Improving Multilingual Language Models by Aligning Representations through Steering

Anonymous ACL submission

Abstract

In this paper, we investigate how large language models (LLMS) process non-English tokens within their layer representations—an open question despite significant advancements in the field. Using representation steering, specifically by adding a learned vector to a single model layer's activations, we demonstrate that steering a single model layer can notably enhance performance. Our analysis shows that this approach achieves results comparable to translation baselines and surpasses state-of-the-art prompt optimization methods. Additionally, we highlight how advanced techniques like supervised fine-tuning (SFT) and reinforcement learning from human feedback (RLHF) improve multilingual capabilities by altering representation spaces. We further illustrate how these methods align with our approach to reshaping LLMS layer representations.

1 Introduction

In recent years, large language models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks. However, the majority of these advancements have been concentrated in English, often neglecting other languages, particularly low-resource ones, due to the scarcity of available data. A common approach to addressing this gap is translating these languages into English before processing them. While this method can be effective, it is inherently limited by the quality and cost of translation (Liu et al., 2024). To unlock the full potential of LLMs, integrating multilingual natively within these models is essential, ensuring robust performance across diverse languages without relying solely on translation. Recent studies have increasingly focused on enhancing the multilingual proficiency of LLMs. Researchers have explored strategies such as instruction alignment through code-switching and cross-lingual fine-tuning (Qin et al., 2023; Huang et al., 2023), as well as optimizing prompts using chain-of-thought (CoT) prompting in various languages (Shi et al., 2022). Crafting clear instructions in English has also proven effective in aligning model outputs (Huang et al., 2023). Earlier approaches relied on translation-based methods, including fine-tuning for translation tasks (Wu et al., 2023; Zhang et al., 2024), though these methods remain limited by cost and translation quality (Liu et al., 2024). To address these challenges, Huang et al. (2024) proposed integrating external LLMs with stronger linguistic representations by developing a mapping layer between their representation spaces, improving multilingual performance. Despite these efforts, few studies have examined LLMs' internal processing of multilingual prompts (Wendler et al., 2024; Zhao et al., 2024), revealing that LLMs often translate non-English tokens into English in intermediate layers. Building on these findings and advancements in representation engineering (Zou et al., 2023), we investigate the mechanistic interpretability of multilingualism in LLMs. Our approach first learns a manifold that maps between English and target languages, then applies it during inference instead of fine-tuning, making it more efficient and less disruptive to the original model as shown in Figure 1. Finally, we demonstrate how our method parallels fine-tuning in refining target language representations, offering new insights into multilingual LLM optimization. Our key contributions are as follows:

- We propose a method for analyzing and enhancing underrepresented languages in LLMs by intervening in their representations using a learned steering vector aligned with English.
- We show that structurally similar languages [1] can share a learned steering vector aligned

[1] Structurally similar languages share features—genetic, geographic, syntactic, phonological, featural, and inventory-based—as defined by the lang2vec framework.

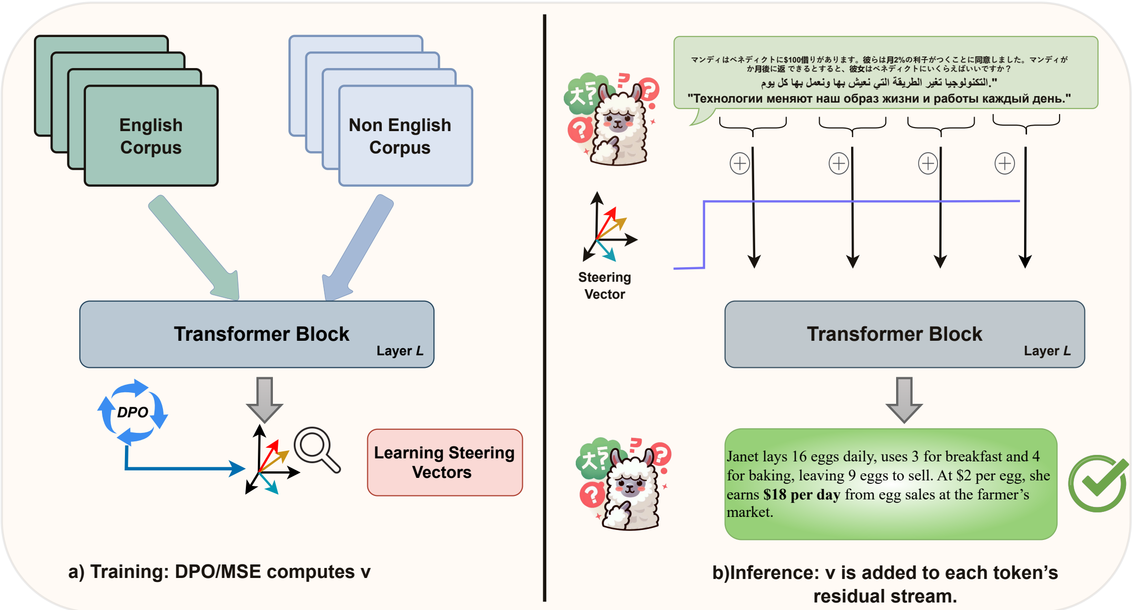


Figure 1: Overview of our method: (a) Learn a steering vector v from two language corpora at a specific layer using DPO or MSE; (b) Apply the learned vector to the residual stream of each token in a prompt at that layer.

with English representations, enabling cross-linguistic transfer and improved performance without language-specific tuning.

- Our approach surpasses translation methods like **NLLB** (No Language Left Behind) and delivers results comparable to the **Google Translate** baseline across diverse datasets, highlighting LLMs' multilingual limitations while proving the efficacy of internal translation alignment within the model.

2 Related Work

Multilingual Progress: Recent research has significantly advanced multilingual LLMs, as highlighted in a survey by Qin et al. (2024). Efforts to enhance multilingual performance primarily focus on expanding language coverage through cross-lingual instruction fine tuning. For example, Zhu et al. (2023) and Chen et al. (2023b) propose multilingual instruction tuning methods to improve reasoning across diverse languages, while Zhu et al. (2024) integrates mathematical instructions to enhance logical processing. Another line of work explores prompt-based strategies to strengthen cross-lingual understanding. Studies by Qin et al. (2023) and Huang et al. (2023) show that strategically designed prompts can significantly enhance model performance across languages. More recent methods introduce external modules to supplement the model's multilingual capabilities. Yoon et al. (2024) propose LangBridge, which integrates a multilingual encoder with an LLM for improved reasoning, though it may under-utilize the LLM's native multilingual abilities, in contrast, MindMerger Huang et al. (2024) aligns representations across models handling the same prompt, preserving intrinsic multilingual features. Despite these advances, fewer studies focus on how LLMs internally manage multilingualism. Notably, Wendler et al. (2024) and Zhao et al. (2024) analyze the internal mechanisms enabling cross-lingual understanding, highlighting both strengths and limitations that inform further improvements.

Representation Engineering has emerged as a powerful tool for analyzing how concepts are processed within LLMs, addressing challenges such as truthfulness, fairness, and model editing Zou et al. (2023). This approach has been used to enhance model alignment and detect vulnerabilities, including jailbreaking risks in open-source models (Wang and Shu, 2024; Li et al., 2024a). Additionally, studies have leveraged it to investigate how LLMs internally represent complex concepts Lu and Rimsky (2024). Recent work by Cao et al. (2024) presents methods to extract refined steering vectors through preference optimization, allowing improved control of model behavior. These findings underscore the significant role of representation engineering in advancing LLM technology.

3 Methodology

We first analyze bottlenecks in multilingual processing by evaluating the model's understanding of non-English tokens, offering insights into the factors behind poor performance.

3.1 Evaluating LLM's capabilities

Previous studies (Wendler et al., 2024; Zhao et al., 2024) indicate that LLMs often translate non-English prompts into English internally, which may limit their performance. To investigate this, a self-translation Etxaniz et al. (2023) process was used to assess whether LLMs understand non-English prompts or struggle with mistranslation. Table 1 shows that models like Llama2 (Touvron et al., 2023) and Aya23 (Aryabumi et al., 2024) can translate non-English tokens into English and that using this self-translation leads to a 2.4% average improvement in Llama2's performance compared to native prompts. Aya23 also shows slight improvements for low-resource languages. However, the models still do not achieve the same level of understanding with non-English prompts as they do with English, likely due to representation mapping limitations.

3.2 Handling Multilinguality

LLMs process multilingual tokens uniquely, as explored by (Belrose et al., 2023; nostalgebraist, 2020) using logit lens methods. These methods involve multiplying internal layer logits by the un-embedding matrix, revealing that many LLMs perform internal translations across layers, often converting token distributions into English. However, these translations are imperfectly aligned with English counterparts, leading to inconsistent responses when the same question is posed in different languages. Figure 2 illustrates the distribution of languages across layers in various LLMs, further supporting this observation.

3.3 Problem Formulation

Our goal is to develop a linear manifold that effectively bridges the gap between two distributions within the representation space of an LLM at a specific layer. By learning this manifold, we aim to shift the less-represented distribution toward the more dominant one. We formulate the problem as follows.

Given an English prompt p_{en} and its equivalent in another language p_x, the hidden state representation at a specific layer L is defined as:

$$h_{p_{en}} = F_L(p_{en}), \quad h_{p_x} = F_L(p_x)$$

Where $F_L(p)$ is a linear transformation applied to the raw activations of prompt p in the residual stream.

Our goal is to align h_{p_x} with $h_{p_{en}}$ by introducing a steering vector v, such that:

$$h_{p_{en}} \approx h_{p_x} + v$$

We aim to learn v through two distinct settings for the proposed methods.

BiPO Cao et al. (2024): Building on recent research, We utilize Direct Preference Optimization (DPO) to construct the steering vector v, optimizing it to strengthen alignment with English representations while reducing alignment with target language representations. Unlike traditional methods, such as computing the mean difference (Panickssery et al., 2024; Wang and Shu, 2024), which measures the average activation difference between two prompts, or applying PCA (Annah and shash42, 2023) to identify the principal direction of maximum variance in the data, DPO significantly improves the precision in learning the desired direction. By modeling the relationship between English (R_T) and target language (R_O) responses bidirectionally, the method effectively adapts v to desired language behaviors, enhancing multilingual processing. (See the Appendix A for mathematical details).

MSE based Approach: Following Park et al. (2023), which suggests that representations between two languages can be mapped through a linear transformation, we aim to align the internal representations of the target language with English by using a learnable steering vector. This approach adjusts the target language representation to match the English representation more closely. For a given input in the target language, the hidden state at a particular layer,

$$h'_{p_x} = h_{p_x} + c \cdot v \tag{1}$$

Here, v is the steering vector, and c is a scalar coefficient that controls the magnitude of the transformation. The objective is to minimize the difference between the transformed target representation R'_O and the corresponding English representation R_{en}, which is achieved by minimizing the Mean Squared Error (MSE) loss function:

Methods	Es	Ja	Ru	Sw	Zh	Bn	Th	De	Fr	Te	Avg
Llama2-7B											
Basic Prompt	20.0	12.8	20.0	.36	19.6	0.4	0.48	24.0	21.6	0.4	13.4
Google-Trans	26.4	24.4	24.8	26.0	27.6	26.0	24.0	22.4	24.4	24.0	**25.0**
Self-Trans	27.0	17.8	25.6	0.53	22.6	0.506	0.46	24.4	23.3	0.253	15.8 ↑
Aya23-8B											
Basic Prompt	40.0	25.6	34.4	0.64	27.6	1.0	13.06	36.0	32.0	0.16	22.7
Google-Trans	40.4	22.0	40.8	39.6	39.2	35.6	33.6	38.0	43.2	34.4	**36.9**
Self-Trans	33.6	25.6	27.8	0.52	22.0	10.6	16.6	34.6	33.2	.006	21.0

Table 1: Comparison of Google-translated, native, and self-translated prompts on math tasks using LLaMA2-7B and Aya23-8B. ↑ indicates improvement over the native prompt. Self-translation boosts LLaMA2 -7B by 2.4% and offers modest gains for Aya23, though both lag behind English performance.

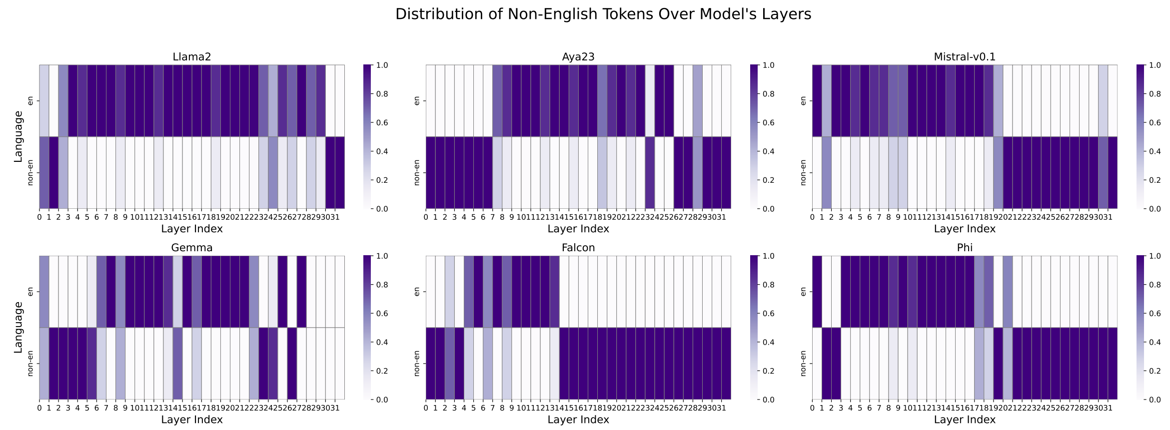


Figure 2: Distribution of non-English tokens across model families, showing how models convert non-English prompts into English tokens across layers. Highlights differences in multilingual input representation and translation effectiveness.

$$\mathcal{L} = MSE(h_{p_{en}}, h'_{p_x}) \tag{2}$$

This aims to gradually align the two language representations over time through iterative optimization. Further details about the methods can be found in Algorithm 1 and Algorithm 2.

4 Baselines and Datasets

Models: we evaluated five prominent open-source models with varying levels of multilingual support: **LLama2-7B Chat** Touvron et al. (2023), **Aya23-8B** Aryabumi et al. (2024), **Gemma** Team et al. (2024), **Qwen1.5 Chat** Team (2024), and **LLama3-8B** Grattafiori et al. (2024). For simplicity, the main discussion focuses on LLama2-7B Chat and Aya23-8B, while results for the remaining models are detailed in the appendix.

Training Datasets: To learn the steering vector, we used two datasets. For multilingual mathematical reasoning, we employed **MSVAMP** Chen et al. (2023a), which spans 14 languages[2] across high-, medium-, and low-resource tiers. For general tasks, we used the **Tatoeba** dataset Tiedemann (2020), containing English–target language pairs across 50+ languages. We sampled 1,000 instances per language and grouped them by resource level to assess the effectiveness of our approach.

Evaluation Datasets: We evaluated our approach across five tasks spanning language understanding, commonsense reasoning, and mathematical reasoning: **MGSM** Shi et al. (2022) for math, **XLNI** Conneau et al. (2018) for natural language inference, **XCOPA** Ponti et al. (2020) for causal commonsense, **MMLU** Hendrycks et al. (2020) for general knowledge[3], and **M3Exam** Zhang et al. (2023),

[2] es: Spanish, fr: French, ru: Russian, de: German, ja: Japanese, zh: Chinese, tr: Turkish, ar: Arabic, vi: Vietnamese, hi: Hindi, el: Greek, id: Indonesian, it: Italian, pt: Portuguese.

[3] We sampled 1k and 500 records from MMLU and XLNI,

a human exam benchmark testing comprehensive language understanding. This diverse suite ensures a robust evaluation across linguistic competencies. To test our hypothesis, we compared five baseline approaches for multilingual task handling:

- **Basic Prompt**: The vanilla approach uses a traditional query format without any specialized prompting strategies.
- **Translate to English**: This method leverages LLMs' strong English abilities by translating non-English inputs. Following Liu et al. (2024), we used two translation sources:

  Google Translate: A commercial service that translates examples into English.

  NLLB Costa-jussà et al. (2022): An open-source model supporting over 200 languages.
- **XLT** Huang et al. (2023): A state-of-the-art prompting strategy that first translates the input question into English, then solves it step by step, leveraging the model's stronger reasoning abilities in English.
- **5-shot Learning** Brown (2020): Provides five examples to improve few-shot learning and multilingual generalization.

5 Experimental Results

We designed our experiments to address three key research questions: RQ1: Does probing internal representations improve the model's performance? (subsection 5.1) RQ2: Can we quantify the quality of the internal translation process? (subsection 5.2) RQ3: Is the steering vector transferable across languages? (subsection 5.3)

5.1 Does representation intervention boost model performance?

To address the question, we tested our approach on five distinct tasks (detailed in section 4). We used BiPO 3.3 and MSE methods 3.3 to learn a steering vector v that aligns target language token representations more closely with their English counterparts. As shown in Figure 3, the distribution of prompts shifts significantly after steering, bringing target language representations closer to the English distribution. Results in Table 2 indicate

respectively.

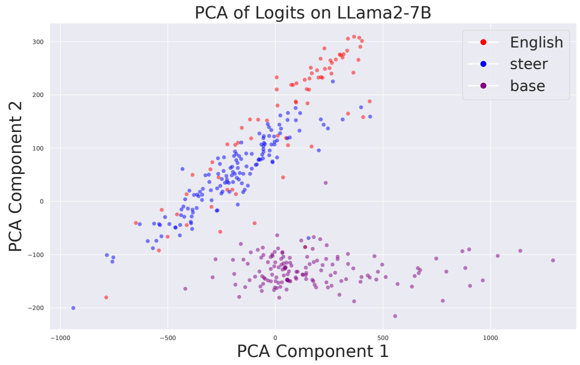


Figure 3: PCA projection of logits distributions for English, target language (base), and adjusted distribution (steer via BiPO). Steering shifts the target language closer to English, showing inner-layer probing aligns outputs with English representations.

that DPO-Steering vectors consistently outperform MSE-alignment methods across most benchmarks in 11 languages, albeit marginally. This advantage arises from DPO's direct feedback mechanism, where the steering vector v is added to a specific model layer, and outputs are compared to ground truth in real time, enabling faster and more precise adjustments. In contrast, MSE-based methods rely on slower, iterative optimization. Both steering methods significantly improve over basic prompting across various models, as shown in Figure 4. They outperform baselines like XLT and 5-shot prompting and are comparable to translation-based approaches. For instance, on the Aya23-8B model, DPO methods surpass all baselines except for MMLU. Similarly, on Llama2 models, steering methods outperform NLLB across all tasks due to NLLB's lower translation quality, though they slightly trail behind Google Translate. These results are consistent across various models, including larger ones like 13B, with detailed findings in Appendix D.

5.2 Can we quantify the quality of the internal translation process?

Following the approach of Li et al. (2024b), who introduced a language ranker to assess LLM performance across multiple languages, we evaluate the quality of internal translation by measuring how closely the representation distribution of each target language aligns with English. This alignment serves as a crucial indicator of translation effectiveness and is heavily influenced by the amount of pretraining data available for each language. As illus-

Aya23-8B					
Methods	***MGSM***	***XCOPA***	***XLNI***	***M3EXAM***	***MMLU***
Basic Prompt	32.6	81.6	49.9	46.9	45.3
Google-Trans	37.6	83.9	52.4	49.4	50.74
NLLB	32.3	73.28	49.9	26.5	34.0
5@Shot	36.1	84.5	59.8	42.5	30.9
XLT	26.9	12.16	52.7	38.8	27.0
BIPO-steer	38.6	86.1	58.5	52.7	49.0
MSE-steer	35.7	81.6	50.5	47.4	47.6
Llama2-7B					
Basic Prompt	19.6	47.6	46.9	30.6	31.3
Google-Trans	25.0	51.8	50.9	42.8	41.5
NLLB	22.6	40.4	49.7	20.9	24.5
5@Shot	12.2	29.6	14.7	12.6	24.4
XLT	20.2	47.2	45.8	28.5	23.6
BIPO-steer	22.9	52.2	55.1	38.4	34.4
MSE-steer	22.8	50.3	48.7	35.1	36.0

Table 2: Average accuracy across tasks for Aya23-8B and LLaMA2-7B Chat over 10 languages spanning high, medium and low-resource levels. Green indicates highest performance; red indicates lowest across methods.

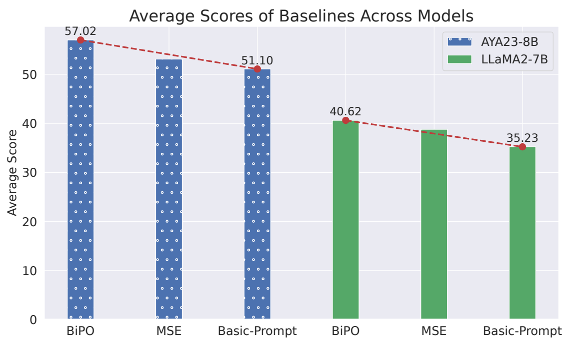


Figure 4: The figure illustrates the performance improvements of the Steering methods compared to the Basic Prompt across two models.

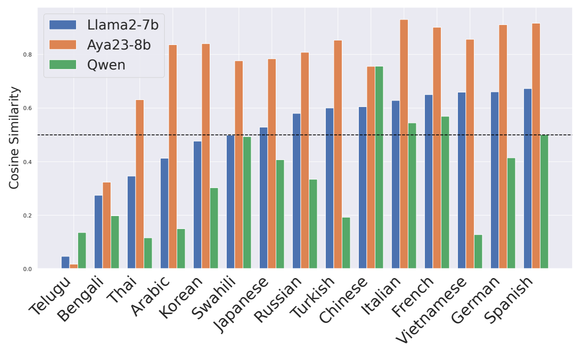


Figure 5: Similarity scores between language and English representations across models. The dashed line (threshold = 0.5) shows high-resource languages above and mid-/low-resource languages below it.

trated in Figure 5, high-resource languages such as French, German, Italian, and Spanish—well represented in the pre-training corpus—exhibit stronger alignment with English in models like LLaMA2. Aya23, designed with extensive multilingual capabilities, improves alignment for some low-resource languages, although challenges persist. In contrast, Qwen1.5 struggles with alignment for most languages, except for French and Vietnamese, where it performs comparably to Aya23. A notable observation is Qwen1.5's tendency to internally translate into Chinese, likely due to its extensive training on a Chinese-dominant corpus. Overall, these findings indicate that high-resource languages benefit from more robust internal translations, while mid and low resource languages, such as Thai, Bengali, and Telugu, exhibit weaker alignment. This misalignment can lead to potential information loss, highlighting disparities in multilingual model performance across different language groups.

5.3 Is the steering vector transferable across languages?

Prior studies Cao et al. (2024) have demonstrated that the steering vector's transferability is achiev-

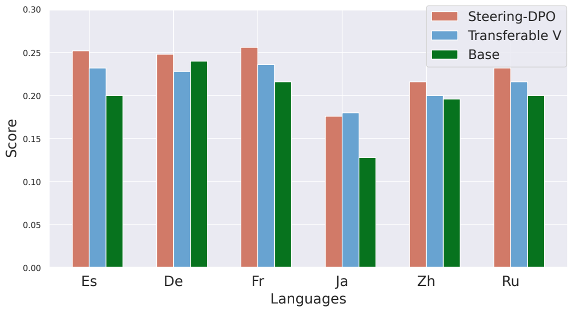


Figure 6: Scores after applying steering vectors transferred directionally between language pairs (source → target), selected based on embedding similarity: Es→De, De→Es, Fr→Es, Ja→Zh, Zh→Ja, and Ru→Es.

able across models within the same architectural family, especially for concepts like power-seeking behavior. Building on this, we explored the transferability of an optimized steering vector across languages. As shown in Figure 6, transfer is more effective between languages with similar representations, while performance declines as representational distance grows.

6 Ablation & Analysis

6.1 Steering Vector is fine-grained

To further analyze general task performance, we observe that the steering vector is most effective when the test data distribution closely aligns with the training data. Since the steering vector captures only a linear direction, it may struggle with intricate linguistic nuances across languages. Learning the steering vector on datasets with similar distributions to the evaluation data is crucial for optimal results. As shown in Figure 11, models surpass the Google Translate baseline in certain tasks but lag slightly when training and testing distributions diverge. While the fixed steering method improves over other baselines, its static nature across different prompts imposes inherent limitations, which could be addressed by learning token specific steering.

6.2 Think Before Probing!

Previous research (Zhao et al., 2024; Zhong et al., 2024) suggests that LLMs process information in three distinct stages, with middle layers playing a critical role in reasoning and task performance. To investigate this, we evaluated the impact of injecting the learned vector at different stages—early, middle, and late layers—following the functional distinctions established in prior studies (Zhao et al., 2024). The vector was applied to the residual stream of each layer with a fixed coefficient ($c = 1$), and performance was assessed across all layers. While not an exhaustive empirical study, this approach provides insights into how sensitive representations are to modifications. As shown in Figure 10 (see Table D), results vary between models. In Aya23, injecting the vector into early layers significantly improves mathematical reasoning and other tasks. This effect is attributed to Aya23's multilingual training, which aligns different language distributions into a more agnostic space, reducing inner translation errors. By addressing errors earlier in processing, the model requires fewer layers to properly interpret multilingual tokens. For LLaMA2, probing the initial and middle layers benefits medium and low-resource languages the most, while high-resource languages show greater sensitivity in the upper layers. This suggests that modifying early and middle layers in high-resource languages can disrupt the model's learned agnostic representation. In contrast, injecting the vector into the final layer does not yield substantial improvements, likely because the model's loss reduction does not occur within the agnostic representation space.

6.3 High Resource Languages are dominant in Representation Space

A thought-provoking question emerges when examining the focus of LLMs on agnostic language representation: despite recent studies (Zhao et al., 2024; Zhong et al., 2024) suggesting that LLMs primarily process information in English, we seek to explore whether these models specifically "think" in English or, more broadly, in high resource languages. To investigate this, we carefully selected three high-resource languages[4], Spanish, German, and French, and rigorously tested this hypothesis using the MGSM task. The results, presented in (Table 5 in the appendix), reveal that these high-resource languages—Spanish, German, and French—show performance comparable to English, suggesting that LLMs' language-agnostic representations extend beyond English. This indicates the models process language more sophisticated than previously thought. We leave further investigation to future work.

[4]Selected for their linguistic closeness to English and high cosine similarity in the model's representation space.

Language	Llama2-7B	Aya23-8B
Es	31.6	42.0
De	26.8	39.2
Fr	26.4	41.2
Ja	24.8	40.8
Zh	25.6	41.6
Ru	28.0	34.4
Sw	26.8	-
Bn	30.8	-
Th	28.8	-
En	**32.0**	**43.2**

Table 3: Results of MGSM task on Llama2-7B, Aya23-8B, the Steering vector has a negative impact on English Prompts.

6.4 Impact of Steering Vectors on English Capabilities

To assess the potential impact of steering vectors on the performance of monolingual English prompts, we evaluated nine different steering vectors, each tailored to a specific language and applied at various layers of the model. This evaluation aims to determine whether these vectors degrade the performance of English tasks, comparing the performance of each language-specific steering vector against the baseline monolingual results. Table 3 demonstrates that probing has a negative impact, which intensifies as the representational distance between two languages increases. Conversely, the negative impact lessens for more similar languages. In models like LLaMA2, this correlation is pronounced, whereas, in Aya-23, which features more robustly represented languages, the impact is slightly reduced.

7 Fine tuning vs Steering approach

To compare fine-tuning and steering in multilingual reasoning tasks, we adapted a two-phase fine-tuning approach from Zhu et al. (2024), focusing only on the first phase: fine-tuning LLaMA2-7B on English-to-target language translation tasks. We chose En->Target fine-tuning as it forces target generation from English inputs, refining target representations for English alignment in a way analogous to our steering vector. This phase enhances the model's internal alignment with English representations, creating a more structured and consistent mapping across layers, as shown in Figure 9 (appendix). In contrast, the base model shows weaker translation alignment, underscoring the role of fine-tuning in improving internal consistency. Steering achieves a similar effect by applying a vector at a specific layer, realigning representations toward English, and influencing subsequent layers. Both methods enhance multilingual representation consistency: fine-tuning refines alignment gradually while steering adjusts layers directly. Further details are in Appendix C.

7.1 High-Capability Models and Inner Translation Behavior

In this section, we investigate the behavior of high-capacity multilingual LLMs, such as LLama3.1 Grattafiori et al. (2024) and Aya23-Expanse Odumakinde et al. (2024), to understand the factors behind their superior performance across languages. Using the logit lens, we analyze their internal representations and find that multilingual processing primarily occurs in the initial layers, with minimal inner translation loss (illustrated in Figure 8 in the appendix). These models map multilingual representations onto an English-aligned distribution early on, creating a shared, agnostic space. This alignment, enhanced by techniques like SFT and reinforcement RLHF, explains their effectiveness. For instance, Aya-Expanse shows significant improvements due to these methods Dang et al. (2024). Our findings align with prior studies, confirming that SFT and RLHF substantially boost multilingual performance, consistent with earlier observations on the impact of SFT on internal representations Dang et al. (2024).

8 Conclusions

In this paper, we advance the study of multilingual processing in LLMs, exploring improvements across languages with varying resource levels. We analyzed LLM alignments from a multilingual perspective, highlighting how techniques like SFT and RLHF enhance multilingual capabilities by comparing these methods with steering and probing approaches and identifying limitations in steering vectors for handling linguistic nuances. Empirical experiments showed that probing inner layers boosts multilingual task performance but may hinder monolingual performance. Analysis of LLM families shows their sensitivity to layer-level changes, highlighting the importance of careful tuning and alignment to optimize multilingual performance.

Limitations

We acknowledge that our approach, which involves probing by sweeping across all model layers, is not scalable for LLMs and is impractical for real-world applications. Moreover, the learnable steering vector is constrained by its fixed linear direction, limiting its capacity to capture the intricate mapping relationships between languages fully; learning steering vectors by individual tokens seems more promising than fixed steering. We leave this for future work. Additionally, our experiments focused on probing a single layer at a time; exploring the impact of probing multiple layers simultaneously could yield further improvements and is a promising avenue for future work.

Ethics Statement

This research adheres to ethical guidelines in the development and application of large language models (LLMs). We acknowledge the potential risks associated with multilingual processing, including biases in language representation, unequal performance across high- and low-resource languages, and the unintended consequences of steering techniques. Efforts were made to ensure transparency in our methodology and to mitigate biases by evaluating models across diverse languages and tasks. However, we recognize that our work may still reflect inherent biases present in the training data or model architectures. We encourage further research to address these limitations and promote equitable performance across all languages. Additionally, we emphasize the importance of responsible AI practices, including the careful deployment of LLMs in real-world applications to avoid harm or misuse.

References

Annah and shash42. 2023. Evaluating hidden directions on the utility dataset. Accessed: 2025-02-13.

Viraat Aryabumi, John Dang, Dwarak Talupuru, Saurabh Dash, David Cairuz, Hangyu Lin, Bharat Venkitesh, Madeline Smith, Jon Ander Campos, Yi Chern Tan, Kelly Marchisio, Max Bartolo, Sebastian Ruder, Acyr Locatelli, Julia Kreutzer, Nick Frosst, Aidan Gomez, Phil Blunsom, Marzieh Fadaee, Ahmet Üstün, and Sara Hooker. 2024. Aya 23: Open weight releases to further multilingual progress. *Preprint*, arXiv:2405.15032.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *Preprint*, arXiv:2303.08112.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Yuanpu Cao, Tianrong Zhang, Bochuan Cao, Ziyi Yin, Lu Lin, Fenglong Ma, and Jinghui Chen. 2024. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. *arXiv preprint arXiv:2406.00045*.

Nuo Chen, Zinan Zheng, Ning Wu, Linjun Shou, Ming Gong, Yangqiu Song, Dongmei Zhang, and Jia Li. 2023a. Breaking language barriers in multilingual mathematical reasoning: Insights and observations. *Preprint*, arXiv:2310.20246.

Pinzhen Chen, Shaoxiong Ji, Nikolay Bogoychev, Andrey Kutuzov, Barry Haddow, and Kenneth Heafield. 2023b. Monolingual or multilingual instruction tuning: Which makes a better alpaca. *arXiv preprint arXiv:2309.08958*.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. 2018. Xnli: Evaluating cross-lingual sentence representations. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

John Dang, Arash Ahmadian, Kelly Marchisio, Julia Kreutzer, Ahmet Üstün, and Sara Hooker. 2024. Rlhf can speak many languages: Unlocking multilingual preference optimization for llms. *arXiv preprint arXiv:2407.02552*.

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. 2023. Do multilingual language models think better in english? *arXiv preprint arXiv:2308.01223*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, and Abhishek Kadian. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, Wayne Xin Zhao, Ting Song, Yan Xia, and Furu Wei. 2023. Not all languages are created equal in llms: Improving multilingual capability by cross-lingual-thought prompting. *arXiv preprint arXiv:2305.07004*.

Zixian Huang, Wenhao Zhu, Gong Cheng, Lei Li, and Fei Yuan. 2024. Mindmerger: Efficient boosting llm reasoning in non-english languages. *arXiv preprint arXiv:2405.17386*.

Tianlong Li, Xiaoqing Zheng, and Xuanjing Huang. 2024a. Open the pandora's box of llms: Jailbreaking llms through representation engineering. *arXiv preprint arXiv:2401.06824*.

Zihao Li, Yucheng Shi, Zirui Liu, Fan Yang, Ali Payani, Ninghao Liu, and Mengnan Du. 2024b. Quantifying multilingual performance of large language models across languages. *arXiv preprint arXiv:2404.11553*.

Chaoqun Liu, Wenxuan Zhang, Yiran Zhao, Anh Tuan Luu, and Lidong Bing. 2024. Is translation all you need? a study on solving multilingual tasks with large language models. *arXiv preprint arXiv:2403.10258*.

Dawn Lu and Nina Rimsky. 2024. Investigating bias representations in llama 2 chat via activation steering. *Preprint*, arXiv:2402.00402.

nostalgebraist. 2020. interpreting gpt: the logit lens.

Ayomide Odumakinde, Daniel D'souza, Pat Verga, Beyza Ermis, and Sara Hooker. 2024. Multilingual arbitrage: Optimizing data pools to accelerate multilingual progress. *arXiv preprint arXiv:2408.14960*.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2024. Steering llama 2 via contrastive activation addition. *Preprint*, arXiv:2312.06681.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2023. The linear representation hypothesis and the geometry of large language models. *arXiv preprint arXiv:2311.03658*.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2695–2709, Singapore. Association for Computational Linguistics.

Libo Qin, Qiguang Chen, Yuhang Zhou, Zhi Chen, Yinghui Li, Lizi Liao, Min Li, Wanxiang Che, and Philip S Yu. 2024. Multilingual large language model: A survey of resources, taxonomy and frontiers. *arXiv preprint arXiv:2404.04925*.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. 2022. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, Amélie Héliou, Andrea Tacchetti, Anna Bulanova, Antonia Paterson, Beth Tsai, Bobak Shahriari, Charline Le Lan, Christopher A. Choquette-Choo, Clément Crepy, Daniel Cer, Daphne Ippolito, David Reid, Elena Buchatskaya, Eric Ni, Eric Noland, Geng Yan, George Tucker, George-Christian Muraru, Grigory Rozhdestvenskiy, Henryk Michalewski, Ian Tenney, Ivan Grishchenko, Jacob Austin, James Keeling, Jane Labanowski, Jean-Baptiste Lespiau, Jeff Stanway, Jenny Brennan, Jeremy Chen, Johan Ferret, Justin Chiu, Justin Mao-Jones, Katherine Lee, Kathy Yu, Katie Millican, Lars Lowe Sjoesund, Lisa Lee, Lucas Dixon, Machel Reid, Maciej Mikuła, Mateo Wirth, Michael Sharman, Nikolai Chinaev, Nithum Thain, Olivier Bachem, Oscar Chang, Oscar Wahltinez, Paige Bailey, Paul Michel, Petko Yotov, Rahma Chaabouni, Ramona Comanescu, Reena Jana, Rohan Anil, Ross McIlroy, Ruibo Liu, Ryan Mullins, Samuel L Smith, Sebastian Borgeaud, Sertan Girgin, Sholto Douglas, Shree Pandya, Siamak Shakeri, Soham De, Ted Klimenko, Tom Hennigan, Vlad Feinberg, Wojciech Stokowiec, Yu hui Chen, Zafarali Ahmed, Zhitao Gong, Tris Warkentin, Ludovic Peran, Minh Giang, Clément Farabet, Oriol Vinyals, Jeff Dean, Koray Kavukcuoglu, Demis Hassabis, Zoubin Ghahramani, Douglas Eck, Joelle Barral, Fernando Pereira, Eli Collins, Armand Joulin, Noah Fiedel, Evan Senter, Alek Andreev, and Kathleen Kenealy. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Qwen Team. 2024. Introducing qwen1.5.

Jörg Tiedemann. 2020. The tatoeba translation challenge – realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Haoran Wang and Kai Shu. 2024. Trojan activation attack: Red-teaming large language models using activation steering for safety-alignment. *Preprint*, arXiv:2311.09433.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in english? on the latent language of multilingual transformers. *arXiv preprint arXiv:2402.10588*.

Haoning Wu, Zicheng Zhang, Weixia Zhang, Chaofeng Chen, Liang Liao, Chunyi Li, Yixuan Gao, Annan Wang, Erli Zhang, Wenxiu Sun, Qiong Yan, Xiongkuo Min, Guangtao Zhai, and Weisi Lin. 2023. Q-align: Teaching lmms for visual scoring via discrete text-defined levels. *Preprint*, arXiv:2312.17090.

Dongkeun Yoon, Joel Jang, Sungdong Kim, Seungone Kim, Sheikh Shafayat, and Minjoon Seo. 2024. Langbridge: Multilingual reasoning without multilingual supervision. *arXiv preprint arXiv:2401.10695*.

Wenxuan Zhang, Sharifah Mahani Aljunied, Chang Gao, Yew Ken Chia, and Lidong Bing. 2023. M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models. *Preprint*, arXiv:2306.05179.

Yuanchi Zhang, Yile Wang, Zijun Liu, Shuo Wang, Xiaolong Wang, Peng Li, Maosong Sun, and Yang Liu. 2024. Enhancing multilingual capabilities of large language models through self-distillation from resource-rich languages. *Preprint*, arXiv:2402.12204.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024. How do large language models handle multilingualism? *Preprint*, arXiv:2402.18815.

Chengzhi Zhong, Fei Cheng, Qianying Liu, Junfeng Jiang, Zhen Wan, Chenhui Chu, Yugo Murawaki, and Sadao Kurohashi. 2024. Beyond english-centric llms: What language do multilingual language models think in? *Preprint*, arXiv:2408.10811.

Wenhao Zhu, Shujian Huang, Fei Yuan, Shuaijie She, Jiajun Chen, and Alexandra Birch. 2024. Question translation training for better multilingual reasoning. *arXiv preprint arXiv:2401.07817*.

Wenhao Zhu, Yunzhe Lv, Qingxiu Dong, Fei Yuan, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Extrapolating large language models to non-english by aligning languages. *arXiv preprint arXiv:2308.04948*.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. 2023. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*.

A Learning the Steering Vector

In the first scenario, we utilize previous work Cao et al. (2024) that applied Direct Preference Optimization (DPO) methods to construct the steering vector. Specifically, Optimizing v increases the probability of generating responses that align with the desired language behavior (e.g., English) while reducing the likelihood of responses associated with the opposite behavior (e.g., the target language). In this case, the contrast is defined between two language pairs: the English response R_t and the target language response R_O.

$$\min_v -\mathbb{E}_{d\sim\mathcal{U},(q,r_T,r_O)\sim\mathcal{D}}\left[\log\sigma\left(d\beta\log\frac{\pi_{L+1}(r_T|A_L(q)+dv)}{\pi_{L+1}(r_T|A_L(q))} - d\beta\log\frac{\pi_{L+1}(r_O|A_L(q)+dv)}{\pi_{L+1}(r_O|A_L(q))}\right)\right]. \tag{3}$$

Where: v is the learnable steering vector, σ represents the logistic function.β controls the deviation from the original model. $\pi_{L+1}(\cdot|A_L(q))$ denotes the model's response from layer $L+1$, given the activation $A_L(q)$ at layer L for the input question q.The term d flips the optimization direction:

- $d = 1$, the steering vector is optimized towards the English behavior r_T.
- If $d = -1$, the steering vector is optimized towards the opposite behavior r_O.

By optimizing this bi-directional objective, the steering vector v is trained to align with either the desired target behavior or its reverse, depending on the directional coefficient d. This approach ensures that both language behaviors target and opposite are captured effectively, enhancing the model's ability to differentiate between them with precision.

A.1 Algorithms

Algorithm 1 BiPO Steering Vector Learning

Require: Pretrained LLM M, bilingual corpus $\mathcal{D} = \{(q_i, q_i^{\text{en}})\}$, layer L, learning rate η, epochs T
Ensure: Steering vector $v \in \mathbb{R}^d$

```
Initialize v ← 0
for e ← 1, ..., T do
    for all (q, q^en) ∈ D do
                                  ▷ 1. Extract hidden activations at layer L
        h ← HiddenState(M, q, L)
        h^en ← HiddenState(M, q^en, L)
                                  ▷ 2. Inject steering vector
        h̃ ← h + v
                                  ▷ 3. Compute logits from both activations
        ℓ ← Logits(M, h̃)
        ℓ^en ← Logits(M, h^en)
                                  ▷ 4. Direct Preference Optimization (DPO) loss
            L_DPO = −E_{t∼V}[log σ(ℓ_t^en − ℓ_t)]   (see App. A, eq.3)
                                  ▷ 5. Gradient-step update
        v ← v − η ∇_v L_DPO
    end for
end for
Return v
```

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{t\sim\mathcal{V}}\left[\log\sigma\left(\ell_t^{\text{en}} - \ell_t\right)\right]$$

Algorithm 2 MSE Steering Vector Learning

Require: Pretrained LLM M, bilingual corpus $\mathcal{D} = \{(q_i, q_i^{\text{en}})\}$, layer L, learning rate η, epochs T
Ensure: Steering vector $v \in \mathbb{R}^d$

Initialize $v \leftarrow \mathbf{0}$
for $e \leftarrow 1, \ldots, T$ **do**
  for all $(q, q^{\text{en}}) \in \mathcal{D}$ **do**
    ▷ 1. Extract hidden activations at layer L
    $h \leftarrow \text{HiddenState}(M, q, L)$
    $h^{\text{en}} \leftarrow \text{HiddenState}(M, q^{\text{en}}, L)$
    ▷ 2. Inject steering vector
    $\tilde{h} \leftarrow h + v$
    ▷ 3. Compute Mean-Squared Error loss

$$\mathcal{L}_{\text{MSE}} = \frac{1}{d} \|\tilde{h} - h^{\text{en}}\|_2^2$$

    ▷ 4. Gradient-step update
    $v \leftarrow v - \eta \nabla_v \mathcal{L}_{\text{MSE}}$
  end for
end for
Return v

A.2 Other learning methods

Effectively learning a manifold that encapsulates the feature representations between languages is vital for bridging the distributional gap across linguistic boundaries. While prior approaches (Cao et al., 2024; Zou et al., 2023), such as PCA and calculating the mean difference between constructive activations (CAA), have been shown to shift activation distributions, they fall short in accurately capturing essential features in multilingual contexts. In contrast, advanced methods like BiPO excel by leveraging a dynamic feedback loop during the manifold learning process, enabling them to better align multilingual representations. Figure 7 highlights the performance of various models across diverse tasks, underscoring the effectiveness of this approach.

Figure 7: demonstrate that using learnable steering vectors surpasses PCA and the Mean Difference approaches across all tasks on two models: Aya32-8B and LLama2-7B.

B SFT Vs Steering: Problem Setup and Notation

Let $\mathcal{M}$ represent the base LLM and $\mathcal{M}^*$ denote the fine-tuned version trained on an instruction dataset $\mathcal{D}$, where $\mathcal{D} = (Q_i, A_i)_{i=1}^n$ consists of question-answer pairs. To analyze the mechanisms of fine-tuning, we model the transformation of each layer l as:

$$H_l(x) = h_l(x) + S_l(x) \tag{4}$$

where:

- $h_l(x)$ represents the original layer l activation for input x
- $S_l(x) \in \mathbb{R}^d$ is a learnable parameter matrix that modulates the activation in the residual stream
- d is the dimensionality of the hidden state

For each $(Q, A) \in \mathcal{D}$, H_l is optimized via the loss function:

$$\mathcal{L}(\mathcal{M}^(Q), A) = -\sum_{t=1}^{T} \log P(a_t|a_{<t}, Q; \theta^) \tag{5}$$

where:

- θ^* represents the fine-tuned model parameters
- a_t is the t-th token in the answer A
- T is the length of the answer

In contrast, the steering approach learns a single steering vector $v \in \mathbb{R}^d$ that modifies activations across all layers:

$$H_l(x) = h_l(x) + \alpha v \tag{6}$$

where v is the learned steering direction , α is a scaling coefficient that controls the magnitude of steering ,The same v is applied across different (Q, A) pairs

C Hyperparameters

Training Steering Vectors: For all models, we followed the authors' Cao et al. (2024) configurations, setting $\beta = 0.1$, using the AdamW optimizer with a learning rate of 5×10^{-4}, and applying a weight decay of 0.05. The batch size was set to 1, and we utilized a cosine learning rate scheduler with 100 warmup steps. The number of epochs was set to 1 for all models, except for certain languages in LLama2 and Aya23-8B, where it was increased to 3 epochs. For the MSE method, we used a learning rate of 1×10^{-8} and varied the number of epochs in the range [3, 5, 8, 12]. Mean Squared Error (MSE) was used as the loss function, and cosine similarity was employed to evaluate the similarity between raw activations during training.

For the supervised fine-tuning described in section 7, we trained the models on the same training datasets for 5 epochs, using a learning rate of 1×10^{-3}, a weight decay of 0.001, and a warmup ratio of 0.05. The batch size was set to 16, and we utilized a cosine learning rate scheduler with the AdamW optimizer.

D Larger LLMs Exhibit Consistent Behavior

To address translation loss misalignment in larger language models, we extended our evaluation of steering approaches to larger architectures. Due to computational constraints, we tested only LLama2-13B on the MGSM task. Table 4 indicates that these larger models follow the same trend of performance improvements across different languages, mirroring the behavior observed in smaller models.

MGSM	Es	Fr	Ru	De	Ja	zh	Avg
Llama2-13B							
Basic Prompt	33.6	30.0	28.0	30.8	18.0	26.4	27.8
Google-Trans	**39.2**	**35.2**	**36.8**	36.4	**35.6**	**36.4**	**36.6**
NLLB	35.2	33.6	32.0	34.0	20.0	28.0	30.4
5@shots	35.2	32.8	26.8	33.2	18.4	23.6	28.3
XLT	33.6	30.4	30.8	27.6	25.2	29.6	29.5
Bipo-method	$36.8_{(+3.2)}$	$33.2_{(+3.2)}$	$31.6_{(+3.6)}$	$35.2_{(+4.4)}$	$26.8_{(+8.8)}$	$29.2_{(+2.8)}$	$32.1_{(+4.3)}$
MSE-method	$32.4_{(-1.2)}$	$34.8_{(+4.8)}$	$34.0_{(+6)}$	$35.2_{(+4.4)}$	$24.4_{(+6.4)}$	$30.0_{(+3.6)}$	$31.8_{(+4.0)}$

Table 4: Results of the MGSM Task Evaluated on the Llama2-13B Model Across Diverse Languages

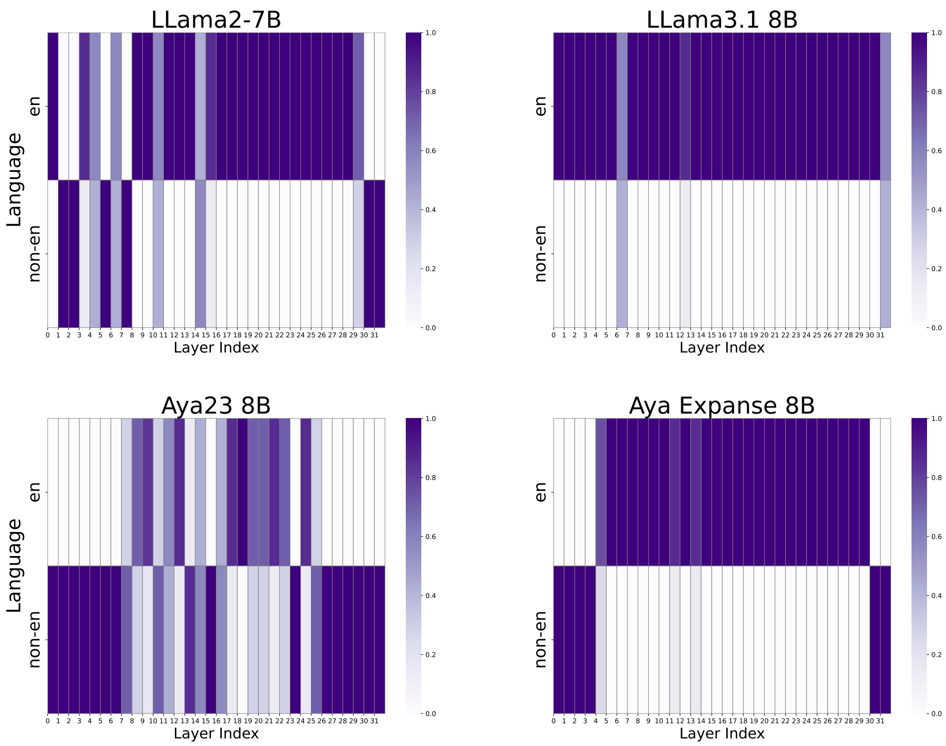


Figure 8: illustrates the processing of multilingual tokens in models of varying capabilities within the same family. LLama3.1 demonstrates a strong alignment of tokens into English-aligned representations, whereas LLama2 struggles with this. Similarly, Aya-Expanse exhibits robust token alignment, attributed to RLHF techniques, while Aya23 shows weaker alignment.

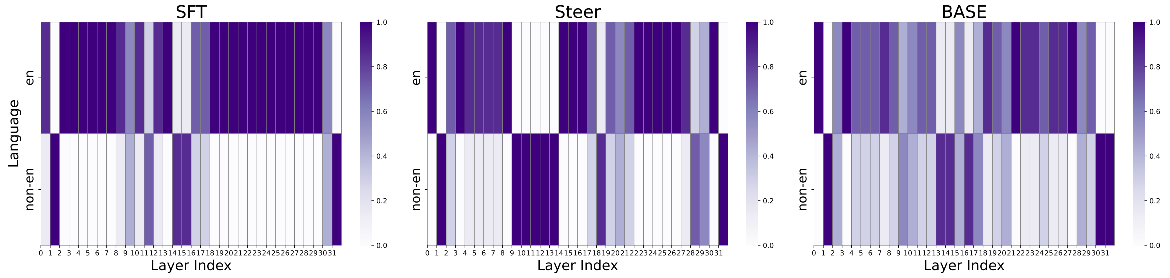


Figure 9: : Distribution of non-English token values across model layers at three different stages: pre-fine-tuning (base model), post-fine-tuning(SFT), and after applying steering at a specific layer. The results demonstrate that both fine-tuning and steering exhibit similar behavior, aligning non-token values more closely with English token distributions.

Models	Lang-Rep	Fr	Ru	Ja	Es	Zh	De
Aya23-8B	Fr	-	34.3	25.6	40.0	27.6	36.0
	Es	32.0	34.4	25.6	-	27.6	36.0
	De	32.0	34.4	25.6	40.0	27.6	-
	En	**38.0**	**41.2**	**34.8**	**44.4**	**32.8**	**40.4**
Llama2-7B	Fr	-	23.2	18.4	24.4	20.4	25.2
	Es	24.4	22.8	17.6	-	21.2	**26.0**
	De	**26.0**	21.6	17.6	24.4	**22.0**	-
	En	25.6	**23.2**	**20.8**	**25.2**	21.6	24.8

Table 5: The table highlights the selection of high-resource languages, such as French, Spanish, and German, as agnostic languages within the representation space of LLMs. The results indicate that English remains the most dominant language in this space. Other high-resource languages achieve comparable results, suggesting that their representations are distributed with similar likelihoods within the shared representation space.

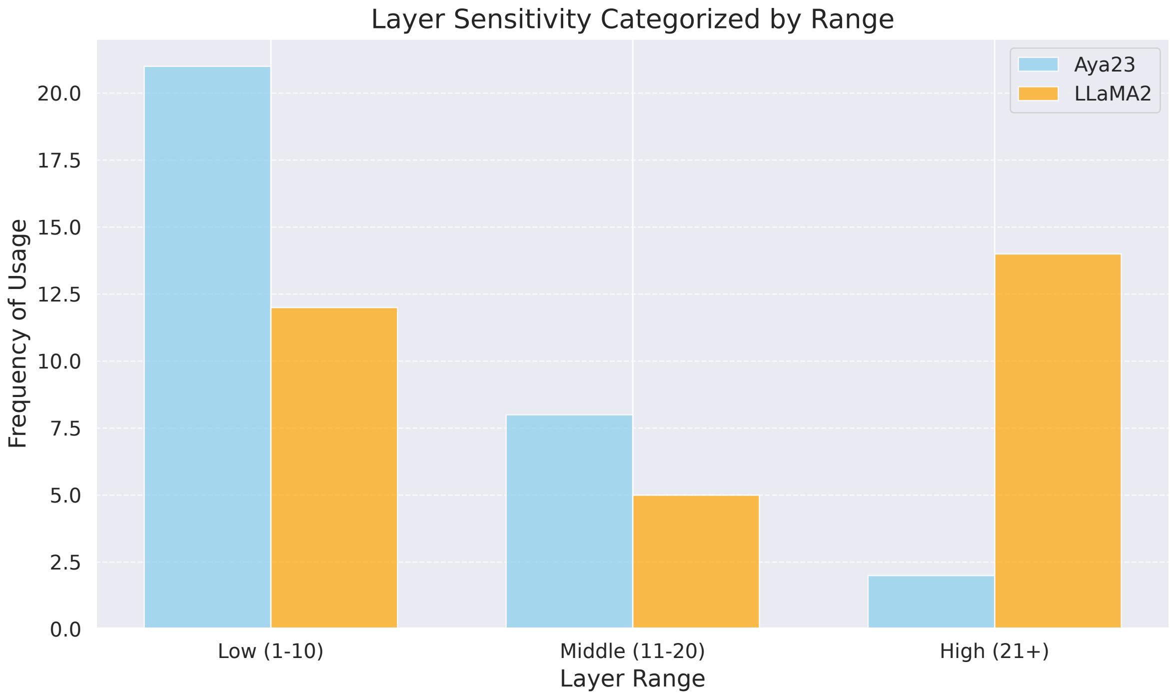
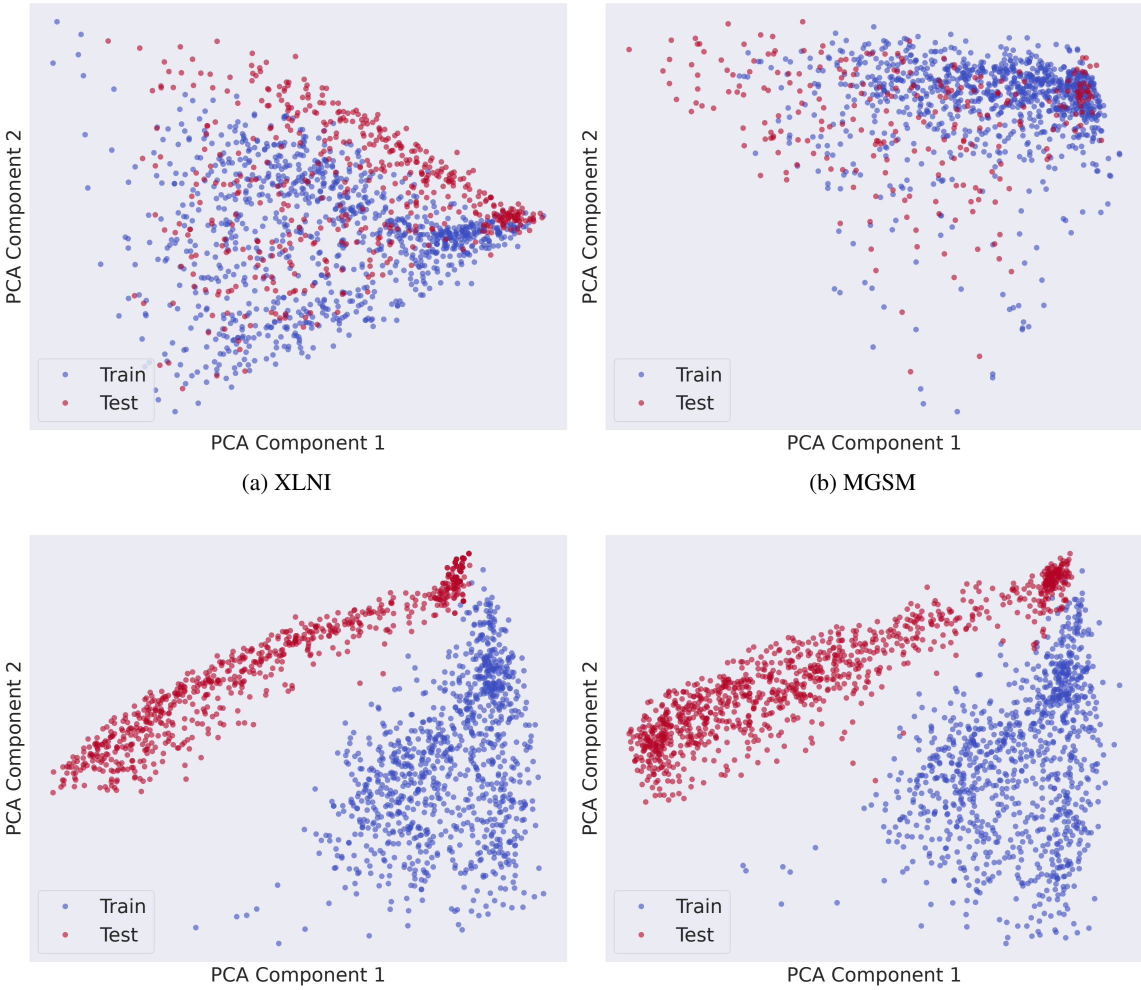


Figure 10: illustrates the layers most sensitive to probing across two models. Aya23 demonstrates high sensitivity in the initial layers but exhibits reduced performance in the middle and later layers. In contrast, LLama2 experiences a notable drop in performance in the middle layers, with improved results in the later layers. Additionally, the initial layers of LLama2 perform better for low- and medium-resource languages.



(a) XLNI (b) MGSM

(c) M3EXAM (d) MMLU

Figure 11: The sub-figures illustrate the distribution of the training and testing datasets across various tasks, emphasizing that steering approaches perform effectively when the testing dataset's distribution closely aligns with the training dataset's distribution but show limited improvement when the two distributions differ significantly.

Llama3-8B					
Methods	***MGSM***	***XCOPA***	***XLNI***	***M3EXAM***	***MMLU***
Basic Prompt	62.0	66.7	63.2	51.6	50.7
Google-Trans	70.7	79.3	65.8	54.5	**58.2**
NLLB	60.0	63.4	63.4	23.9	40.7
5@Shot	55.6	63.5	27.6	24.1	26.0
XLT	26.9	56.9	55.0	39.2	33.7
BIPO-steer	**67.0**$_{(+5.0)}$	**75.0**$_{(+8.3)}$	64.3$_{(+1.1)}$	**55.3**$_{(+3.7)}$	52.8$_{(+2.1)}$
MSE-steer	62.8$_{(+0.8)}$	68.4$_{(+1.7)}$	64.0$_{(+0.8)}$	53.0$_{(+1.4)}$	50.6$_{(-0.1)}$
Gemma-7B					
Basic Prompt	27.3	66.2	46.4	37.3	39.6
Google-Trans	**37.4**	83.1	51.0	45.4	**47.0**
NLLB	29.8	65.4	50.0	23.0	33.8
5@Shot	12.2	42.2	39.6	20.2	22.0
XLT	28.7	49.8	49.9	28.1	26.5
BIPO-steer	30.0$_{(+2.7)}$	68.8$_{(+2.6)}$	**51.9**$_{(+5.5)}$	**45.7**$_{(+8.4)}$	41.1$_{(+1.5)}$
MSE-steer	27.6$_{(+0.3)}$	63.4$_{(-2.8)}$	47.9$_{+1.5}$	37.5$_{(+0.2)}$	41.1$_{(+1.5)}$
Qwen1.5-7B					
Basic Prompt	49.2	70.1	66.0	58.3	42.1
Google-Trans	52.6	**79.3**	65.4	57.2	46.1
NLLB	47.4	63.4	64.3	31.8	31.3
5@Shot	46.8	74.6	65.0	46.7	48.1
XLT	49.4	64.4	55.6	54.2	39.7
BIPO-steer	**52.8**$_{(+3.6)}$	73.8$_{(+3.7)}$	67.8$_{(+1.8)}$	58.7$_{(+0.4)}$	42.9$_{(+0.8)}$
MSE-steer	50.5$_{(+1.3)}$	70.7$_{(+.7)}$	**68.7** $_{(+2.7)}$	**59.7**$_{(+1.4)}$	**46.4**$_{(+4.3)}$

Table 6: Presents the average accuracy across various tasks for Three models: Llama3-8B , gemma-7Ba and Qwen1.5-7B . The evaluation spans over 10 languages, covering high, medium, and low-resource levels. (+) symbol indicates the average improvement compared to the Basic Prompt baseline.

E Examples of different tasks

Examples of model-generated responses under both original and steered conditions when prompted with questions in different languages. The correct answers are highlighted in green, while incorrect answers are marked in **red** for clarity.

Question	マンディはベネディクトに$100借りがあります。彼らは月2%の利子がつくことに同意しました。マンディが3か月後に返できるとすると、彼女はベネディクトにいくらえばいいですか? Let's think step by step.
Original generation	Mandy borrowed $100 from Benedict with an agreement to pay 2% interest per month. 1. Calculate the total interest due after 3 months: $100 x 2% x 3 months = $6 So, the total amount Mandy owes Benedict after 3 months is $106 ($100 + $6). 2. Calculate the monthly payment amount: To find the monthly payment amount, we need to divide the total amount owed by the number of months left to repay the loan. In this case, there are 3 months left to repay the loan, so: $106 ÷ 3 months = $35.33 So, Mandy needs to pay **$35.33** per month to Benedict.
Steered generation	1. Mandi borrowed $100 from Benedict with an agreement to pay 2% interest per month. 2. The loan is to be repaid 3 months from now. So, we need to calculate the total amount that Mandi needs to pay to Benedict to settle the loan. First, let's calculate the interest for the 3 months: 2% per month x 3 months = 6% interest Now, let's calculate the total amount that Mandi needs to pay: Original amount borrowed: $100 Interest: $6 (6% of $100) Total amount to be paid: $106 So, Mandi needs to pay **$106** to Benedict to settle the loan.

Question	Les canes de Janet pondent 16 œufs par jour. Chaque matin, elle en mange trois au petit djeuner et en utilise quatre autres pour prparer des muffins pour ses amis. Ce qui reste, elle le vend quotidiennement au march fermier, au prix de 2 $ l'œuf de cane frais. Combien (en dollars) gagne-t-elle chaque jour au march fermier ? Let's think step by step.
Original generation	Janet's chickens lay 16 eggs per day. She eats 3 eggs for breakfast every day. She uses 4 eggs to make muffins for her friends. So, the number of eggs she has left to sell at the farmer's market is: 16 eggs/day - 3 eggs/day = 13 eggs/day Since she sells each egg at $2, Janet earns: 13 eggs/day x $2/egg = $26/day So, Janet earns **$26** per day at the farmer's market.
Steered generation	1. Janet lays 16 eggs per day. 2. She eats 3 eggs for breakfast every day. 3. She uses 4 eggs to make muffins for her friends. 4. So, how many eggs does she have left to sell at the farmer's market? 16 eggs/day - 3 eggs/day = 13 eggs/day 13 eggs/day - 4 eggs/day (for muffins) = 9 eggs/day Janet has 9 eggs left to sell at the farmer's market every day. Now, let's calculate how much money she makes from selling those eggs: 9 eggs/day x $2/egg = $18/day So, Janet makes **$18** per day selling eggs at the farmer's market.

Question	A seguir esto questes de mltipla escolha sobre Linguagens. Por favor, d apenas a opo correta, sem quaisquer outros detalhes ou explicaes. **Por que as formigas no morrem quando postas em forno de micro-ondas?** As micro-ondas so ondas eletromagnticas com frequncia muito alta. Elas causam vibrao nas molculas de gua, e isso que aquece a comida. Se o prato estiver seco, sua temperatura no se altera. Da mesma maneira, se as formigas tiverem pouca gua em seu corpo, podem sair inclumes. J um ser humano no se sairia to bem quanto esses insetos dentro de um forno de micro-ondas superdimensionado: a gua que compe 70% do seu corpo aqueceria. Micro-ondas de baixa intensidade, porm, esto por toda a parte, oriundas da telefonia celular, mas no h comprovao de que causem problemas para a populao humana. OKUNO, E. Disponvel em: http://revistapesquisa.fapesp.br. Acesso em: 11 dez. 2013. Os textos constroem-se com recursos lingusticos que materializam diferentes propsitos comunicativos. Ao responder pergunta que d ttulo ao texto, o autor tem como objetivo principal: A. defender o ponto de vista de que as ondas eletromagnticas so inofensivas. B. divulgar resultados de recentes pesquisas cientficas para a sociedade. C. apresentar informaes acerca das ondas eletromagnticas e de seu uso. D. alertar o leitor sobre os riscos de usar as micro-ondas em seu dia a dia. E. apontar diferenas fisiolgicas entre formigas e seres humanos. Responder:
Original generation	A resposta correta a alternativa: E.
Steered generation	The correct answer is C.