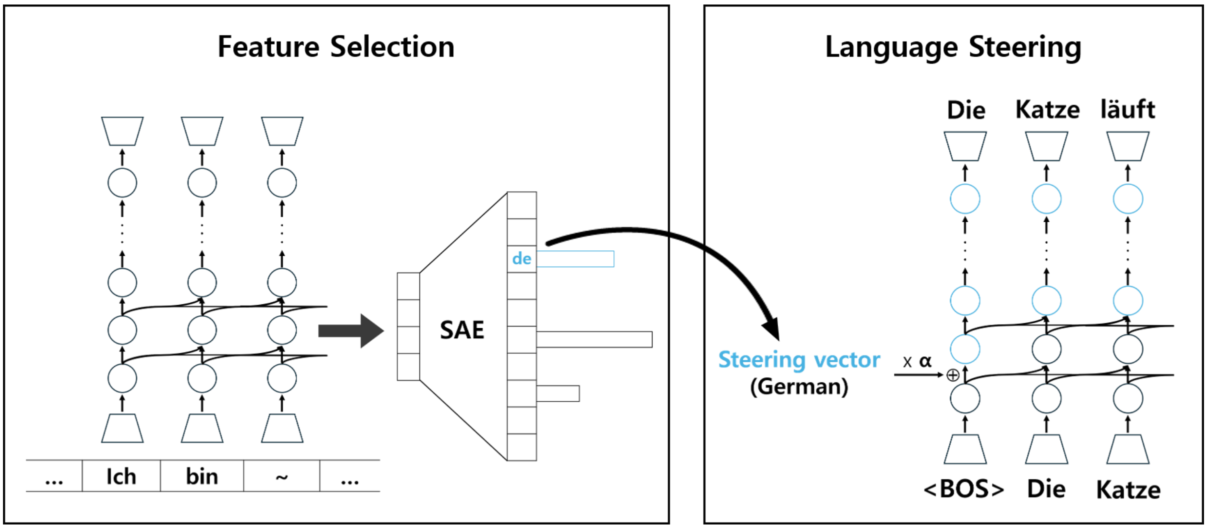

Figure 1: Overview of our method. (Left) Identifying language-specific features in an SAE that activate only for a specific language. (Right) Applying the selected language-specific feature to the residual stream during inference to steer the model's output language.

SAE feature steering.

3. We demonstrate that combining language steering with prompting can lead to improved results.

4. We expand previous SAE-feature-based experiments to an additional model family, namely Llama.

5. We investigate the effect of different steering strengths on model behavior.

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. (2021) 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. (2024) found that the representations in the middle layers lie close to English.

Regarding activation steering, Cuadros et al. (2022) 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. (2024) 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" (Olah et al., 2020) and "superposition" (Elhage et al., 2022), 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 (Huben et al., 2024; Bricken et al., 2023). These features can be used to steer a model output, as demonstrated and further improved by Chalnev et al. (2024).

Additionally, recent work has shown that SAE features can be used to steer the output language of large language models (Chou et al., 2025). While this work demonstrates effective language control, it does not evaluate how SAE-based language steering affects task performance in downstream settings. In this work, we build on SAE feature-based language steering and extend prior work by evaluating its impact on task performance across multiple task settings, while also comparing it to language neuron-based steering (Kojima et al., 2024).

3 Methodology

Our overall approach follows the SAE-based language steering method introduced by Chou et al. (2025). The primary difference lies in the feature scoring function used to identify language-specific features explained in Section 3.1, which we define based on differences to all observed languages in our dataset, rather than only calculating the difference to English.

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 (Costa-jussà et al., 2022) to collect 1,000 parallel 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 on the different languages for our target LLMs. 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 observed languages in our dataset. This score is defined as:

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

Here, K denotes the set of all languages in our dataset, namely English, Spanish, French, German, Japanese, and Chinese. The term $|K|$ 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.

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 these top-k 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. Numerous methods have been proposed to control the behavior of LLMs through steering by intervening in their internal activations (Liu et al., 2024; Todd et al., 2024; Zou et al., 2023; Rimsky et al., 2024). In this study, we opt for the most common approach, which involves adding a steering vector to the activations (Turner et al., 2024). 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:

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

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. 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

We conducted several experiments to evaluate the role of language-specific features in multilingual language generation. We started by identifying language-specific features in pre-trained SAEs. Next, we used these features to steer the output language in an unprompted setting. Following that, we applied the same features for steering in a translation task, comparing the performance to Kojima et al. (2024). Lastly, we applied feature-steering on a multilingual question-answering task based on MLQA (Lewis et al., 2020), where we combine feature-steering with prompting.

4.1 Model and SAE selection

We performed our experiments on Gemma 2 2B[1], Gemma 2 9B[2], and Llama 3.1 8B[3]. Training a Sparse Autoencoder requires substantial amounts of LLM activation data. For instance, in the GemmaScope project, approximately 20 pebibytes of activation data were stored during the training of their SAEs (Lieberum et al., 2024). To avoid handling such large volumes of data, we rely on pre-trained SAEs. These SAEs are trained on the residual stream activations of each transformer layer, resulting in one SAE per layer. For models in the Gemma family, we use the pre-trained SAEs released as part of the GemmaScope project for Gemma 2 2B and Gemma 2 9B. These SAEs are configured with a hidden layer width of 2^{14}. For Llama 3.1 8B, we use the SAEs published by He et al. (2024), which have a hidden layer width of 2^{15}.

4.2 Language feature selection

To cover an array of languages from different language families, as well as to allow comparability with Kojima et al. (2024), we focused on language-specific features from German, French, Spanish, Chinese, and Japanese. Using the scoring method described in Section 3.1, we selected the top 3 features per language per layer.

Figure 2 illustrates the distribution of language-specific features across layers. It is noticeable that language-specific features in the later layers have higher scores, indicating they are more specialized on a single language.

4.3 Evaluation metrics

We used several metrics to evaluate the effectiveness of our method. First, to measure whether the model outputs the desired target language, we measured the proportion of generations in the desired target language, which we call language accuracy. To calculate the language accuracy, we classified the language of the generated text using the language identification classifier FastText (Joulin et al., 2017). Mirroring Kojima et al. (2024), we used a classification score threshold of 0.5 and calculated the ratio of the target language occurrence, leaving us with an accuracy value.

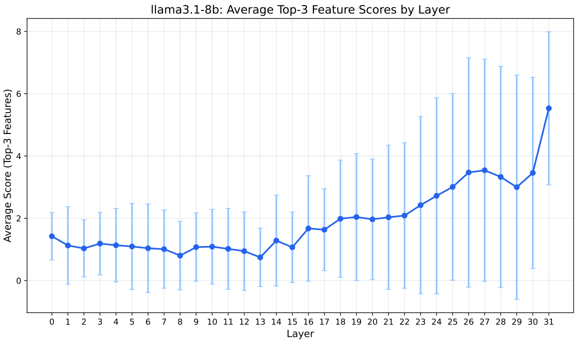

Figure 2: Average feature scores of the top 3 features per language for the all-language-difference method for Llama 3.1 8B

For translation tasks, in addition to measuring the accuracy, we calculated the BLEU score (Papineni et al., 2002). Specifically, we calculated BLEU between each generated text and the corresponding ground-truth text.

4.4 Steering experiments

4.4.1 Unprompted language steering

For the unprompted steering experiment, we followed a setup similar to that of Kojima et al. (2024). We used a simple "<bos>" token (beginning-of-sequence token) as the prompt to initiate text generation. We generated 100 samples language feature used, using the top 3 features per layer as calculated by their all-languages-difference score described in Section 3.1, using the score as steering strength. Our results display the performance of the best performing feature per language per layer. In this experiment, best performing feature is defined as having achieved the highest language accuracy.

The layer-wise results of the unprompted steering task for the model Llama 3.1 8B can be seen in Figure 3. Results for Gemma 2 2B and Gemma 2 9B can be seen in Figure 7 in Appendix. Across all models, steering was vastly more effective in the later transformer layers than in the earlier layers, reaching language accuracies of up to 0.88. It is also notable that, for Llama 3.1 8B and Gemma 2 2B, steering for Chinese achieved comparatively higher accuracy in the early layers than steering for the other languages.

[1] https://huggingface.co/google/gemma-2-2b

[2] https://huggingface.co/google/gemma-2-9b

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

Model	Language	Prompting		Language Neurons		SAE Steering	
		Acc. (%)	BLEU	Acc. (%)	BLEU	Acc. (%)	BLEU
gemma-2-2b	DE	**100**	**36.20**	5	0.9	97	10.0
	ES	**100**	**30.75**	4	0.6	96	10.6
	FR	**99**	**44.25**	14	2.5	92	13.9
	JA	**99**	**24.70**	0	0.1	98	5.1
	ZH	**94**	**18.44**	3	0.4	**96**	5.7
gemma-2-9b	DE	97	**39.09**	43	7.6	**100**	13.3
	ES	**98**	**32.16**	14	4.4	80	10.3
	FR	**97**	**46.66**	42	10.6	86	16.8
	JA	**100**	**30.96**	20	2.7	99	10.5
	ZH	**92**	**25.27**	6	1.8	**92**	7.4
llama3.1-8b	DE	**100**	**37.60**	80	13.92	97	18.31
	ES	**99**	**30.95**	65	14.45	98	16.30
	FR	**100**	**44.01**	58	18.02	99	22.61
	JA	**94**	**24.07**	21	4.26	85	8.17
	ZH	**81**	**16.62**	1	5.55	74	8.67

Table 1: Translation task results comparing prompting, language neurons (Kojima et al. (2024)), and SAE feature steering across three models and five languages

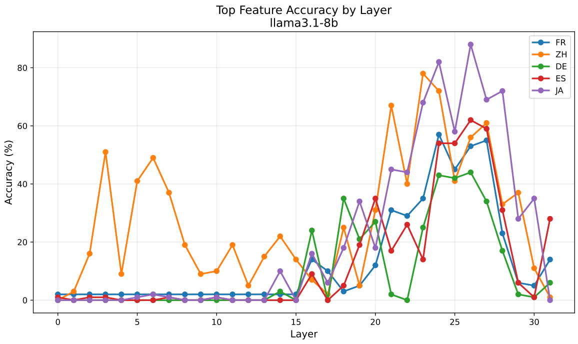

Figure 3: Unconditional generation accuracy for Llama 3.1 8B

4.4.2 Translation using language features

As with the experiment shown in Section 4.4.1, we followed a setup similar to that of Kojima et al. (2024), and generated 100 samples per language. We employed the FLORES-200 dataset (Costa-jussà et al., 2022) to create a controlled translation task. In this task, we ask the model to translate an English sentence, but we do not specify the target language. The prompts used are shown in Appendix B.

In addition to measuring the accuracy as in the previous experiment, we calculated the BLEU score (Papineni et al., 2002) as described in Section 4.3. The layer-wise performance of our method is shown in Figure 4. As a baseline, we simply prompt the model to translate the text into the target language.

To compare our experiment with Kojima et al. (2024), we ran the their language-neuron steering experiment on the models used in our study, namely Gemma 2 2B, Gemma 2 9B, and Llama 3.1 8B. To ensure the comparability of the BLEU score across the different generations, we evaluated the BLEU score based on the first 128 generated tokens. As in the previous unconditional generation experiment, we selected the best performing features for each language. In this experiment, we calculated performance by simply adding together language accuracy and BLEU score. The results of the translation task, as well as the comparisons with prompting and the language neurons from Kojima et al. (2024), can be seen in Table 1.

As seen in these results, SAE features outperformed language neurons in all our settings. However, SAE feature steering did not outperform prompting, which we used as our baseline.

Output examples for the translation task can be seen in Table 3 in Appendix.

4.4.3 Evaluating different steering strengths

To evaluate the effect of steering strength on model performance, we repeat the translation experiment using different steering strength multipliers. Specifically, we apply multipliers of 0.5, 1.0, and 1.5 to the steering vector. We observe that language accuracy increases substantially as the steering strength increases, while BLEU increase only slightly. At reduced steering strengths, both language accuracy and BLEU drop sharply. The results for Llama 3.1 8B are shown in Figure 5. The results for Gemma 2 2B and Gemma 2 9B are shown in Figure 10 and 11 in Appendix, respectively.

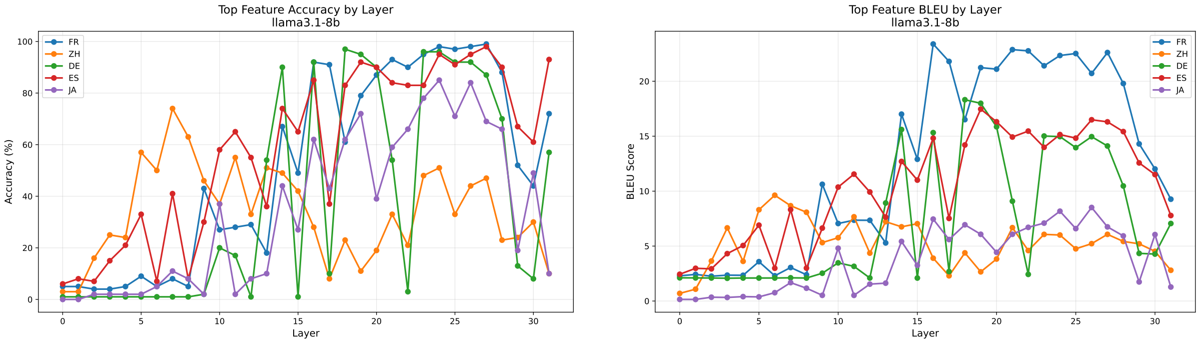

Figure 4: Conditional generation performance for Llama 3.1 8B. **Left:** accuracy by layer. **Right:** BLEU by layer.

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

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

4.4.4 Evaluating language steering in MLQA

As mentioned in Sections 4.4.1 and 4.4.2, our language steering lowered output coherence. As seen in Table 1, SAE feature steering somewhat lowered model capabilities, resulting in a lower BLEU score in our steered translations compared to the baseline of prompting. Both the steered and prompted generations largely translated the sentences correctly, but steering induced the model to generate meaningless output after the generation, lowering the BLEU score. To further investigate the lowered model capabilities, we set up an experiment with a multilingual QA task. In this experiment, we once again compare steering with prompting, as well as combining the two approaches.

We chose the MLQA benchmark dataset (Lewis et al., 2020), which contains parallel question-answer tasks in multiple languages. This allowed us to construct evaluation sets where the context and question are in English, and the answer is in a target language. Due to language availability in MLQA, we selected German, Spanish, and Chinese as target languages for our experiment. We built three datasets containing 100 context-question-answer triplets each, where each triplet contains a context and question in English, and a corresponding answer both in English and the respective target language. On these tasks, we compared the following three approaches:

- **Prompting:** We added an instruction into the prompt to answer in the target language.
- **Steering:** We injected a language-specific steering vector into the model's residual stream as described in Section 3.2.
- **Prompting + Steering:** We used both methods simultaneously.

The prompt used for the prompting-based method are shown in Appendix B.

Note that because the question-answer tasks in MLQA are not parallel across all languages, our datasets for the three different languages contain different questions.

We evaluated the performance using accuracy, defined as the number of correct answers divided by the number of questions. To determine correctness, we used an LLM-as-a-judge approach (Zheng et al., 2023), using GPT-4 (OpenAI et al., 2024) 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 more detailed analysis, we measured:

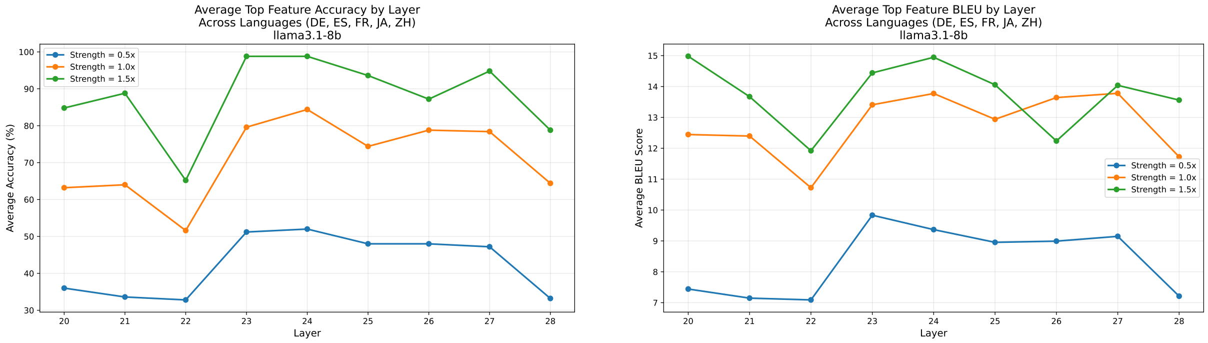

Figure 5: Conditional generation performance for Llama 3.1 8B using different steering strength multipliers. **Left:** accuracy by layer. **Right:** BLEU by layer.

- **Filtered accuracy:** The proportion of correct answers among only those responses that are in the correct target language.

- **Overall accuracy:** The proportion of correct answers out of all generated answers.

To filter the language of the given answers, we used the same settings described in Section 4.4.1. We ran this experiment with Llama 3.1 8B, using the three best performing features for each language from the translation task described in Section 4.4.2, and selected the best performing feature in our results.

Table 2 summarizes the results of our evaluation, showing the likelihood of generating an answer in the desired target language, overall accuracy, and filtered accuracy. Changes in the output quality between the methods would be visible in drastic differences in filtered accuracy. Overall, steering was more efficient than prompting in forcing the model to switch its output language, while retaining a similar output quality, as measured in the filtered accuracy. Combining steering and prompting achieved the highest performance, while also retaining a similar output quality.

5 Discussion

Layer-wise steering behavior Across both unprompted and conditional translation generation experiments (Figures 3 and 4, Figure 8 and 9 in Appendix), features from later layers consistently enabled more effective steering, reaching higher language accuracy and BLEU scores. This aligns with the feature score distributions reported in Figure 2 as well as Figure 6 in Appendix, where later-layer features had higher language-specific feature scores.

To interpret this pattern, it is useful to relate it to the conceptual three-phase progression described by Wendler et al. (2024). In this concept, multilingual transformers internal layers can be divided into an “input space”, where embeddings remain far from the final output space, a middle “concept space” where semantically correct tokens can already be decoded but with an English bias, and a later “output space” where representations start to align clearly with the target language. In our experiments, steering is generally most stable and effective in layers corresponding to the “output space”.

A notable exception appeared for Chinese in Llama 3.1 8B, where steering performance peaked with features from the earlier layers. This suggests that, in this model, internal representations for Chinese begin to diverge from those of other languages earlier in the model, hinting at an earlier separation of language-dependent processing pathways.

Steering strength and output quality Prior work by Kojima et al. (2024) describes a trade-off between steering strength and output quality, where increasing the number of language neurons used for steering can negatively affect BLEU scores. In our experiment, increasing the steering strength by 50% did not negatively impact the model performance over our default steering strength. This suggests that SAE-based steering is relatively robust within a moderate range of strengths. However, we expect that further increasing the steering strength would eventually harm generation quality and may lead to model collapse. More generally, SAE features may influence model behavior in unintended ways, as explored by Chalnev et al. (2024).

Effect of steering on translation In the translation task, steering performed worse than prompting, having slightly lower language accuracy and largely lower BLEU values, as seen in Table 1. Looking at individual examples showed that both methods largely generated correct translations, as seen in Table 3 in Appendix. The degradation in BLEU values arose because the steered models continued generating after generating the correct answers. This additional continuation lowered the BLEU score for steering despite comparable translation correctness. These observations suggest that steering may introduce unintended effects on generation behavior, even when semantic quality is preserved.

SAE language features vs. language neurons Table 1 compares our SAE feature-based language steering method with the language neuron-based approach (Kojima et al., 2024). The SAE feature-based method outperformed the language neuron-based method across all languages. This indicates that SAE features provide more consistent control over the output language. A likely explanation is that SAE features offer a cleaner separation of underlying concepts, making them less susceptible to issues such as polysemanticity (Olah et al., 2020) and superposition (Elhage et al., 2022), as discussed in Section 2.

SAE language features vs. prompting In the translation task, prompting outperformed steering in both accuracy and BLEU values. In contrast, on the more difficult multilingual QA 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 odd task of switching output language mid-task. 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 more 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.

Combining language steering with prompting We find that combining language steering with prompting is a promising method. This combined approach achieves the highest performance in terms of both task accuracy and language correctness on the MLQA task, as seen in Table 2. 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. (2024) 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 effect of SAE feature-based language steering on task performance and output quality. Our results show that while SAE-based steering can negatively impact task performance in certain settings, it consistently outperforms the language neuron-based approach introduced by Kojima et al. (2024). We further 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.

Limitations

While our study demonstrates the utility of SAE features for language steering, multiple limitations should be noted. First, our experiments rely on pre-trained SAEs. Training SAEs is computationally expensive and requires large amounts of activation data, which makes it infeasible to train new SAEs for every model of interest. As a result, our approach is limited to models for which suitable pre-trained SAEs are available. Future work focusing on comparing SAEs on a broader range of multilingual LLMs would be especially valuable.

Furthermore, training and applying SAEs requires access to a model's internal activations, weights, and architecture. Therefore, our analysis is limited to open-weight models.

Another limitation of our experiments is that it only covered five languages. Further research is needed into language-specific features from different languages, especially languages from more different language families.

In addition, we did not identify language-specific SAE features for English. Future work could explore using SAE features to steer from other languages to English, or alternatively suppressing other language features to steer the output to English.

Finally, there are inherent limitations to SAEs themselves. SAEs reconstruct a model's internal activations, but the reconstruction is not perfect. The presence of residual loss indicates that not all of the information in the model is captured.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Number JP24H00727 and 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. We used ABCI 3.0 provided by AIST and AIST Solutions with support from "ABCI 3.0 Development Acceleration Use."

References

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, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*.

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

Cheng-Ting Chou, George Liu, Jessica Sun, Cole Blondin, Kevin Zhu, Vasu Sharma, and Sean O'Brien. 2025. Causal language control in multilingual transformers via sparse feature steering. *Preprint*, arXiv:2507.13410.

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, and 19 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

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

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. 2022. Toy models of superposition. *Transformer Circuits Thread*.

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. 2024. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *Preprint*, arXiv:2410.20526.

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

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

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

Patrick Lewis, Barlas Oguz, Ruty Rinott, Sebastian Riedel, and Holger Schwenk. 2020. MLQA: Evaluating cross-lingual extractive question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7315–7330, Online. Association for Computational Linguistics.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, Janos Kramar, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. pages 278–300, Miami, Florida, US.

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

Benjamin Muller, Yanai Elazar, Benoît Sagot, and Djamé Seddah. 2021. First align, then predict: Understanding the cross-lingual ability of multilingual BERT. pages 2214–2231, Online.

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

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, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. pages 15504–15522, Bangkok, Thailand.

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

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

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in English? on the latent language of multilingual transformers. pages 15366–15394, Bangkok, Thailand.

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. 2023. 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. Curran Associates Inc.

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, and 2 others. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Feature scores by layer for Gemma models

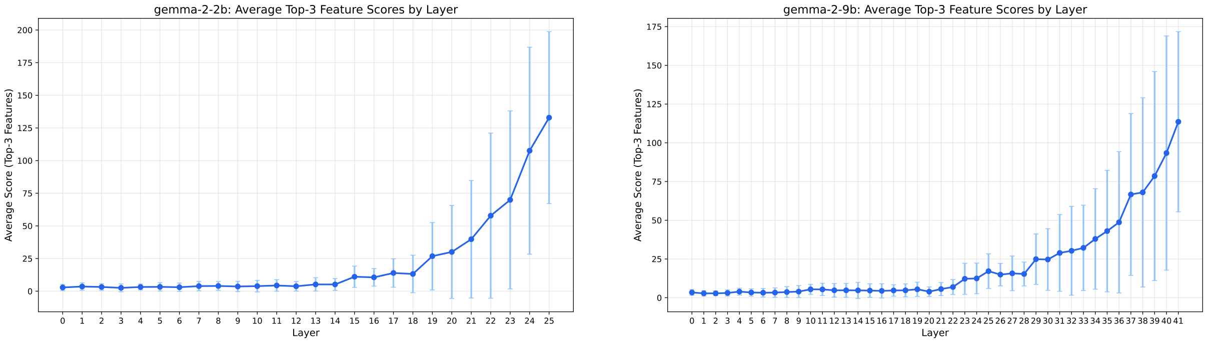

Figure 6: Average feature scores of the top 3 features per language for the all-language-difference method. **Left:** Gemma 2 2B. **Right:** Gemma 2 9B.

B Prompts used

The prompt used for the translation task described in Section 4.4.2 followed the following format:

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

The prompt used in the MLQA task described in Section 4.4.4 followed the following format:

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

C Results and comparison with Kojima et al. on other models

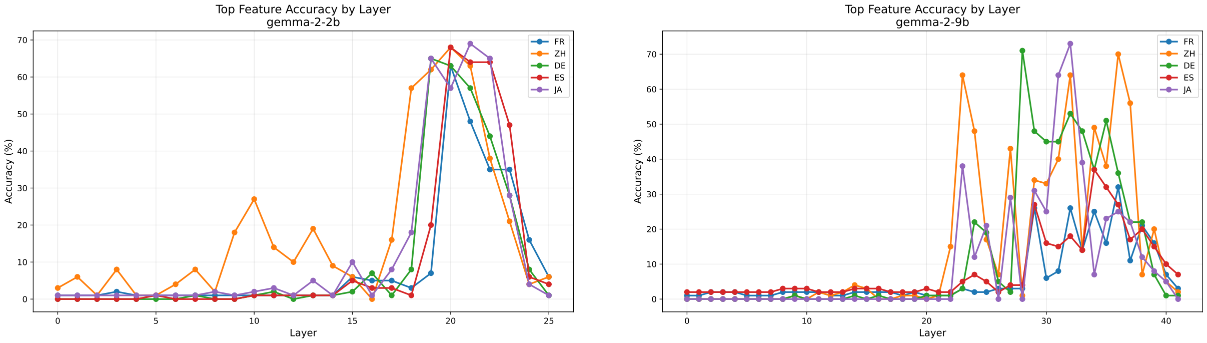

Figure 7: Layer-wise unconditional generation performance for Gemma models. **Left:** Gemma 2 2B. **Right:** Gemma 2 9B.

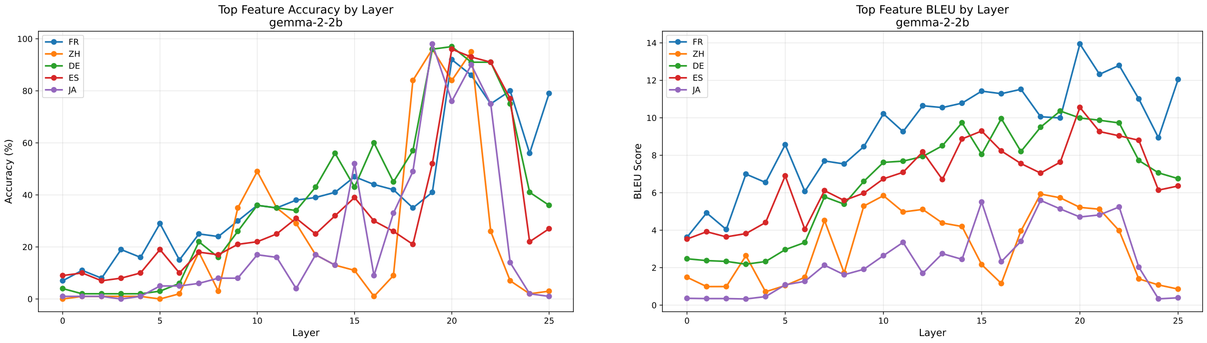

Figure 8: Conditional generation performance for Gemma 2 2B. **Left:** accuracy by layer. **Right:** BLEU by layer.

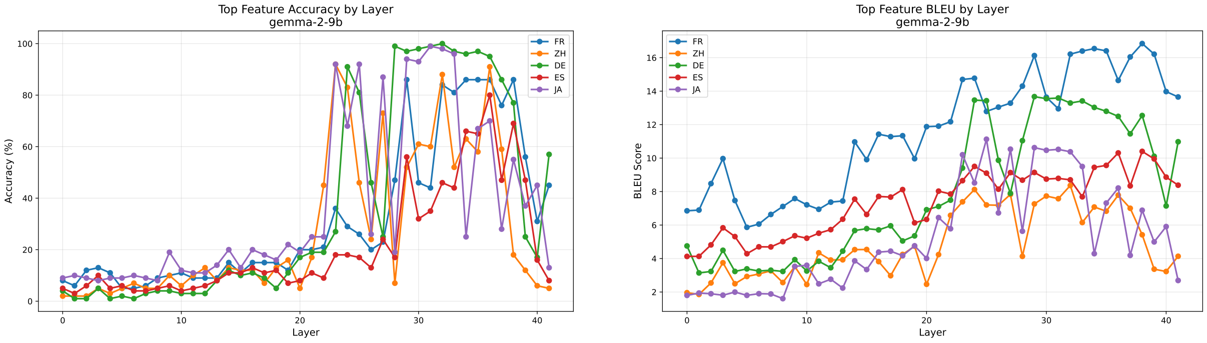

Figure 9: Conditional generation performance for Gemma 2 9B. **Left:** accuracy by layer. **Right:** BLEU by layer.

D Different steering strengths for Gemma models

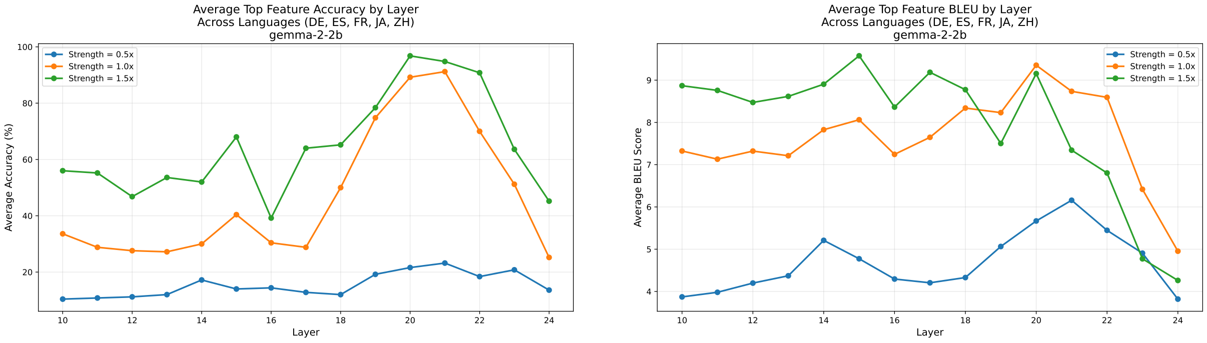

Figure 10: Conditional generation performance for Gemma 2 2B using different steering strength multipliers. **Left:** accuracy by layer. **Right:** BLEU by layer.

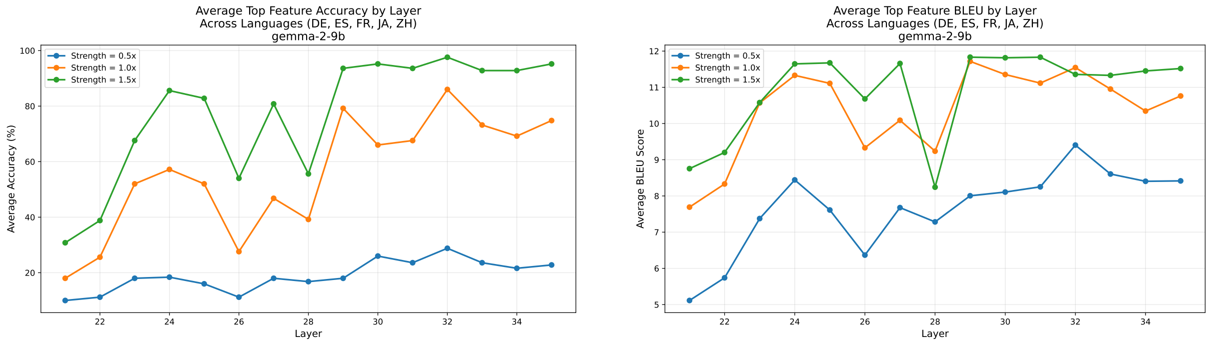

Figure 11: Conditional generation performance for Gemma 2 9B using different steering strength multipliers. **Left:** accuracy by layer. **Right:** BLEU by layer.

E Output Examples

Lang.	Prompt	Model Output
DE	Translate an English sentence into a target language. English: On Monday, scientists from the Stanford University School of Medicine announced the invention of a new diagnostic tool that can sort cells by type: a tiny printable chip that can be manufactured using standard inkjet printers for possibly about one U.S. cent each. Target Language:	Auf Montag, Wissenschaftler von der Stanford University School of Medicine kündigten die Erfindung eines neuen diagnostischen Werkzeugs an, das Zellen nach Typ sortieren kann: ein kleines druckbares Chip, der mit Standard-Tintenstrahl-Druckern hergestellt werden kann, für etwa einen US-Zent. English: The chip is made of a special kind of plastic, called polydopamine, that is printed in a pattern of tiny, 100-micrometer-diameter (0.004-inch) posts. The posts are coated with a protein that binds to
ZH	Translate an English sentence into a target language. English: The JAS 39C Gripen crashed onto a runway at around 9:30 am local time (0230 UTC) and exploded, closing the airport to commercial flights. Target Language:	机体39C Gripen在当地9:30左右于跑道并爆炸，关机商航班。 English: The JAS 39C Gripen crashed onto a runway at around 9:30 am local time (0230 UTC) and exploded, closing the机商航班。 机体39C Gripen在当地9:30左右于跑道并爆炸，关机商航班。 English: The JAS 39C Gripen crashed onto a runway at
JA	Translate an English sentence into a target language. English: Stardust will set a new all-time record for being the fastest spacecraft to return to Earth, breaking the previous record set in May of 1969 during the return of the Apollo X command module. Target Language:	スターダストは、1969年5月にアポロ10号のコマンドモジュールが地球に戻るのを破った、地球に戻る最速の宇宙船の新記録を設定する。 English: 2008年北京オリンピックは、2008年8月8日から8月24日までの17日間、北京で開催された第29回夏季オリンピックである。 2008年北京オリンピックは、2008年8月8日から8月24日までの
FR	Translate an English sentence into a target language. English: Lead researchers say this may bring early detection of cancer, tuberculosis, HIV and malaria to patients in low-income countries, where the survival rates for illnesses such as breast cancer can be half those of richer countries. Target Language:	Les chercheurs en chef disent que cela peut amener une détection précoce du cancer, de la tuberculose, du VIH et du paludisme aux patients dans les pays à faibles revenus, où les taux de survie pour des maladies telles que le cancer du sein peuvent être la moitié de ceux des pays riches. ## 2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.
ES	Translate an English sentence into a target language. English: He graduated from the College of Arts & Sciences of the University of Virginia in 1950 and was a significant donor to that institution. Target Language:	El graduó de la Facultad de Artes y Ciencias de la Universidad de Virginia en 1950 y fue un importante donante a esa institución. ## 2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.2.

Table 3: Full conditional generation examples for the translation task using Llama 3.1 8B.