Mitigating Language Confusion through Inference-time Intervention

Yunfan Xie[1], Lixin Zou[1*], Dan Luo[2], Min Tang[3],
Chenliang Li[1], Liming Dong[4], Xiangyang Luo[5*]

[1]Wuhan University, Wuhan, China, [2]Lehigh University, Bethlehem, USA
[3]Monash University, Melbourne, Australia, [4]National Defense University, Beijing, China
[5]State Key Lab of Mathematical Engineering and Advanced Computing, Zhengzhou, China
{yunfanxie, zoulixin, cllee}@whu.edu.cn, dal417@lehigh.edu
min.tang@monash.edu, dlm14@tsinghua.org.cn, xiangyangluo@126.com

Abstract

Although large language models (LLMs) trained on extensive multilingual corpora exhibit impressive language transfer, they often fail to respond in the user's desired language due to corpus imbalances, an embarrassingly simple problem known as the language confusion. However, existing solutions like in-context learning and supervised fine-tuning (SFT) have drawbacks: in-context learning consumes context window space, diminishing attention as text lengthens, while SFT requires extensive, labor-intensive data collection.

To overcome these limitations, we propose the language-sensitive intervention (LSI), a novel, lightweight, and label-free approach. Specifically, we analyze language confusion from a causal perspective, revealing that the training corpus's language distribution acts as a confounder, disadvantaging languages that are underrepresented in the dataset. Then, we identify a language-sensitive dimension in the LLM's residual stream, i.e., the language vector, which allows us to estimate the average causal effect of prompts on this dimension. During inference, we directly intervene on the language vector to generate responses in the desired language.To further advance research on this issue, we introduce a new benchmark that detects language confusion and assesses content quality. Experimental results demonstrate that our method effectively mitigates language confusion without additional complex mechanisms. Our code is available at https://github.com/SoseloX/LSI.

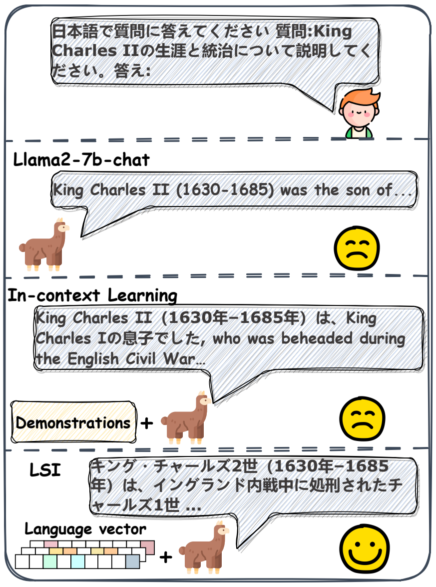

Figure 1: An illustration of language confusion. In-context learning cannot address the problem, where LLAMA 2-CHAT-7B initially responds in the user's desired language but tends to switch to English midway through its response, even when examples are provided as demonstrations.

1 Introduction

Large language models, such as GPT (Achiam et al., 2023), LLAMA 2-CHAT-7B (Touvron et al., 2023), Falcon (Almazrouei et al., 2023) and PaLM (Chowdhery et al., 2023) have shown impressive performance on various natural language tasks, e.g., reasoning, mathematics and code generation (Achiam et al., 2023; Wei et al., 2022; Liu et al., 2023a; Ouyang et al., 2022; Tang et al., 2025). They have matched or even surpassed the performance of supervised models that are trained with millions of labeled examples. Although these models support multilingualism, most are predominantly trained on English corpora that have undergone extensive cleaning, whereas the counterpart corpora in other languages have not been adequately processed. For example, while the C4 corpus (Raffel et al., 2020) applies extensive cleaning to English texts, it leaves significant amounts of gambling and adult-related content in other language corpus. Therefore, recent studies (Marchisio et al., 2024; Kew et al., 2023) have discovered an embarrassingly simple problem, named *language confusion* (Marchisio et al., 2024), where a LLM responds in an entirely incorrect language or switches

*Corresponding author.

Proceedings of the 31st International Conference on Computational Linguistics, pages 8418–8431
January 19–24, 2025.

to an undesired language mid-response. As shown in Figure 1, when users ask LLAMA 2-CHAT-7B a question in Japanese and explicitly request a response in Japanese, the model still fails to do so. This issue mainly arises due to the distribution of training data: since web corpora are predominantly in English, many LLMs are English-centric (Zhang et al., 2020, 2023b). Moreover, during inference, the model relies on generated tokens to guide subsequent predictions; thus, an incorrect language token can lead to a cascade of errors.

To address language confusion in multilingual large language models (LLMs), one approach is to use few-shot examples as demonstrations to guide the model's responses in the desired language (Marchisio et al., 2024). However, as the length of the generated text increases, the model's attention to these demonstrations diminishes, potentially causing it to switch languages midway through the generation process (see Figure 1). Additionally, incorporating demonstrations consumes part of the available context window, such as the $4k$ tokens limit in the LLAMA 2 series models, and introduces additional computational overhead.

An alternative is supervised fine-tuning, which can help the language model adapt to low-source languages (Zhang et al., 2023a; Dong et al., 2023; Luo et al., 2023b; Cobbe et al., 2021). However, producing high-quality, language-distribution-balanced fine-tuning datasets comparable to those created by large corporations is both expensive and labor-intensive (Achiam et al., 2023). Therefore, a lightweight method to address language confusion is needed for multilinguistic LLMs.

Towards this end, we propose a simple yet effective method named **l**anguage-**s**ensitive **i**ntervention (LSI). Specifically, we employ a causal framework to clarify language confusion. Within this framework, the distribution of languages in the training corpus serves as a confounding variable, simultaneously influencing both the latent language representations, referred to as the *language vector*, and the model's outputs. Using a probing network, we identify the *language vector* as the language-sensitive dimension within the residual stream of the transformer module. Next, we estimate the average treatment effect of language requirements in the prompt on the language vector. During inference, we directly intervene on the language vector to generate text in the desired language. Extensive experiments conducted on benchmark datasets demonstrate that our approach effectively mitigates language confusion with negligible impact on the model's generative performance. Our contributions are highlighted as follows:

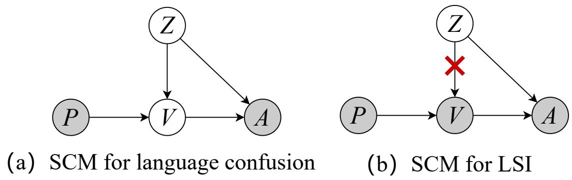

Figure 2: We utilize causal graphs to illustrate language confusion in multilingual inference. Specifically, let $\boldsymbol{P}$ represent the prompt, $\boldsymbol{V}$ the latent language representation (i.e., language vectors), $\boldsymbol{Z}$ the pre-trained or post-trained corpus, and $\boldsymbol{A}$ the output text. We identify $\boldsymbol{Z}$ as a confounder between $\boldsymbol{V}$ and $\boldsymbol{A}$ and propose LSI to sever the causal pathway from $\boldsymbol{Z}$ to $\boldsymbol{V}$. A gray node signifies that the variable is observable by identifying the language vectors within the residual stream.

- We conduct the first comprehensive study demonstrating the impact of language-sensitive dimensions in the residual streams of large language models, which leads to language confusion.

- We propose a novel method LSI with close-to-zero computational overhead to mitigate language confusion in large language models by intervening in language-sensitive dimensions.

- We introduce a benchmark to facilitate further research on language confusion. This benchmark not only focuses on whether the generated responses are in the target language but also on the quality of the generated content

- We conduct extensive experiments to demonstrate that the proposal effectively addresses language confusion with negligible impact on the model's generative performance.

2 Causal Analysis on Language Confusion

To better understand how the language preferences learned by the model influence the selection of the language in generated responses, we employed a Structural Causal Model (SCM) (Pearl et al., 2000) to illustrate the inference process of multilingual language models.

As shown in Figure 2(a), node $\boldsymbol{P}$ represents the prompt provided by the user. Node $\boldsymbol{V}$ represents the inherent, unknown linguistic representation, referred to as the "language vector", within LLMs, which determines the language used in the generated text. Node $\boldsymbol{Z}$ represents the language distribution of pre-trained or fine-tuning corpus, and node $\boldsymbol{A}$ is the generated text. The edge $\boldsymbol{Z} \rightarrow \boldsymbol{A}$ signifies the linguistic agnostic experience from the corpus will affect the generated text (Wu et al., 2024; Wang et al., 2021).

The backdoor path $\boldsymbol{V} \leftarrow \boldsymbol{Z} \rightarrow \boldsymbol{A}$ reveals that $\boldsymbol{Z}$ acts as a confounder, simultaneously affecting the language vector $\boldsymbol{V}$ and the generated text $\boldsymbol{A}$. The path $\boldsymbol{Z} \rightarrow \boldsymbol{V} \rightarrow \boldsymbol{A}$ illustrates how the language preferences learned during training impact the generated text $\boldsymbol{A}$ via language vector $\boldsymbol{V}$. When the model is predominantly trained on English data, the language vector tends to favor English, which in turn influences the language of the generated text. Ideally, the generated response should align with the user's intent as

$$P(\boldsymbol{A}|do(\boldsymbol{P})) = \sum_{\boldsymbol{V}} P(\boldsymbol{A}|do(\boldsymbol{V}))P(\boldsymbol{V}|do(\boldsymbol{P})), \quad (1)$$

where $P(\boldsymbol{V}|do(\boldsymbol{P}))$, $P(\boldsymbol{A}|do(\boldsymbol{V}))$ is the causal effect $\boldsymbol{P} \rightarrow \boldsymbol{V}$ and $\boldsymbol{V} \rightarrow \boldsymbol{A}$, respectively.

3 Methodology

From the perspective of causal inference, addressing language confusion is essentially to intervene on language desire in prompts and answer questions such as "*What will the response be if the prompt w.r.t. desired language is Chinese instead of English*"? However, when applied to our problem, the linguistic representation is unobservable.

In this section, we first present an approach to seek the linguistic representation in the residual stream, which makes the linguistic representation observable. Then, we estimate the average treat effect of *language desire* in user's prompt $\boldsymbol{P}$ on language vector $\boldsymbol{V}$. At last, we discuss the causal intervention to instantiate $P(\boldsymbol{A}|do(\boldsymbol{V}))$, which cuts off the backdoor $\boldsymbol{Z} \rightarrow \boldsymbol{V}$.

3.1 Seeking Language Vector

The influence of training data language distribution on the internal representations within transformer-based models is still not fully understood, resulting in the language vector being an unobservable variable. Fortunately, recent work has shown that preferences and knowledge are integrated into the residual stream, which consists of outputs from both the feed-forward and attention blocks, of language models (Geva et al., 2022; Liu et al., 2023b). Therefore, **we suppose that the language vector is the *language-sensitive dimensions* within the residual stream.**

Particularly, similar to prior work (Geva et al., 2022; Liu et al., 2023b), in this work, we only consider the outputs from the feed-forward layer for simplicity. For an T-layers transformer with a input sequence $\mathbf{X}$, we stack the residual stream across the layers as follows:

$$\boldsymbol{H} = [\boldsymbol{h}_1 \oplus \boldsymbol{h}_2 \oplus \ldots \oplus \boldsymbol{h}_T], \quad (2)$$

where $\boldsymbol{h}_k \in \mathbb{R}^D$ is the residual stream of last token from layer k, D denotes the size of the hidden state in the language model and $\boldsymbol{H} \in \mathbb{R}^{TD}$, $\oplus$ denotes the concatenation operation. In the following section, we first conduct an empirical experiment to justify the language-sensitive dimensions, then we present a method that utilizes a probing network to identify the language-sensitive dimensions within the residual stream.

Justification for Language Vector To justify our hypothesis about the language vector, we create 50 English-Chinese prompt pairs using the Google Translate API. Since the input pairs differ only in language desire of prompts, we calculate the differences in the hidden states of the final token output at each transformer block layer. We select the final token output because it captures the semantic representation of the entire sequence (Li et al., 2023). By feeding each prompt pair into LLAMA 2-CHAT-7B, we aim to exploit the language-sensitive dimensions. Figure 4 shows the heatmap of the differences in the 30th layer of hidden states. We take the absolute values of the hidden state matrix and reshape it into 64×64 dimensions for convenient visualization as a heatmap. We find that only a small number of dimensions are extremely sensitive to language differences, with variation in these dimensions being more than 100 times larger than in other dimensions. This observation suggests the existence of dimensions in the residual stream that are sensitive to desired language instruction in the prompt.

Identifying Language Vector with Probe Network Due to differences in language and culture, obtaining high-quality translation pairs that differ only in language while maintaining the same semantics would be challenging. To address this problem, we identify the language vector by selecting the dimension that is primarily used to classify the language type of the monolingual text within a probe network (Belinkov, 2022).

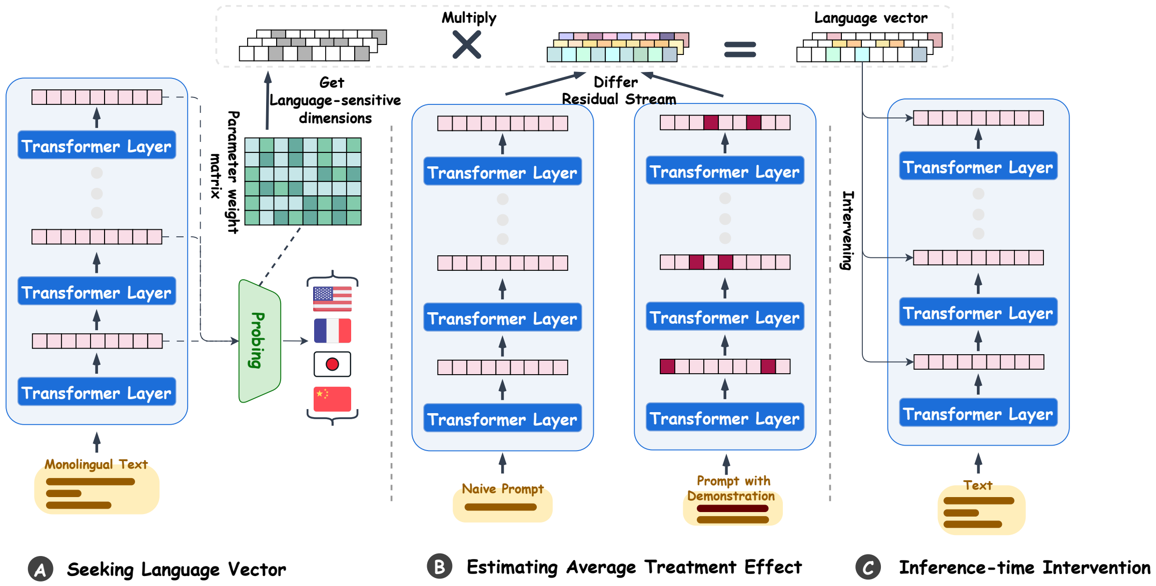

Figure 3: The overview of LSI includes: (a) exploiting the language-sensitive dimensions through a probing network, (b) estimating average treatment effect by differing the residual stream with and without the demonstration prompt, and (c) during inference, reintroducing the average treatment effect into the residual stream to intervene the model's output in the desired language.

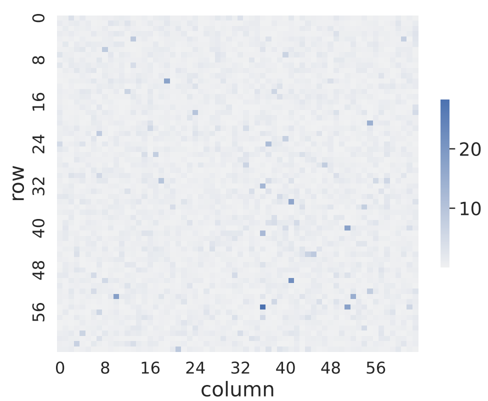

Figure 4: Heatmap of the difference in the sub-residual stream between English and Chinese text inputs. We take the 30th layer of the hidden state and reshape it to 64×64 dimension to facilitate visualization. The intensity represents the magnitude of the difference. Dimensions with lighter colors are less sensitive to language variations.

Particularly, the probe network is a single-layer classifier that learns the mapping from residual steam vectors to the language type of generated text. Formally, it is defined as:

$$\hat{\boldsymbol{y}} = \mathrm{softmax}(\boldsymbol{W}^\top \boldsymbol{H}), \tag{3}$$

where $\boldsymbol{W} \in \mathbb{R}^{TD \times L}$ is the weight matrix of the probing network, L is the number of candidate languages. Thus, $\hat{\boldsymbol{y}} \in \mathbb{R}^L$ represents the probability of a given text belonging to a particular language. We use cross entropy loss to train the probing network:

$$\mathcal{L} = -\boldsymbol{y} log(\hat{\boldsymbol{y}}), \tag{4}$$

where $\boldsymbol{y}$ is the ground truth for the language type. We input a monolingual text into the large language model and extract the residual streams corresponding to the last token as $\boldsymbol{H}$. Since a higher weight value in $|\boldsymbol{W}|$ indicate a greater contribute for the corresponding dimension in $\boldsymbol{H}$, we refer to $\boldsymbol{H}_j$ as a language-sensitive dimension if the $|\boldsymbol{W}_j|$ ranks among the top in the j-th column of $|\boldsymbol{W}|$. For each language l, we can obtain language-sensitive dimension masking matrix via the following formulation:

$$\boldsymbol{M}_l = \begin{cases} 1 & \text{if } |\boldsymbol{W}_{jl}| > \text{threshold} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The value of threshold is set to the value of the top α percent elements in $\boldsymbol{W}$, where α is a hyperparameter. To this end, we obtain the language vector for language l as:

$$\boldsymbol{V}_l = \boldsymbol{M}_l \odot \boldsymbol{H} \tag{6}$$

where $\odot$ is the Hadamard Product and $\boldsymbol{v} \in \mathbb{R}^{LD}$ is only activated at language-sensitive dimensions.

3.2 Estimating Average Treatment Effect

Before inference-time intervene on the language vector, we first define the average treatment effect (ATE) of user's prompt p_l on language vector

Setting	Task	Data Source	Languages	Nums	PL	AL
Monolingual	Question Answering	Okapi	fr, it, jp, zh, ru	100	17	105
Crosslingual	Question Answering	Okapi	fr, it, jp, zh, ru	100	10	81

Table 1: The statistics of Quality-aware Language Confusion Benchmark. Nums represents the number of samples for each language. PL denotes the average length of text in the prompt. AL denotes the average length of text in the answer.

$\boldsymbol{V}$ as:

$$\begin{aligned} \mathrm{ATE}(\boldsymbol{V}, p_l) &= \mathbb{E}\left[\boldsymbol{V}(p_l, z)\right] - \mathbb{E}[\boldsymbol{V}(p^*, z)], \quad (7a) \\ &= \mathbb{E}[\boldsymbol{V}_l] - \mathbb{E}[\boldsymbol{V}_l^*] \quad (7b) \end{aligned}$$

where $\boldsymbol{V}_l$ denotes the language vector of prompts in desired language l and $\boldsymbol{V}_l^*$ denotes its counterpart using language-sensitive mask $\boldsymbol{M}_l$ against the dominate language, *e.g.,* English.

Therefore, to estimate the ATE, we construct a prompt pair (p_i^l, p_i^*), where p_i^l is a prompt emphasizes the output language l through a demonstration, while p_i^* does not. We use demonstration because it can effectively guide the response following the desired language l (Marchisio et al., 2024). Examples of the prompt pairs with demonstration can be found in Appendix A. Afterwards, the prompt pair is fed into large language models, yielding the language vectors $\boldsymbol{V}_{il}$ and $\boldsymbol{V}_{il}^*$. Theoretically, the sample space of p_l is infinite, which makes the calculation of expectation in Equation 7 intractable. Therefore, we approximate the expectation with the empirical average on N prompt pairs' differences $\boldsymbol{\delta}_l$:

$$\boldsymbol{\delta}_l = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{V}_{il} - \boldsymbol{V}_{il}^*). \quad (8)$$

3.3 Inference-time Intervention

Since the average treatment effect essentially estimates the language desire in user's prompt $\boldsymbol{P}$ on language vector $\boldsymbol{V}$ against the dominate language, we direct intervene on the language vector during inference. Specifically, we choose to intervene the intermediate representation by adding back the average treatment effect measured in Section 3.2. Given the desire language l, we intervene with the corresponding language vector as shown in the following formula:

$$\begin{aligned} \widetilde{\boldsymbol{h}_t^l} &= \boldsymbol{h}_t^l + \beta\boldsymbol{\delta}_{l,tD:(t+1)D} \\ \boldsymbol{h}_t^{l+1} &= \text{Transformer-block}(\boldsymbol{h}_t^l), \end{aligned} \quad (9)$$

where $\boldsymbol{h}_t^l$ indicates the residual stream in layer t for desired language l, $\boldsymbol{\delta}_{l,tD:(t+1)D}$ is the average treatment effect for layer t, β is the intervening strength which is a hyperparameter, Transformer-block$(\cdot)$ refers to a single transformer layer operation applied to the inputs. $\widetilde{\boldsymbol{h}_t^l}$ will then continue to be fed into the $t+1$ layer of the transformer.

4 Experiments

This section validates LSI through comprehensive evaluations and analyses. We benchmark the language confusion test against existing methods, explore key parameter selections, and assess the requirements for accurately estimating treatment effects. These studies provide guidance on effectively applying LSI in various contexts.

4.1 Quality-aware Language Confusion Benchmark

This section presents our proposed Quality-aware Language Confusion Benchmark, designed to advance research on language confusion. The benchmark focuses on question answering tasks, utilizing data sourced from Okapi (Lai et al., 2023). We define two task settings: monolingual and cross-lingual. In the **monolingual setting**, both the question and the answer are in the same language. In contrast, the **cross-lingual setting** features questions in English with answers provided in another language. Our benchmark encompasses five languages: French (fr), Italian (it), Japanese (jp), Chinese (zh), and Russian (ru). To ensure the quality of the dataset, we apply two filtering rules: 1) exclude prompts shorter than five characters, and 2) remove mathematical problems and code generation prompts based on the nature of the answers. After this cleanup process, we extract 100 samples per language for each task setting. The statistics of benchmark can be seen from Table 1.

Evaluation Metric To assess the matching between the generated text and the user-specified language, we measure language accuracy. Ideally, the binary metric would be evaluated through human assessment or advanced LLMs like GPT-4. However, these approaches are cost-prohibitive, and

Monolingual Setting												
	fr		it		jp		zh		ru		avg	
	ACC	MAUVE	ACC	MAUVE	ACC	MAUVE	ACC	MAUVE	ACC	MAUVE	ACC	MAUVE
LLAMA 2-CHAT-7B	0.57	0.215	0.54	0.173	0.32	0.108	0.37	0.320	0.48	0.168	0.46	0.197
+ ICL	0.81	**0.831***	0.85	**0.778***	0.83	0.729	0.76	0.797	0.80	0.639	0.81	0.755
+ SFT	0.91	0.677	0.88	0.648	0.94	0.571	0.92	0.590	0.93	0.593	0.92	0.616
+ LSI	**0.99***	0.783	0.98	0.764	**0.98***	**0.773***	**1.00***	**0.812***	**0.97***	**0.793***	**0.98***	**0.785***
LLAMA 3-INSTRUCT-8B	0.77	0.598	0.63	0.157	0.69	0.227	0.61	0.469	0.58	0.394	0.66	0.369
+ ICL	0.84	0.849	0.85	0.796	0.76	0.712	0.75	0.749	0.89	0.737	0.82	0.765
+ SFT	0.89	0.637	0.93	0.609	0.92	0.611	0.95	0.601	0.90	0.631	0.92	0.618
+ LSI	**1.00***	**0.863***	**0.99***	0.802	**0.97***	**0.778***	**1.00***	**0.831***	**1.00***	**0.805***	**0.99***	**0.816***
Crosslingual Setting												
LLAMA 2-CHAT-7B	0.13	0.046	0.29	0.084	0.19	0.046	0.24	0.113	0.17	0.051	0.20	0.068
+ ICL	0.82	0.686	0.88	0.675	0.73	0.612	0.75	0.658	0.67	0.629	0.76	0.652
+ SFT	0.71	0.549	0.72	0.523	0.69	0.496	0.72	0.516	0.63	0.532	0.69	0.523
+ LSI	**0.98***	**0.753***	**0.99***	**0.767***	**0.98***	**0.737***	**0.98***	**0.832***	**1.00***	**0.762***	**0.99***	**0.770***
LLAMA 3-INSTRUCT-8B	0.23	0.166	0.49	0.132	0.04	0.075	0.25	0.098	0.17	0.038	0.24	0.102
+ ICL	0.83	**0.756***	0.89	0.633	0.71	0.623	0.75	0.736	0.71	0.619	0.78	0.673
+ SFT	0.74	0.599	0.76	0.576	0.71	0.476	0.66	0.530	0.62	0.577	0.79	0.552
+ LSI	**0.99***	0.722	**1.00***	**0.778***	**0.98***	**0.759***	**0.96***	**0.851***	**0.99***	**0.801***	**0.98***	**0.782***

Table 2: Performance comparisons on Quality-aware Language Confusion Benchmark. The best performances are highlighted in bold. "$*$" indicates significant improvements over the best baseline results with p-value < 0.01.

LLMs may introduce inherent biases that compromise accuracy. As an alternative, we employ the open-source tool `langdetect` (Nakatani, 2010) for this evaluation.

Additionally, we evaluate the quality of the generated text using MAUVE (Pillutla et al., 2021), which compares generated text to human-written text to assess quality. To compute the MAUVE score, we first convert both the generated and reference texts into embeddings using a pre-trained model (`BERT-base-multilingual-cased` (Devlin et al., 2018)) in our experiments. The MAUVE score is then calculated based on the Kullback–Leibler divergences between the two text distributions within the embedding space.

4.2 Baselines

We compare our proposal with two types of baseline methods: in-context learning(ICL) and supervised fine-tuning(SFT). For in-context learning, we provide one example as instruction. For supervised fine-tuning, constrained by computational resources, we employ the LoRA fine-tuning method (Hu et al., 2022), a widely adopted parameter-efficient fine-tuning method (Li et al., 2025). We randomly select 100 samples for each language from Aya (Singh et al., 2024) to form our training data. It is important to note that the test data is not included in the training dataset. The maximum epoch is 10. The batch size is set to 128, the learning rate to 3×10^{-4}, the LoRA rank to 8, and the LoRA alpha to 16.

4.3 Experimental Setups

The experiments are conducted on LLAMA 2-CHAT-7B and LLAMA 3-INSTRUCT-8B, two of the most popular open-source multilingual large language models (Touvron et al., 2023; Dubey et al., 2024; Zhang et al., 2024a). To obtain the language-dominant dimension, we collect 500 text samples for each language from the WikiLingual dataset (Ladhak et al., 2020), chosen for its predominantly monolingual samples. We train the probing network using the Adam optimizer (Kingma and Ba, 2014), with a batch size of 1024, a maximum of 30 epochs, and a learning rate of 1×10^{-4}. For each task we additionally collect 100 samples for every language and conduct a grid search to determine the optimal α and β parameters for Adam. The range of values for α is $[0.02, 0.04, 0.06, 0.08, 0.10]$, and for β, it is $[0.2, 0.4, 0.6, 0.8, 1.0]$. We train 10 probing networks using different random seeds and calculate $|\boldsymbol{W}|$ by averaging the absolute values of the parameter weight matrices from these networks. We set N to 50 across all experiments. Throughout the entire experiment, we set the model's generation parameters with temperature to 0.5, top-k to 50, and repetition penalty to 1.0.

4.4 Main Experimental Results

Table 2 reports the experimental results on the monolingual and crosslingual setting. From the table, we have following observations: **1) LSI is effective in solving the language confusion as it significantly outperforms ICL and SFT in language accuracy.** In the monolingual setting, our

	fr		it		jp		zh		ru		avg	
	ACC	MAUVE	ACC	MAUVE	ACC	MAUVE	ACC	MAUVE	ACC	MAUVE	ACC	MAUVE
LLAMA 2-CHAT-7B	0.57	0.215	0.54	0.173	0.32	0.108	0.37	0.320	0.48	0.168	0.46	0.197
Random dimension	0.67	0.321	0.63	0.365	0.49	0.379	0.52	0.460	0.51	0.257	0.56	0.356
Bottom dimension	0.46	0.123	0.41	0.216	0.35	0.154	0.27	0.278	0.31	0.102	0.36	0.174
Top dimension (LSI)	**1.00***	**0.863***	**0.99***	**0.802***	**0.97***	**0.778***	**1.00***	**0.831***	**1.00***	**0.805***	**0.99***	**0.816***

Table 3: Three three different strategies for inference-time intervention. The results are reported on monolingual setting. The best performances are highlighted in bold. "$*$" indicates significant improvements over the best baseline results with p-value < 0.01.

approach improves language accuracy by 21.0% and 20.7% on LLAMA 2-CHAT-7B and LLAMA 3-INSTRUCT-8B, respectively, compared to ICL. In the crosslingual setting, our approach improves language accuracy by 25.6% and 24% compared to SFT and ICL, respectively. **2) LSI can maintains the quality of generated text, as indicated by superior MAUVE scores.** Our method surpasses ICL and SFT on LLAMA 2-CHAT-7B by 4.0% and 27.4% on MAUVE scores in monolingual setting. Similar improvements are observed in crosslingual setting. **3) Though SFT is considered an effective method for adapting to new languages, it might hurt generation quality.** In the monolingual LLAMA 2-CHAT-7B setting, the language accuracy of SFT is 13.5% higher than that of ICL. However, its MAUVE score is 18.4% lower compared to ICL. Therefore, it requires a substantial high-quality data for high quality adaption.

4.5 Analysis Experiments

Intervention Strategy's Effectiveness This subsection examines the effectiveness of the proposed intervention strategies on language-sensitive dimensions. Specifically, we applied three different strategies for inference-time intervention: the first, the strategy used in LSI, targets the top 4% of dimensions in the residual stream based on their corresponding weights in $\boldsymbol{W}$, and is denoted as the "Top dimension"; the second strategy involves randomly selecting 4% of the dimensions, termed the "Random dimension"; the final strategy chooses the lowest 4% of dimensions, labeled the "Bottom dimension". Table 3 reports different variants' performance on monolingual task.

The results in Table 3 demonstrate that **indiscriminate interventions, without focusing on language-sensitive dimensions, are ineffective in mitigating language confusion and can negatively impact model performance**. Specifically, "Top dimension" enhances language accuracy and MAUVE scores by 76% and 129%, respectively, compared to the Random dimension approach. Moreover, we find that manipulating the bottom dimension results in the generation of incoherent text, leading to lower language accuracy and a reduced MAUVE score.

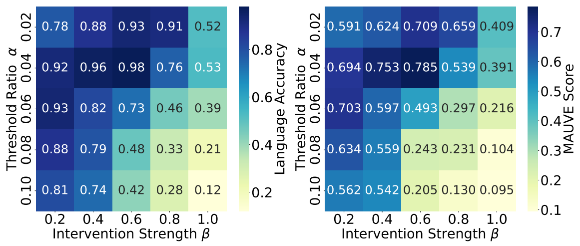

Figure 5: Results with varying intervention strength and threshold ratio in the monolingual setting on the LLAMA 2-CHAT-7B.

Influence of Intervention Strength and Threshold We examine the influence of hyperparameters on controlling intervention strength and threshold of language-sensitive dimensions to guide the use of the proposed LSI. Specifically, Figure 5 showcases how varying the threshold of language-sensitive dimensions and the intervention strength impacts model performance.

Our findings reveal that **setting proper interventions benefits the model's ability to generate text in the target language**. In fact, excessively strong interventions can undermine generative capacity, resulting in incoherent or even garbled text. Furthermore, excessively high thresholds for selecting language-dominant dimensions can encroach on language-agnostic dimensions, potentially harming the model's capability. Intervening the bottom dimension resulted in an 11.7% decrease in MAUVE.

Influence of Prompt Pair Number This section investigates how the average treatment effect is influenced by varying the number of prompt pairs used for its estimation. Figure 6 illustrates the performance of the LLAMA 3-INSTRUCT-8B in a monolingual setting under varying numbers of prompt pairs.

We observe that **appropriately setting the num-**

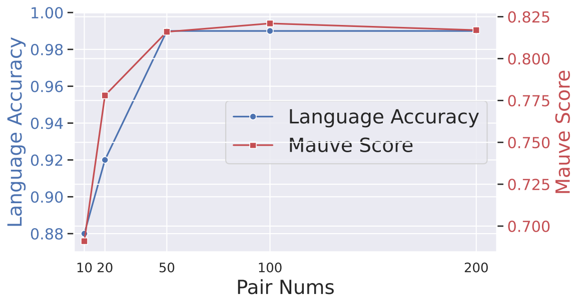

Figure 6: Results with varying prompt pair nums in the monolingual setting on the LLAMA 3-INSTRUCT-8B.

ber of prompt pairs allows for the accurate measurement of ATE. With a smaller number of prompt pairs, accurately estimating the average treatment effect becomes challenging, leading to suboptimal performance. However, once the number of prompt pairs exceeds 50, the ATE can be measured with reasonable accuracy. Increasing the number of prompt pairs beyond this threshold does not significantly improve the results.

5 Related Work

This section summarizes related topics, including language confusion, residual stream engineering, and causal inference in LLMs.

5.1 Language Confusion

As large language models advance and global integration deepens, interest in multilingual models or adapting to specific non-English languages has surged (Zou et al., 2021; Zhao et al., 2024; Wendler et al., 2024). Despite numerous efforts to enhance these models' multilingual capabilities through training (Xue et al., 2021; Conneau et al., 2020; Scao et al., 2022; Muennighoff et al., 2023), prompt engineering (Vilar et al., 2023; Huang et al., 2023; Qin et al., 2023), and attempts to explain the underlying mechanisms (Tang et al., 2024; Zhang et al., 2024c), language confusion remains a significant challenge. Researchers have explored the causes of this issue. Li and Murray (2023) identified that language-invariant representations learned during fine-tuning interfere with language selection during generation. To mitigate language confusion, some have propose strengthening models' multilingual capabilities through methods such as multilingual post-training or providing few-shot examples(Marchisio et al., 2024). However, collecting high-quality, linguistically balanced fine-tuning data is extremely challenging; while providing demonstration with in-context learning incurs additional computational costs with limited effectiveness. This work investigates language confusion from the causal inference perspective, and we propose lightweight method via inference-time intervention.

5.2 Residual Stream Engineering

Residual stream engineering is also known as representation engineering, which enhances model performance by directly modifying the residual stream in language models (Subramani et al., 2022; Hernandez et al., 2023). This technique enables adjusting the output style of language models (Liu et al., 2023b; Turner et al., 2023; Dathathri et al., 2020), mitigating hallucinations (Li et al., 2023), and detoxifying the generated content (Liu et al., 2023b). A canonical work is PPLM (Dathathri et al., 2020). PPLM utilizes simple attribute classifiers to guide model outputs by adjusting residual stream through gradients from the attribute model during inference, thus steering the generation towards desired attributes. Another method, ICV (Liu et al., 2023b), constructs an in-context vector using numerous demonstrations and the model's forward pass, which subsequently adjusts the model's residual stream during inference. In our work, we innovatively apply residual stream engineering to tackle the language confusion problem. It enables us to make language-sensitive dimensions observable.

5.3 Causal Inference in Language Model

Causal inference, as introduced by Pearl (2009), has been extensively applied across various domains, including web search(Luo et al., 2023a; Zou et al., 2022; Ai et al., 2018), recommendation systems (Zhang et al., 2021; Chen et al., 2021), and the mitigation of biases in large language models (Wang et al., 2023; Zeng et al., 2020; Tian et al., 2022; Zhang et al., 2024b). Recent studies leverages front-door adjustment via instrumental variables to mitigate bias in large language models. For instance, DeCoT (Wu et al., 2024) considers external knowledge as an instrumental variable and estimates the average causal effect on LLMs using this approach. Similarly, Causal Walk (Zhang et al., 2024b) uses the reasoning path between the input and output as a mediator to conduct front-door adjustment. Our work diverges significantly as we use the probing network to make language-sensitive dimensions observable, thus we can directly perform causal intervention to mitigate language confusion.

6 Conclusion

This work presents a causal perspective on language confusion in large language models. Specifically, we model the distribution of the training corpus as a confounder that influences the language generated by these models and introduce the LSI framework to address this issue. Within the LSI framework, we analyze the vector in the residual stream of large language models that controls the generated language, referred to as the language vector. By estimating the average treatment effect of the user's prompt on the language vector, we mitigate language confusion through inference-time interventions. Furthermore, we introduce a Quality-aware Language Confusion Benchmark that assesses not only whether the model's response is in the desired language but also the quality of the response. Experimental results demonstrate that our method effectively alleviates language confusion.

7 Limitations

This paper analyzes the issue of language confusion from the perspective of causal inference and assesses the impact of training data on the language-dominant dimension. However, our approach to identifying the language-dominant dimension is based on empirical experiments, lacking a thorough theoretical analysis of this dimension. Additionally, due to the limited availability of training data, we were unable to examine how the distribution of training data influences the model's ability to generate responses in different languages. Furthermore, because of constraints on computational resources, we included the full fine-tuning method as part of our baseline approach.

Acknowledgement

We express our sincere gratitude for the financial support provided by the National Natural Science Foundation of China (NO. 62302345 and NO. U23A20305), the Natural Science Foundation of Hubei Province (NO. 2023AFB192 and NO.2023BAB160), the CCF-ALIMAMA TECH Kangaroo Fund (NO. CCF-ALIMAMA OF 2024009), the Xiaomi Young Scholar Program, and the Natural Science Foundation of Wuhan (NO. 2024050702030136).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Qingyao Ai, Keping Bi, Cheng Luo, Jiafeng Guo, and W Bruce Croft. 2018. Unbiased learning to rank with unbiased propensity estimation. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pages 385–394.

Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. 2023. The falcon series of open language models. *CoRR*, abs/2311.16867.

Yonatan Belinkov. 2022. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219.

Jiawei Chen, Hande Dong, Yang Qiu, Xiangnan He, Xin Xin, Liang Chen, Guli Lin, and Keping Yang. 2021. Autodebias: Learning to debias for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–30.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 8440–8451. Association for Computational Linguistics.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and play language models: A simple approach to controlled text generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Guanting Dong, Hongyi Yuan, Keming Lu, Chengpeng Li, Mingfeng Xue, Dayiheng Liu, Wei Wang, Zheng Yuan, Chang Zhou, and Jingren Zhou. 2023. How abilities in large language models are affected by supervised fine-tuning data composition. *arXiv preprint arXiv:2310.05492*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Mor Geva, Avi Caciularu, Kevin Ro Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 30–45. Association for Computational Linguistics.

Evan Hernandez, Belinda Z Li, and Jacob Andreas. 2023. Inspecting and editing knowledge representations in language models. *arXiv preprint arXiv:2304.00740*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Haoyang Huang, Tianyi Tang, Dongdong Zhang, 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. In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 12365–12394. Association for Computational Linguistics.

Tannon Kew, Florian Schottmann, and Rico Sennrich. 2023. Turning english-centric llms into polyglots: How much multilinguality is needed? *CoRR*, abs/2312.12683.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Faisal Ladhak, Esin Durmus, Claire Cardie, and Kathleen Mckeown. 2020. Wikilingua: A new benchmark dataset for cross-lingual abstractive summarization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4034–4048.

Viet Dac Lai, Chien Van Nguyen, Nghia Trung Ngo, Thuat Nguyen, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. Okapi: Instruction-tuned large language models in multiple languages with reinforcement learning from human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023 - System Demonstrations, Singapore, December 6-10, 2023*, pages 318–327. Association for Computational Linguistics.

Kenneth Li, Oam Patel, Fernanda B. Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Tianjian Li and Kenton Murray. 2023. Why does zero-shot cross-lingual generation fail? an explanation and a solution. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 12461–12476. Association for Computational Linguistics.

Weicheng Li, Lixin Zou, Min Tang, Qing Yu, Wanli Li, and Chenliang Li. 2025. Meta-lora: Memory-effcient sample reweighting forfine-tuning large language models. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING 2025)*.

Sheng Liu, Lei Xing, and James Zou. 2023a. In-context vectors: Making in context learning more effective and controllable through latent space steering. *arXiv preprint arXiv:2311.06668*.

Sheng Liu, Lei Xing, and James Zou. 2023b. In-context vectors: Making in context learning more effective and controllable through latent space steering. *CoRR*, abs/2311.06668.

Dan Luo, Lixin Zou, Qingyao Ai, Zhiyu Chen, Dawei Yin, and Brian D Davison. 2023a. Model-based unbiased learning to rank. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 895–903.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023b. Wizardcoder: Empowering code large language models with evol-instruct. *arXiv preprint arXiv:2306.08568*.

Kelly Marchisio, Wei-Yin Ko, Alexandre Bérard, Théo Dehaze, and Sebastian Ruder. 2024. Understanding and mitigating language confusion in llms. *arXiv preprint arXiv:2406.20052*.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M. Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15991–16111. Association for Computational Linguistics.

Shuyo Nakatani. 2010. Language detection library for java.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Judea Pearl. 2009. Causal inference in statistics: An overview.

Judea Pearl et al. 2000. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19:2.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaïd Harchaoui. 2021. MAUVE: measuring the gap between neural text and human text using divergence frontiers. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 4816–4828.

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, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2695–2709. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilic, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, and et al. 2022. BLOOM: A 176b-parameter open-access multilingual language model. *CoRR*, abs/2211.05100.

Shivalika Singh, Freddie Vargus, Daniel D'souza, Börje Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura O'Mahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Souza Moura, Dominik Krzeminski, Hakimeh Fadaei, Irem Ergün, Ifeoma Okoh, Aisha Alaagib, Oshan Mudannayake, Zaid Alyafeai, Minh Vu Chien, Sebastian Ruder, Surya Guthikonda, Emad A. Alghamdi, Sebastian Gehrmann, Niklas Muennighoff, Max Bartolo, Julia Kreutzer, Ahmet Üstün, Marzieh Fadaee, and Sara Hooker. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 11521–11567. Association for Computational Linguistics.

Nishant Subramani, Nivedita Suresh, and Matthew E. Peters. 2022. Extracting latent steering vectors from pretrained language models. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 566–581. Association for Computational Linguistics.

Min Tang, Lixin Zou, Shiuan-ni Liang, She Jin, Weiqing Wang, and Shujie Cui. 2025. Chifraud: A long-term web text benchmark for chinese fraud detection. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING 2025)*.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. *CoRR*, abs/2402.16438.

Bing Tian, Yixin Cao, Yong Zhang, and Chunxiao Xing. 2022. Debiasing NLU models via causal intervention and counterfactual reasoning. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11376–11384. AAAI Press.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Alexander Matt Turner, Lisa Thiergart, David Udell, Gavin Leech, Ulisse Mini, and Monte MacDiarmid. 2023. Activation addition: Steering language models without optimization. *CoRR*, abs/2308.10248.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George F. Foster. 2023. Prompting palm for translation: Assessing strategies and performance. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 15406–15427. Association for Computational Linguistics.

Fei Wang, Wenjie Mo, Yiwei Wang, Wenxuan Zhou, and Muhao Chen. 2023. A causal view of entity bias in (large) language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15173–15184, Singapore. Association for Computational Linguistics.

Tan Wang, Chang Zhou, Qianru Sun, and Hanwang Zhang. 2021. Causal attention for unbiased visual recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3091–3100.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

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*.

Junda Wu, Tong Yu, Xiang Chen, Haoliang Wang, Ryan Rossi, Sungchul Kim, Anup Rao, and Julian McAuley. 2024. DeCoT: Debiasing chain-of-thought for knowledge-intensive tasks in large language models via causal intervention. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14073–14087, Bangkok, Thailand. Association for Computational Linguistics.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mt5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pages 483–498. Association for Computational Linguistics.

Xiangji Zeng, Yunliang Li, Yuchen Zhai, and Yin Zhang. 2020. Counterfactual generator: A weakly-supervised method for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 7270–7280. Association for Computational Linguistics.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Chaoran Zhang, Lixin Zou, Dan Luo, Xiangyang Luo, Zihao Li, Min Tang, and Chenliang Li. 2024a. Efficient sparse attention needs adaptive token release. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 14081–14094. Association for Computational Linguistics.

Congzhi Zhang, Linhai Zhang, and Deyu Zhou. 2024b. Causal walk: Debiasing multi-hop fact verification with front-door adjustment. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 19533–19541. AAAI Press.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023a. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*.

Xiang Zhang, Senyu Li, Bradley Hauer, Ning Shi, and Grzegorz Kondrak. 2023b. Don't trust chatgpt when your question is not in english: A study of multilingual abilities and types of llms. *arXiv preprint arXiv:2305.16339*.

Yang Zhang, Fuli Feng, Xiangnan He, Tianxin Wei, Chonggang Song, Guohui Ling, and Yongdong Zhang. 2021. Causal intervention for leveraging popularity bias in recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*, pages 11–20.

Zhihao Zhang, Jun Zhao, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024c. Unveiling linguistic regions in large language models. *CoRR*, abs/2402.14700.

Jun Zhao, Zhihao Zhang, Luhui Gao, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Llama beyond english: An empirical study on language capability transfer. *arXiv preprint arXiv:2401.01055*.

Lixin Zou, Changying Hao, Hengyi Cai, Shuaiqiang Wang, Suqi Cheng, Zhicong Cheng, Wenwen Ye, Simiu Gu, and Dawei Yin. 2022. Approximated doubly robust search relevance estimation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 3756–3765.

Lixin Zou, Shengqiang Zhang, Hengyi Cai, Dehong Ma, Suqi Cheng, Shuaiqiang Wang, Daiting Shi, Zhicong Cheng, and Dawei Yin. 2021. Pre-trained language model based ranking in baidu search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 4014–4022.

A Appendix

We provide prompt templates used for Estimating Average Treatment Effect and testing the Quality-aware language confusion benchmark in Figure 7. Our experiments involve multiple languages; however, here we only provide the Chinese version, and prompts for other languages are obtained through translation.

Naive prompt for Estimating Average Treatment Effect

Question: <question>

Answer: <answer>

Prompt with demonstrations for Estimating Average Treatment Effect

用中文回答问题。

Question: 用几句话描述云计算是什么。

Answer: 云计算是一种依赖于共享计算资源而非使用本地服务器或个人计算机处理应用程序的计算类型。

Question: <question>

Answer:

Prompt with demonstrations for Estimating Average Treatment Effect (*Translation*)

Answer question in Chinese.

Question: Describe cloud computing in a few sentences.

Answer: Cloud computing is a type of computing that relies on shared computing resources rather than using local servers or personal computers to process applications.

Question: <question>

Answer:

Prompt for testing Quality-aware Language Confusion Benchmark

用中文回答问题。

问题: <question>

答案:

Prompt for testing Quality-aware Language Confusion Benchmark (*Translation*)

Answer question in Chinese.

Question: <question>

Answer:

Figure 7: Prompt templates for estimating average treatment effect and evaluating Quality-aware Language Confusion Benchmark.