Efficient Alignment: Improving Multilingual Models Through Lightweight Steering

Anonymous ACL submission

Abstract

This paper investigates how Large Language Models (LLMs) represent non-English tokens—a question that remains underexplored despite recent progress. We propose a lightweight intervention method using *representation steering*, where a learned vector is added to the residual stream at a single model layer to enhance multilingual performance. Through extensive experiments across seven competitive baselines—including prompt optimization, supervised fine-tuning (SFT), in-context learning, cross-lingual transfer, and translation-based methods—we show that our approach consistently outperforms most alternatives. In particular, it achieves performance on par with production-grade translation systems while requiring far fewer resources. We further explore the complementarity between our method and SFT, demonstrating that steering offers a direct, efficient way to realign internal representations. These findings underscore the potential of activation-level interventions as a powerful tool for improving the multilingual capabilities of LLMs.

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 explored various strategies to enhance the multilingual proficiency of LLMs. These include cross-lingual fine-tuning (Qin et al., 2023), instruction alignment via code-switching (Huang et al., 2023), and chain-of-thought prompting in multiple languages (Shi et al., 2022). Other approaches focus on mapping representations between models, such as MindMerger, which integrates an external LLM's linguistic knowledge by learning a mapping between representation spaces (Huang et al., 2024). Despite these advances, investigations into the internal mechanisms of multilingual processing remain limited. The few studies in this area suggest that LLMs often default to translating non-English inputs into English representations within their intermediate layers (Wendler et al., 2024; Zhao et al., 2024).

Closely related to our work, recent research (Wang et al., 2024) has used representation steering to align hidden states between English and a target language. This was achieved by learning a steering vector through a least-squares optimization and applying it across all layers of the model. While effective, this method modifies the model's representations globally. This leaves a critical question unexplored: *can multilingual alignment be achieved more efficiently by targeting only a single, specific layer, and what does this reveal about the model's internal structure?* In this paper, we investigate this question from a mechanistic interpretability perspective. We propose a method that first learns a transformation manifold mapping English representations to a target language. This mapping is then applied as a steering vector to the activations of only a single layer during inference, without any fine-tuning. This lightweight approach is more efficient and less disruptive to the base model's capabilities, as illustrated in Figure 1. By demonstrating that our single-layer intervention parallels the effects of full fine-tuning, we provide new insights into how LLM representations can be precisely and efficiently optimized for multilingual tasks. Our key contributions are as follows:

- We propose and validate a method to enhance the multilingual capabilities of LLMs by steering the representations of a single layer, using a learned alignment with English.
- We demonstrate that a single steering vector can be shared across structurally similar languages [1] , enabling zero-shot cross-linguistic transfer without language-specific fine-tuning.
- Our method significantly surpasses the performance of the NLLB translation baseline and achieves results competitive with Google Translate across multiple datasets.

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; 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 underutilize 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., 2024). 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.

Inference Time Intervention: using steering vectors is an established technique in the field of model editing (Li et al., 2023; Panickssery et al., 2023). These methods modify model behavior by directly manipulating internal states; for example, vectors can be added to a model's residual stream to improve truthfulness (Wang et al., 2025) or removed from its hidden states to induce refusal behaviors (Arditi et al., 2024). However, the application of these powerful steering techniques in multilingual settings remains largely unexplored.

3 Background

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.

Problem Formulation. Let a transformer model process a sequence of input tokens

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

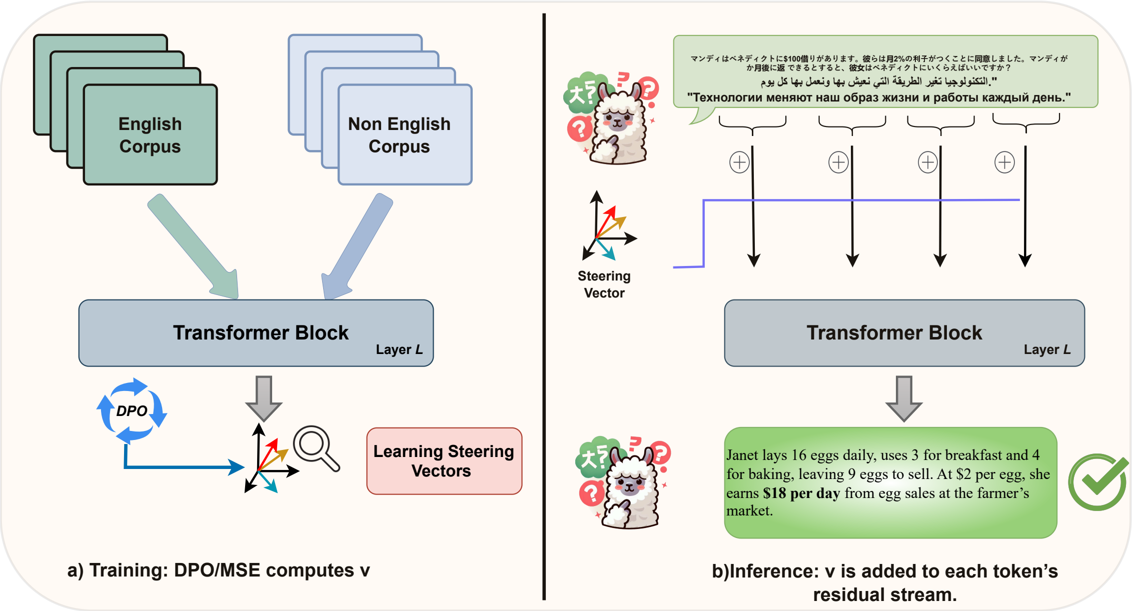

English Corpus
Non English Corpus
Transformer Block
Layer *L*
DPO
Learning Steering Vectors
a) Training: DPO/MSE computes v
マンディはベネディクトに$100借りがあります。彼らは月2%の利子がつくことに同意しました。マンディが
か月後に返 できるとすると、彼女はベネディクトにいくらえばいいですか？
التكنولوجيا تغير الطريقة التي نعيش بها ونعمل بها كل يوم."
"Технологии меняют наш образ жизни и работы каждый день."
Steering Vector
Transformer Block
Layer *L*
Janet lays 16 eggs daily, uses 3 for breakfast and 4 for baking, leaving 9 eggs to sell. At $2 per egg, she earns **$18 per day** from egg sales at the farmer's market.
b)Inference: v is added to each token's residual stream.

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.

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	0.51	0.46	24.4	23.3	0.25	15.8 ↑
Aya23-8B											
Basic Prompt	40.0	25.6	34.4	0.64	27.6	1.0	13.1	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	.01	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.

$t = (t_1, t_2, \ldots, t_n) \in \mathcal{V}^n$, producing a sequence of output probability distributions $y = (y_1, y_2, \ldots, y_n) \in \mathbb{R}^{n\times|\mathcal{V}|}$. Denote by $x_i^{(l)}(t) \in \mathbb{R}^d$ the residual stream activation at token position i at the beginning of layer l. The residual stream is initialized via token embeddings: $x_i^{(1)} = \text{Embed}(t_i)$.

Each transformer layer updates the residual stream with attention and MLP components:

$$\tilde{x}_i^{(l)} = x_i^{(l)} + \text{Attn}^{(l)}(x_1^{(l)}, \ldots, x_n^{(l)}) \quad (1)$$

$$x_i^{(l+1)} = \tilde{x}_i^{(l)} + \text{MLP}^{(l)}(\tilde{x}_i^{(l)}). \quad (2)$$

After L layers, the model computes the final logits $\text{logits}_i = \text{Unembed}(x_i^{(L+1)}) \in \mathbb{R}^{|\mathcal{V}|}$, followed by softmax to obtain the output distribution:

$$y_i = \text{softmax}(\text{logits}_i) \in \mathbb{R}^{|\mathcal{V}|}. \quad (3)$$

Activation Extraction and Alignment. We denote the post-layer-L residual activations for English and a target language as $x_{\text{en}}^{(L+1)}$ and $x_{\text{target}}^{(L+1)}$, respectively. Our goal is to align the target language representations with their English counterparts by applying an additive transformation:

$$x_{\text{altered}}^{(L+1)} = x_{\text{target}}^{(L+1)} + v^{(L)}, \quad (4)$$

where $v^{(L)}$ is a learned steering vector specific to layer L. We propose two methods for learning the alignment vector $v^{(L)}$:

1. **Direct Preference Optimization (DPO).** Inspired by recent work, we apply Direct Preference Optimization (DPO) to learn $v^{(L)}$ by aligning target language representations with English ones while explicitly disaligning from original target activations. Unlike conventional approaches that compute the mean difference between activations (Panickssery et al., 2024; Wang and Shu, 2024) or use PCA to extract principal directions (Annah and shash42, 2023), DPO learns a direction that better captures the bidirectional preference relationship between $x_{\text{en}}^{(L+1)}$ and $x_{\text{target}}^{(L+1)}$. This leads to improved multilingual alignment. See Appendix A for mathematical details.

2. **Loss-Based Activation Alignment** Following the methodology (Park et al., 2023), which suggests that representations in different languages may be linearly mappable, we also learn $v^{(L)}$ by minimizing the mean squared error between the aligned and English representations:

$$\mathcal{L}_{\text{MSE}} = \text{MSE}\left(x_{\text{en}}^{(L+1)}, x_{\text{altered}}^{(L+1)}\right). \quad (5)$$

Intervention. After learning the steering vector $v^{(L)}$, we apply an activation intervention by modifying the residual stream of a new target-language prompt at layer L. Specifically, given a new activation $x_{\text{target}}^{(L+1)}$ at layer $L+1$, we compute the intervened activation as:

$$x_{\text{altered}}^{(L+1)} = x_{\text{target}}^{(L+1)} + v^{(L)}. \quad (6)$$

This altered activation is then propagated through the remaining layers of the model, allowing us to observe how the intervention affects the model's output distribution. The goal is to steer the model's internal representations of the target-language input to better align with English-like behavior.

4 Experiments

In this section, we outline the experimental setup necessary for the evaluations presented in Section 4.1 and the corresponding results discussed in Section 4.2.

4.1 Experimental Setup

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, **XNLI** (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), a human exam benchmark testing comprehensive language understanding. This diverse suite ensures a robust evaluation across linguistic competencies.

Baselines: We compared seven 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.

[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 XNLI, respectively.

- **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.

- **Supervised Fine-Tuning (SFT)**: This approach fine-tunes all model parameters on a non-English dataset and evaluates performance on downstream tasks.

4.2 Results

Our evaluation demonstrates in Table 2 that **activation-based steering is a highly efficient and scalable approach for improving multilingual language models**. Unlike resource-intensive methods such as Supervised Fine-Tuning (SFT), which require task-specific data, prolonged training, and careful hyperparameter tuning, our proposed techniques achieve competitive performance at a fraction of the computational and operational cost. Notably, DPO yields a substantial 26.7% improvement over SFT, underscoring the effectiveness of targeted, activation-level interventions.

The advantages of this lightweight method are evident across a wide range of open-source baselines. Both DPO and MSE steering produce marked improvements over standard prompting strategies and even advanced cross-lingual transfer (XLT) prompts. The most significant gains are observed against a 5-shot in-context learning (ICL) baseline, where DPO achieves a 38.8% improvement. **This result highlights steering's ability to correct internal representational misalignments that ICL, despite leveraging contextual examples, fails to resolve.** Further, DPO outperforms the open-source translation model NLLB by 25.4%, demonstrating that steering is not merely a fine-tuning shortcut but a viable alternative to full translation pipelines. **It effectively aligns multilingual representations internally, without reliance on external systems.** While steering does not yet exceed the performance of proprietary systems such as Google Translate, the margin is surprisingly narrow. DPO trails Google Translate by just 3.08%, illustrating that internal, model-native interventions can approach the performance of large-scale production-grade translation APIs. This finding is particularly promising given the simplicity, interpretability, and deployability of the proposed steering method.

Across all experiments, **DPO consistently outperforms MSE-based steering.** We attribute this superiority to the directional optimization signal embedded in the DPO framework, which not only penalizes misalignment but actively guides the model toward improved representations. In contrast, the MSE objective quantifies error magnitude without providing gradient directionality, making optimization less efficient and less targeted. This fundamental distinction explains DPO's effectiveness as a principled and robust method for steering multilingual behavior in pretrained language models. Moreover, the steering approach yields performance gains as model size increases, suggesting that larger language models benefit more from targeted activation interventions. We provide a detailed analysis of this trend in the Appendix D

5 Analysis

In this section, we analyze the proposed methods from various perspectives: Can we measure the quality of translation? What are the Challenges of steering? How transferable is the direction? And finally, which languages dominate the model's representation space?

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

Our analysis reveals a critical factor in a model's multilingual performance: the quality of its internal translation. When a model fails to accurately represent a language internally, it leads to information loss and significant performance gaps. We quantify this internal translation quality by measuring how closely a language's internal representation aligns with English, a proxy for how well it has been integrated into the model's core space. Unsurprisingly, this alignment is directly tied to the volume of pre-training data for each language. Models like **LLaMA2 clearly illustrate this principle: high-resource languages with ample data (French, German)** show strong alignment with English, while low-resource languages (Thai, Telugu) exhibit much weaker connections. While specialized multilingual models like **Aya23 improve this alignment for some under-represented languages**, significant challenges remain see(Figure 2) In contrast, Qwen1.5 highlights a different

Methods	*MGSM*	*XCOPA*	*XNLI*	*M3EXAM*	*MMLU*
Base	38.1	66.4	54.3	44.9	41.8
Google trans	44.6	75.5	57.1	49.8	48.7
NLLB	38.4	61.2	55.4	25.2	32.8
5 shot	32.5	58.9	41.3	29.2	30.2
XLT	30.4	46.1	51.7	37.7	30.1
SFT	34.7	55.2	46.1	39.9	35.4
DPO-Steer	$42.2_{+(4.1)}$	$71.2_{+(4.8)}$	$59.5_{+(5.2)}$	$50.1_{+(5.2)}$	$44.1_{+(2.3)}$
MSE-Steer	$39.9_{+(1.8)}$	$66.9_{+(0.5)}$	$56.0_{+(1.7)}$	$46.6_{+(1.7)}$	$44.3_{+(2.5)}$

Table 2: Average results across five open-source models in 14 languages. '+' indicates an improvement over the Base Prompts.

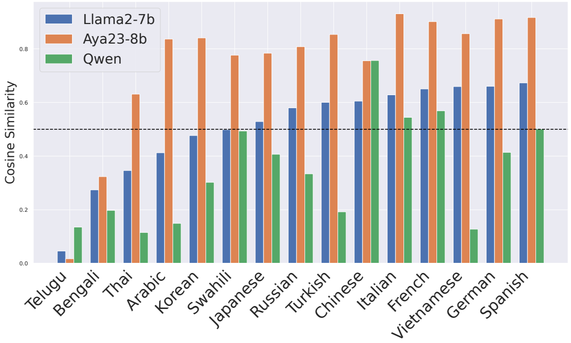


Figure 2: Similarity scores between language and English representations across models. The dashed line (threshold = 0.5) shows high-resource languages above and mid-/low-resource languages below it.

risk: its training was so dominated by Chinese data that it often defaults to translating other languages into Chinese internally, struggling with alignment for most other languages. Ultimately, these findings confirm that languages with weaker internal representations are fundamentally disadvantaged. This "representational misalignment" is not just a technical artifact; it is a direct cause of performance disparities across different language groups.

5.2 Challenges in Steering Vector Generalization

While effective, the steering vector approach faces two core limitations: it is distribution-sensitive and lacks contextual precision.

It is distribution-sensitive because its effectiveness depends heavily on the training distribution. As shown in Figure 8, performance drops significantly on out-of-distribution tasks. It lacks contextual precision because it applies a single, static correction across all inputs, ignoring prompt-specific variations. This uniform adjustment fails to capture the rich contextual variability inherent in natural language. These limitations suggest a clear direction for future work: developing context-aware steering mechanisms that adapt dynamically to each prompt, as explored by (Tran et al., 2025).

5.3 Is the steering vector transferable across languages?

Building on prior work by (Cao et al., 2024), we examine whether a steering vector trained on one language can transfer effectively to another. Our results **indicate that transferability is feasible, but largely limited to languages within the same linguistic family, likely due to shared representational structures.** As shown in Figure 3, a vector trained on a source language consistently improves performance when applied to a related target language. For example, a vector trained on Spanish (Es) transfers well to German (De), French (Fr), and Russian (Ru) all Indo-European languages. Similarly, transfers between Japanese (Ja) and Chinese (Zh) are effective. However, these successes also expose the method's limitations. Cross-family transfers, such as from Spanish to Japanese, are ineffective, suggesting that while the vector captures more than language-specific patterns, it lacks the abstraction needed for generalization across distant language families.

5.4 High Resource Languages are dominant in Representation Space

A common assumption is that large language models "think" in English, merely translating other languages into English-like representations for processing. Our findings challenge this notion, revealing a more nuanced reality: the primary bottleneck is not English per se, but membership in a broader set of high-resource languages. To in-

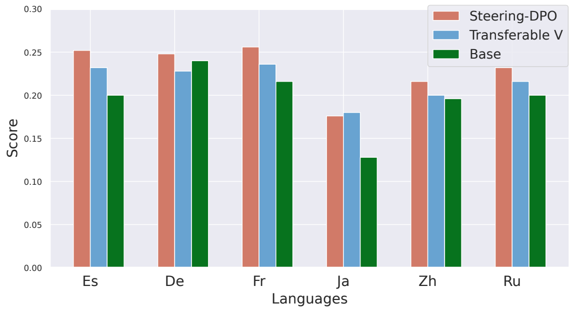


Figure 3: Scores after applying steering vectors transferred directionally between language pairs (source → target), selected based on embedding similarity: Es→De, De→Es, Fr→Es, Ja→Zh, Zh→Ja, and Ru→Es.

vestigate this, we selected three high-resource languages—Spanish, German, and French—chosen for their linguistic proximity and high representational similarity to English in the model. We then evaluated model performance on the MGSM reasoning task across these languages. Table 3 shows that : performance in Spanish, German, and French **closely matched that of English, with negligible differences.** These findings suggest that the **"English-centric**" view is overly reductive. Instead, current models appear to operate within a privileged set of high-resource languages capable of supporting complex reasoning. The central challenge for multilingual modeling is thus not merely accommodating many languages, but extending this inner circle to include low-resource languages.

6 Ablation Studies

6.1 Impact of Injection Across Model Layers

Our ablation studies reveal a critical insight: the optimal layer for steering is not universal. Instead, it is a direct reflection of a model's architecture and training data, as shown by the starkly different behaviors of Aya23 and LLaMA2. **Aya23, a model explicitly designed for multilingualism, benefits most from steering in its early layers**. Its architecture is built to quickly translate different languages into a shared, language-neutral space. By correcting errors at this early stage—before they can cascade through the network—we see significant performance gains across all tasks, including mathematical reasoning (Figure 7). In contrast, LLaMA2 presents a more complex picture. **For low-resource languages, steering the early and middle layers is highly effective**, as it helps these languages "catch up" and align with the model's dominant representations. However, for high-resource languages like English or German, this same intervention can be disruptive, interfering with already well-formed representations. Finally, across both models, steering the final layers yields little to no improvement. This suggests that by this late stage, the model has already "committed" to its interpretation in its internal representation space. Intervening here is simply too late to have a meaningful effect. This confirms that to be effective, steering must happen "upstream," before the model's final reasoning process is complete.

6.2 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. Results in Table 4 **demonstrate 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

Our findings suggest that **activation steering can achieve the same internal alignment benefits as fine-tuning,** but does so through a single, targeted intervention rather than a lengthy training process. To demonstrate this, we designed an experiment to visualize *how* each method forces the model to align its internal representations with English.

We employed a "logit lens" analysis (nostalgebraist, 2020), a technique that allows us to **peek inside the model**. At a specific layer $L + 1$, we take the model's internal state the post-activation output $x_i^{(L+1)}$ [2] and project it back into the vocabulary space using the unembedding matrix. In simple terms, we ask the model: ***"Based on your current state, what English word does this most look like?"***

$$\text{logits}_i^{(L+1)} = \text{Unembed}(x_i^{(L+1)}) \in \mathbb{R}^{|V|}$$

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 3: 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.

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

By applying a softmax function to these logits, we can generate a probability distribution over the entire vocabulary, showing us the model's "best guess" for the next token.

$$y_i = \text{softmax}(\text{logits}_i^{(L+1)}) \in \mathbb{R}^{|V|}$$

We then use a language detection tool[4] on the most likely tokens to see if the model's internal state has successfully aligned with English. We compared three scenarios, with the full results shown in Figure 6 (Appendix):

1. **Base Model:** Exhibits weak cross-lingual alignment; target language representations remain distant from English.

2. **Fine-Tuned Model:** Learns to align target and English representations after extensive translation fine-tuning.

3. **Steered Model:** Matches this alignment instantly using a single steering vector.

Both fine-tuning and steering improve representational alignment with English: fine-tuning achieves this gradually over many steps, while steering provides an immediate, targeted correction. This efficiency raises a key question: *do multilingual models still need steering?* Logit lens analysis shows that **advanced multilingual models already exhibit strong alignment with English, without requiring intervention.** Thus, the need for steering reflects limitations in base model training. While steering is an effective fix, better multilingual pretraining may eliminate the issue entirely. Full details are in Appendix C.

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.

[4] https://github.com/Mimino666/langdetect

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 were also intentionally focused, isolating a single layer and language at a time. This controlled approach was necessary to establish a clear baseline, but a truly powerful system must learn to juggle multiple languages and layers simultaneously through more advanced, multi-objective training.

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.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083.

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.

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*.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530.

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

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. 2023. Steering llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*.

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.

Quan Hung Tran, Svetha Venkatesh, Hung Le, et al. 2025. Dynamic steering with episodic memory for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 13731–13749.

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

Tianlong Wang, Xianfeng Jiao, Yinghao Zhu, Zhongzhi Chen, Yifan He, Xu Chu, Junyi Gao, Yasha Wang, and Liantao Ma. 2025. Adaptive activation steering: A tuning-free llm truthfulness improvement method for diverse hallucinations categories. In *Proceedings of the ACM on Web Conference 2025*, pages 2562–2578.

Weixuan Wang, Minghao Wu, Barry Haddow, and Alexandra Birch. 2024. Bridging the language gaps in large language models with inference-time cross-lingual intervention. *arXiv preprint arXiv:2410.12462*.

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*.

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.

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

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.

$$\min_v -\mathbb{E}_{d\sim\mathcal{U},(q,r_T,r_O)\sim\mathcal{D}}\left[\log\sigma\left(d\beta\log\frac{\pi_{L+1}(r_T|A_L(q)+dv)}{\pi_{L+1}(r_T|A_L(q))}-d\beta\log\frac{\pi_{L+1}(r_O|A_L(q)+dv)}{\pi_{L+1}(r_O|A_L(q))}\right)\right]. \tag{7}$$

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 $\eta$, 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 logits from both activations
        $\ell \leftarrow \text{Logits}(M, \tilde{h})$
        $\ell^{\text{en}} \leftarrow \text{Logits}(M, h^{\text{en}})$
                                          ▷ 4. Direct Preference Optimization (DPO) loss
            $\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{t\sim\mathcal{V}}\big[\log\sigma\big(\ell_t^{\text{en}} - \ell_t\big)\big]$   (see App. A, eq.7)
                                          ▷ 5. Gradient-step update
        $v \leftarrow v - \eta\nabla_v\mathcal{L}_{\text{DPO}}$
    end for
end for
Return $v$
```

Algorithm 2 MSE Steering Vector Learning

```
Require: Pretrained LLM $M$, bilingual corpus $\mathcal{D} = \{(q_i, q_i^{\text{en}})\}$, layer $L$, learning rate $\eta$, 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 $L$
        $h \leftarrow \mathrm{HiddenState}(M, q, L)$
        $h^{\text{en}} \leftarrow \mathrm{HiddenState}(M, q^{\text{en}}, L)$
                                        ▷ 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 4 highlights the performance of various models across diverse tasks, underscoring the effectiveness of this approach.

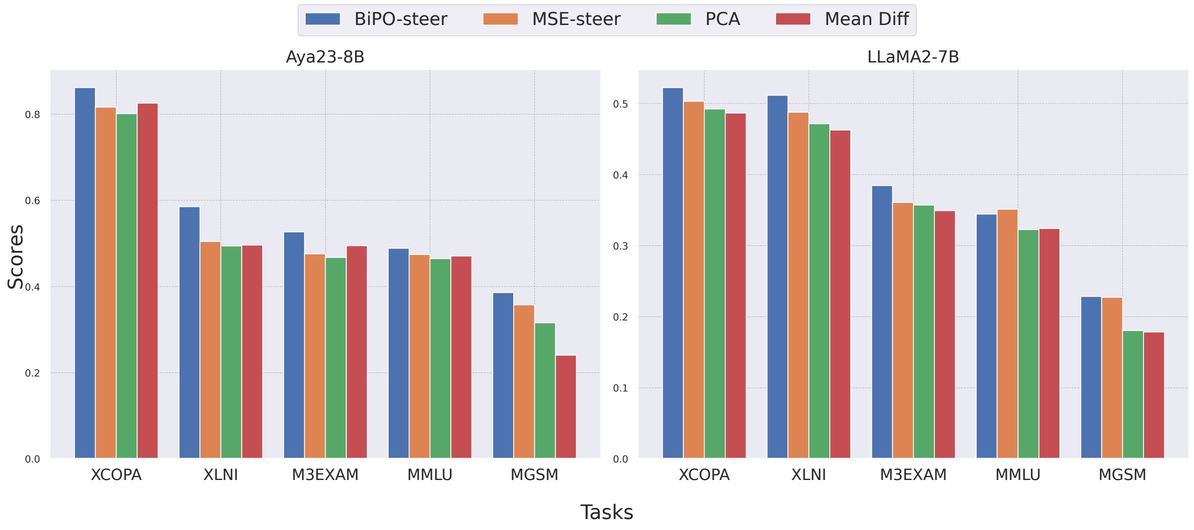


Figure 4: demonstrate that using learnable steering vectors surpasses PCA and the Mean Difference approaches across all tasks on two models: Aya32-8B and LLama2-7B.

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:

$$H_l(x) = h_l(x) + S_l(x) \tag{8}$$

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:

$$\mathcal{L}(\mathcal{M}^{(}Q), A) = -\sum_{t=1}^{T} \log P(a_t | a_{<t}, Q; \theta^{)}) \tag{9}$$

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:

$$H_l(x) = h_l(x) + \alpha v \tag{10}$$

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.

C.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 5). These models map multilingual representations onto an

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-Tr	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
SFT	35.4	35.0	31.8	34.4	26.0	28.1	31.7
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 5: Results of the MGSM Task Evaluated on the Llama2-13B Model Across Diverse Languages

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).

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 5 indicates that these larger models follow the same trend of performance improvements across different languages, mirroring the behavior observed in smaller models.

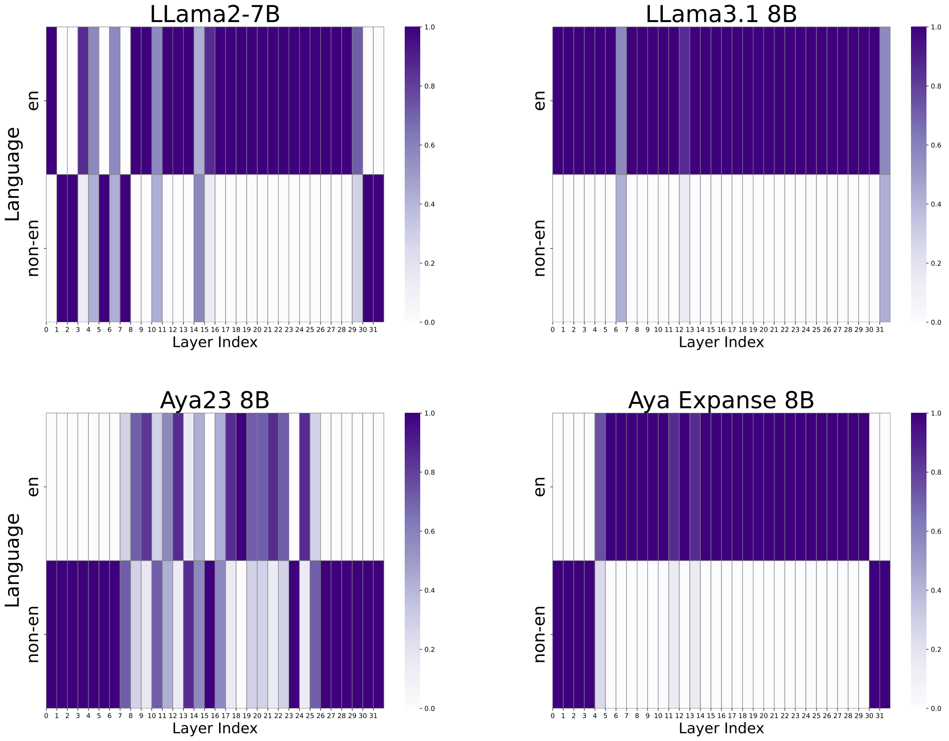


Figure 5: illustrates the processing of multilingual tokens in models of varying capabilities within the same family. LLama3.1 demonstrates a strong alignment of tokens into English-aligned representations, whereas LLama2 struggles with this. Similarly, Aya-Expanse exhibits robust token alignment, attributed to RLHF techniques, while Aya23 shows weaker alignment.

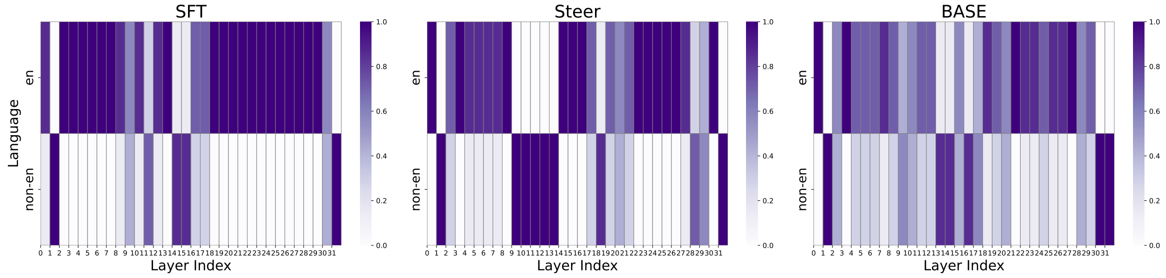


Figure 6: : Distribution of non-English token values across model layers at three different stages: pre-fine-tuning (base model), post-fine-tuning(SFT), and after applying steering at a specific layer. The results demonstrate that both fine-tuning and steering exhibit similar behavior, aligning non-token values more closely with English token distributions.

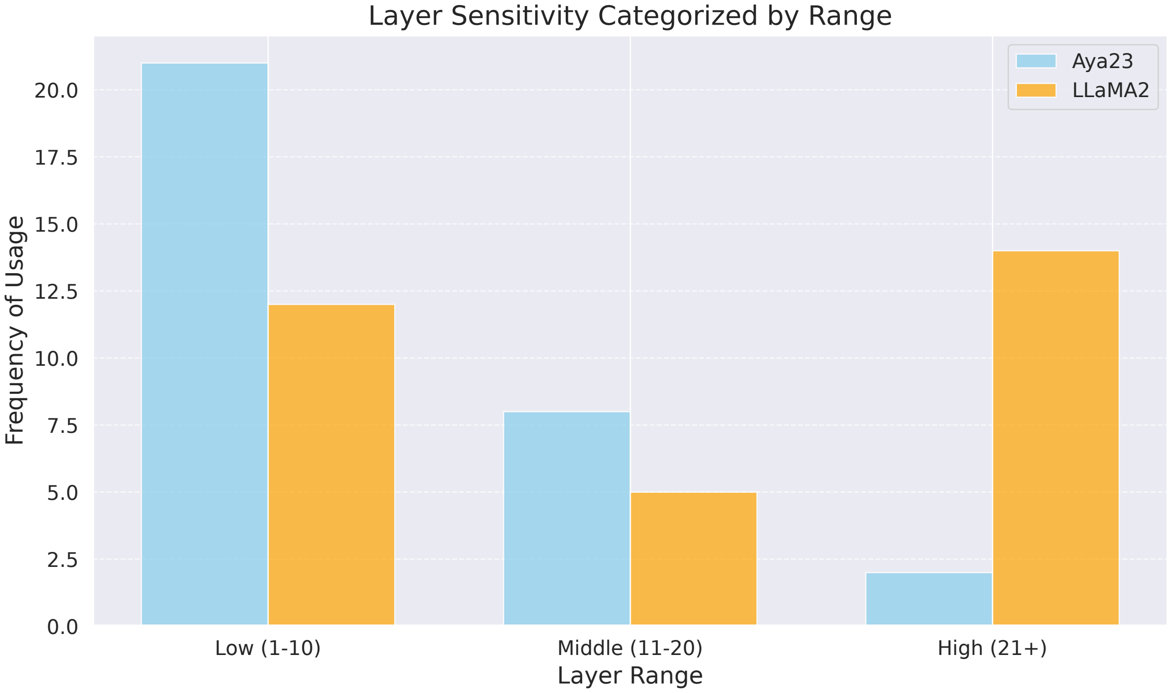
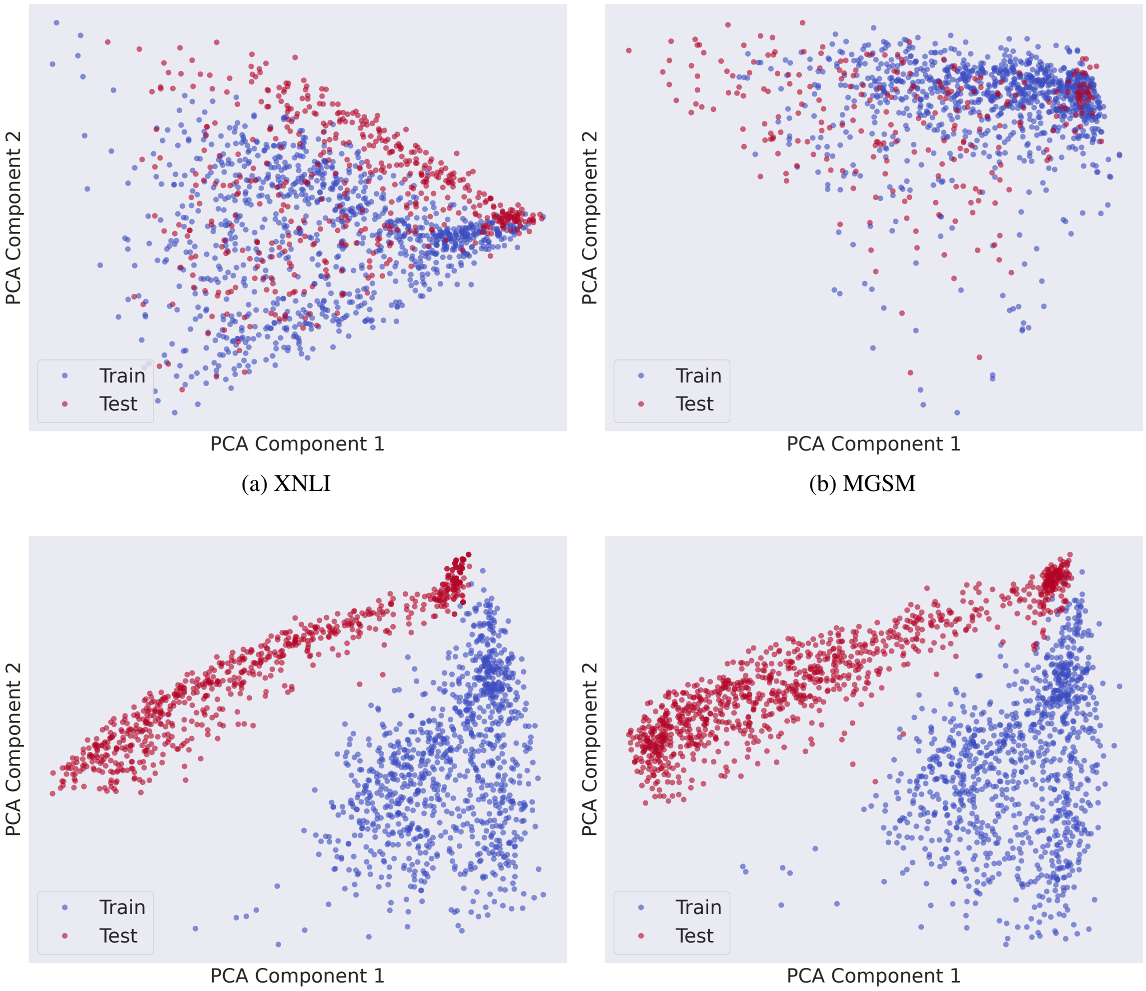


Figure 7: illustrates the layers most sensitive to probing across two models. Aya23 demonstrates high sensitivity in the initial layers but exhibits reduced performance in the middle and later layers. In contrast, LLama2 experiences a notable drop in performance in the middle layers, with improved results in the later layers. Additionally, the initial layers of LLama2 perform better for low- and medium-resource languages.



(a) XNLI

(b) MGSM

(c) M3EXAM

(d) MMLU

Figure 8: 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.

Methods	*MGSM*	*XCOPA*	*XNLI*	*M3EXAM*	*MMLU*
Aya32-8B					
Basic	32.6	81.6	49.9	46.9	45.3
Google-Tr	37.6	83.9	52.4	49.4	50.7
NLLB	32.3	73.2	49.9	26.5	34.0
5@Shot	36.1	84.5	59.8	42.5	30.9
XLT	26.9	12.1	52.7	38.8	27.0
SFT	34.4	82.0	49.8	47.4	46.0
DPO-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	19.6	47.6	46.9	30.6	31.3
Google-Tr	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
SFT	24.0	49.9	49.4	36.4	34.0
DPO-Steer	22.9	52.2	55.1	38.4	34.4
MSE-steer	22.8	50.3	48.7	35.1	36.0
Llama3-8B					
Basic	62.0	66.7	63.2	51.6	50.7
Google-Tr	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
SFT	64.7	72.2	63.9	53.8	51.6
DPO-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	27.3	66.2	46.4	37.3	39.6
Google-Tr	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
SFT	28.6	67.8	49.2	43.1	40.8
DPO-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$_{(+0.3)}$	68.0$_{(+1.8)}$	47.9$_{(+1.5)}$	37.5$_{(+0.2)}$	41.1$_{(+1.5)}$
Qwen1.5-7B					
Basic	49.2	70.1	66.0	58.3	42.1
Google-Tr	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
SFT	50.6	71.6	67.4	58.2	43.8
DPO-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.