Language Surgery in Multilingual Large Language Models

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have demonstrated remarkable generalization capabilities across tasks and languages, revolutionizing natural language processing. This paper investigates the naturally emerging representation alignment in LLMs, particularly in the middle layers, and its implications for disentangling language-specific and language-agnostic information. We empirically confirm the existence of this alignment, analyze its behavior in comparison to explicitly designed alignment models, and demonstrate its potential for language-specific manipulation without semantic degradation. Building on these findings, we propose Inference-Time Language Control (ITLC), a novel method that leverages latent injection to enable precise cross-lingual language control and mitigate language confusion in LLMs. Our experiments highlight ITLC's strong cross-lingual control capabilities while preserving semantic integrity in target languages. Furthermore, we demonstrate its effectiveness in alleviating the cross-lingual language confusion problem, which persists even in current large-scale LLMs, leading to inconsistent language generation. This work advances our understanding of representation alignment in LLMs and introduces a practical solution for enhancing their cross-lingual performance. [1]

1 Introduction

Large Language Models (LLMs) have revolutionized natural language processing, demonstrating remarkable generalization capabilities across diverse tasks and languages (Brown et al., 2020; Le Scao et al., 2023; Anil et al., 2023; Team et al., 2025; Cohere et al., 2025; Singh et al., 2025). Their ability to adapt to new tasks in few-shot and even zero-shot settings highlights their efficiency and versatility (Bang et al., 2023; Susanto et al., 2025). Prior works have identified a naturally emerging

[1]We will release the code under the Apache 2.0 license.

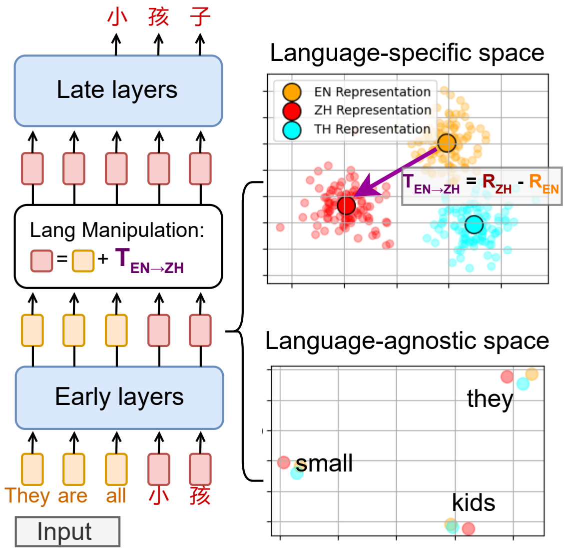


Figure 1: We inspect the alignment in the middle layer representation of LLMs, allowing us to disentangle the language-specific and language-agnostic information. By exploiting this behavior, we are able to achieve Inference-Time Language Control (ITLC), alleviating the language confusion problem in LLMs.

representation alignment across layers in LLMs, particularly in the middle layers of LLMs (Chang et al., 2022; Zhao et al., 2024a). This emerging alignment in LLMs is the key factor in their ability to handle multiple languages (Cahyawijaya, 2024; Tang et al., 2024; Wilie et al., 2025), which is pivotal for their cross-lingual capabilities. However, several questions remain open, such as whether this emerging alignment behaves similarly to alignment in models trained with enforced alignment objectives (Reimers and Gurevych, 2020; Yang et al., 2019a; Feng et al., 2022; Limkonchotiwat et al., 2022, 2024), how this alignment can be utilized to further enhance LLMs, etc.

In this work, we investigate the phenomenon of representation alignment in LLMs, focusing on its occurrence, distinction, and potential applications. We aim to confirm the presence of this representation alignment and contrast it with alignment in LLMs with strictly designed alignment, such as multilingual SentenceBERT (Reimers and Gurevych, 2019) or LaBSE (Feng et al., 2022). Our findings highlight that, unlike LLMs with strictly designed alignment, the naturally emerging alignment in recent LLMs demonstrates a much stronger retention of language-specific information with $\sim$30% performance drop compared to LLMs with strictly designed alignment, with almost >90% performance drops relative to other non-aligned layers.

Upon further investigation, we found a potential method to disentangle language-specific and language-agnostic information in the aligned representation. By exploiting the disentangled language-specific and language-agnostic information, we demonstrate a simple but effective method to control the generation of language from such a representation, enabling us to achieve Inference-Time Language Control (`ITLC`) as showcased in Figure 1. We demonstrate the effectiveness of `ITLC` in two downstream applications: 1) zero-shot cross-lingual language generation and 2) mitigating language confusion in LLMs (Marchisio et al., 2024).

Our contribution in this work is fourfold:

- We confirm the presence of representation alignment in LLMs, providing empirical evidence of this phenomenon (§3.2).
- We contrast natural alignment with strictly designed alignment, highlighting their comparable impact on cross-lingual generalization while emphasizing their differences in alignment locations and the extent of language-specific information retention (§3.2).
- We investigate a method to extract language-specific information from aligned representations, showcasing the potential for language-specific manipulation while preserving the semantic integrity of the generation (§4.1).
- We introduce `ITLC`, a novel method that enables cross-lingual language control and mitigates language confusion problems that retain semantic integrity in target languages (§5).

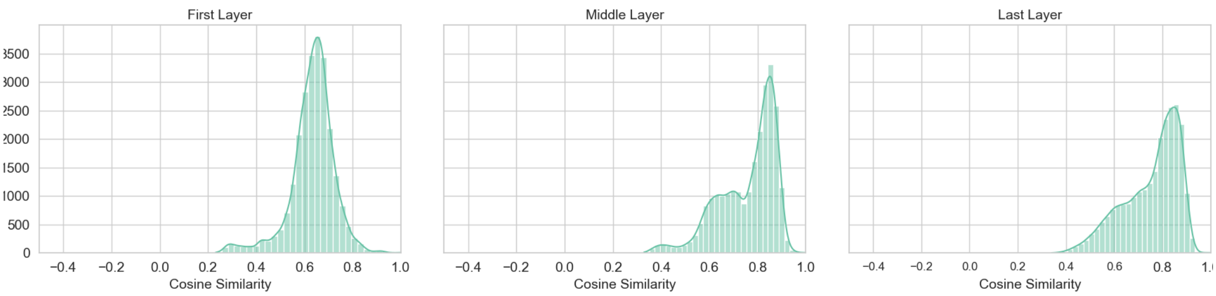
2 Related Work

2.1 Representation Alignment in LLMs

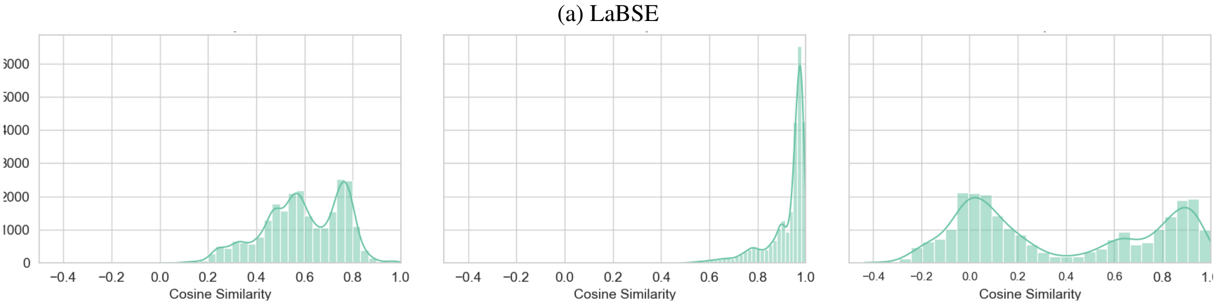
Representation alignment refers to the process by which semantically identical inputs expressed in different languages are mapped to similar internal embeddings within LLMs (Park et al., 2024; Wu and Dredze, 2020; Chang et al., 2022). Originally, representation alignment is strictly embedded into the modeling objective to ensure output consistency across languages and to enable generalization in cross-lingual transfer tasks (Pires et al., 2019; Wu and Dredze, 2019; Reimers and Gurevych, 2020; Feng et al., 2022; Choenni et al., 2024). Several studies have observed a tendency for LLMs to align representations across different languages (Wendler et al., 2024; Zhao et al., 2024b; Mousi et al., 2024). This is done by measuring the similarity between embeddings of parallel sentences across different languages (Ham and Kim, 2021; Gaschi et al., 2023; Cahyawijaya, 2024). Several benchmark datasets can be used for this purpose. Our work measure the degree of alignment across various layers between strictly and naturally aligned models to contrast the two and understand its relation to language-specific and language-agnostic capabilities (Kulshreshtha et al., 2020; Libovický et al., 2020; Hua et al., 2024; Wilie et al., 2025) of LLMs.

2.2 Latent Controllability in LLMs

LLMs controllability is crucial for ensuring that the systems adhere with human intentions. Through mechanisms such as adapter (Pfeiffer et al., 2020; Hu et al., 2022), prompting (Lin et al., 2021; Bai et al., 2022), latent manipulation (Madotto et al., 2020; Ansell et al., 2021), etc, we aim to gain control over the behavior of LLMs. Various aspects have been explored in LLM controllability, including internal knowledge (Madotto et al., 2020; Xu et al., 2022), styles & personas (Lin et al., 2021; Wagner and Ultes, 2024; Cao, 2024), languages (Üstün et al., 2020; Ansell et al., 2021), human values (Bai et al., 2022; Cahyawijaya et al., 2025a), etc. Recent works show that latent states in LLMs exhibit discernible patterns for distinguishing truthful outputs from hallucinated ones, suggesting an intrinsic awareness of fabrication (Li et al., 2023; Duan et al., 2024; Ji et al., 2024; Chen et al., 2024). Similar methods are also introduced for stylistic and safety control (Subramani et al., 2022; Kwak et al., 2023). These studies underscore the potential of latent interventions for precise control over LLM behavior. `ITLC` extends the latent manipulation methods for controlling the generated language in inference time, demonstrating how language-specific information can be extracted and manipulated without losing semantic meaning. This opens new avenues for controlling language generation and mitigating confusion problems.



(a) LaBSE



(b) Qwen2.5-0.5B

Figure 2: Cross-lingual similarity across different layers in LaBSE and Qwen2.5-0.5B. LaBSE exhibits high cross-lingual similarity in its final layer, whereas Qwen2.5-0.5B shows this similarity in the middle layer. This difference suggests that the alignment of representations occurs at distinct positions within the two models.

3 Understanding Representation Alignment in LLMs

Prior works (Chang et al., 2022; Zhao et al., 2024a; Cahyawijaya, 2024; Wilie et al., 2025; Payoungkhamdee et al., 2025) demonstrate the existence of emerging representation alignment in LLMs. We take a step further to provide a deeper understanding to this behavior by contrasting it with alignment in strictly-aligned LLMs. Specifically, we observe the correlation between the degree of alignment with the *cross-lingual generalization* and *language identification* (LID) capability, which are the proxies to their language-agnostic and language-specific capabilities, respectively.

3.1 Experiment Setting

Modeling As a measure of alignment, we compute the average cosine similarity of the latent representation of a sentence in one language with the representation of parallel sentences in the other languages. For the LLM with strictly designed alignment, we employ LaBSE (Feng et al., 2022). For the LLM with emerging representation alignment, we employ multilingual decoder-only LLM, i.e., Qwen2.5 (Qwen et al., 2025). Specifically, we employ Qwen2.5-0.5B with 500M parameters to have a comparable scale with the LaBSE model with 471M parameters. For measuring the cross-lingual generalization of LaBSE, we perform monolingual few-shot fine-tuning with SetFit (Pannerselvam et al., 2024) and evaluate on all other languages. For Qwen2.5-0.5B, we employ few-shot cross-lingual in-context learning. We incorporate 10 few-shot samples for SetFit and 2 for in-context learning. To measure the LID capability, we take the latent representation of both models in the first, middle, and last layers. In this case, we are interested in comparing the behavior between the strictly aligned representation in LaBSE and the emerging aligned representation in Qwen2.5-0.5B. Following Cahyawijaya et al. (2025b), we measure LID performance by linear probing and kNN to measure linear separability and cluster closeness within each language class. More details about the experiment are presented in Appendix A.

Dataset We employ a set of multilingual evaluation datasets. To measure the degree of alignment, we employ 7 datasets: FLORES-200 (Team et al., 2022), NTREX-128 (Federmann et al., 2022), NusaX (Winata et al., 2023), NusaWrites (Cahyawijaya et al., 2023), BUCC (Zweigenbaum et al., 2017), Tatoeba (Tiedemann, 2020), and Bible Corpus (McCarthy et al., 2020). For cross-lingual evaluation, we incorporate 4 datasets: SIB200 (Ade-

Dataset	LaBSE	Qwen2.5-0.5B
SIB200	0.210	0.123
INCLUDE-BASE	-0.021	0.142
XCOPA	0.144	-0.139
PAWS-X	0.146	0.532
Avg	**0.1198**	**0.1645**

Table 1: Pearson correlation between the downstream cross-lingual performance and the degree of alignment between the corresponding language pairs.

lani et al., 2024), INCLUDE-BASE (Sridhar et al., 2020), XCOPA (Ponti et al., 2020), and PAWS-X (Yang et al., 2019b). For LID evaluation, we incorporate 3 datasets, i.e., FLORES-200, NTREX-128, and NusaX. The detailed description of each dataset is shown in Appendix A.

3.2 Experiment Result

Strictly and Naturally Aligned LLMs LaBSE and Qwen2.5-0.5B demonstrate distinct patterns in cross-lingual representation alignment. As shown in Figure 2, LaBSE demonstrates a distributed alignment strength across deeper layers, with the middle and last layers achieving high average similarity scores (0.758 and 0.754, respectively). This aligns with the training objective of LaBSE, which aligns the representation on the last layer. In contrast, Qwen2.5-0.5B exhibits a more localized alignment pattern, with the middle layer showing a strikingly higher average similarity (0.922) than both the first (0.591) and last (0.375) layers. This suggests that Qwen2.5-0.5B concentrates representation alignment sharply in the middle layer, achieving both higher and more stable cross-lingual representation. See detailed analysis in Appendix B.1.

This result displays distinct layer-wise behaviors in retaining the language-specific and language-agnostic information within the two different types of LLMs. Specifically, for model with strict alignment, aligned representation is located in the layer where the objective is applied to – the last layer in the case of LaBSE –, while in LLMs with natural alignment, the aligned representation is formed in the middle layers and breaks as the representation goes closer into the last layer. This aligns with prior works (Chang et al., 2022; Tang et al., 2024; Zhao et al., 2024a; Wilie et al., 2025), which demonstrate the naturally representation alignment emerge in the middle layer of LLMs.

Representation Alignment and Cross-Lingual Generalization We further measure the impact of the degree of representation alignment to the downstream cross-lingual generalization capability of the models. We measure the cross-lingual performance by training on one language and evaluating on the other languages in the datasets using the method described in §3.1 and correlate them with the cosine similarity of the corresponding language pair averaged across all alignment datasets. As shown in Table 1, the degree of alignment shows a positive correlation to the downstream cross-lingual performance. Nonetheless, the correlation is weak, which is potentially caused by the few-shot tuning setting conducted on both models. Despite that, the positive correlation between degree of alignment and cross-lingual generalization on both strictly-aligned and naturally-aligned LLMs signifies the important role of representation alignment in improving the cross-lingual generalization of LLMs. See Appendix B.2 for detailed and further analysis.

Representation Alignment and Language-Specific Information As shown in Table 2, the LID performance of LaBSE and Qwen2.5-0.5B models evaluated using both KNN and linear probing reveals that the first layer consistently achieves the highest LID F1 scores across all datasets. For LaBSE, the aligned representation in the last layer exhibits notably weaker performance, particularly for the FLORES-200 and NusaX datasets. Similarly, in Qwen2.5-0.5B, the aligned representation in the middle layer shows weaker LID performance compared to the first and last layers. These empirical findings highlight three key insights: (1) language-specific information, such as surface-level features and general linguistic patterns, is more dominant in the early layers; (2) the degree of alignment is negatively correlated with the amount of language-specific information retained; and (3) unlike strictly aligned LLMs, the aligned representation in LLMs with emerging alignment retains more language-specific information, which potentially serves as the basis for determining the language of the generated sequence.

4 Inference-Time Language Control

Building on the insights presented in §3, we explore a method to control the language of the generated sequence with minimal semantic loss. Specifically, we develop a method to extract language-

Method	Layer	LaBSE FLORES-200	LaBSE NTREX-128	LaBSE NusaX	Qwen2.5-0.5B FLORES-200	Qwen2.5-0.5B NTREX-128	Qwen2.5-0.5B NusaX
Linear Probing	First	95.13	93.29	97.30	94.21	91.42	95.55
	Middle	94.18	92.68	94.51	**91.76**	**90.04**	**87.09**
	Last	**70.89**	**74.36**	**65.44**	92.46	90.27	88.77
KNN	First	88.35	90.43	81.78	83.69	86.06	65.79
	Middle	78.85	81.30	45.37	**55.32**	**54.73**	**25.05**
	Last	**3.92**	**1.63**	**0.00**	71.73	81.86	29.39

Table 2: LID performance by layer and classification method for LaBSE and QWEN2.5-0.5B. **Red bold text** highlights the LID scores on the layer where alignment occurs in each corresponding model. LID performance is consistently lower in a layer where the representation is aligned across all models and classification methods.

specific information at the layer where representation alignment occurs in LLMs. Using this information, we gather language-specific vectors from each language and use them to manipulate the language-specific information during the inference time. With this language-specific intervention, we aim to steer the model toward utilizing language-specific features, allowing us to perform Inference-Time Language Control (`ITLC`).

4.1 Methods

Latent Extraction Latent extraction techniques are employed to isolate language-specific information from the model's representations. Specifically, we extract Qwen2.5-0.5B (Qwen et al., 2025) hidden states to capture language-specific features at its middle representation. Given an input sequence from the FLORES-200 dataset (Team et al., 2022), we compute the hidden states $\mathbf{h} \in \mathbb{R}^d$ at a specified layer, where $d = 896$ is the embedding dimension of Qwen2.5-0.5B. Finally, we apply mean pooling to ensure that only meaningful token embeddings contribute to the final representation.

Linear Discriminant Analysis To disentangle language-specific information, we apply Linear Discriminant Analysis (LDA) to maximize class separability and reduce dimensionality. We use the Singular Value Decomposition (SVD) solver in order to handle high-dimensional embeddings efficiently and select the top k eigenvectors corresponding to the largest eigenvalues to form $\mathbf{W} \in \mathbb{R}^{d\times k}$. Let $\mathcal{D} = \{(\mathbf{h}_i, l_i)\}_{i=1}^N$ denote a dataset of hidden states $\mathbf{h}_i \in \mathbb{R}^d$ labeled with language classes $l_i \in \{1, \ldots, K\}$, this projects hidden states to a lower-dimensional space $\mathbf{z} = \mathbf{h}^T\mathbf{W} \in \mathbb{R}^k$.

To validate the quality of the projection and select the optimal number of components k, we train a neural network classifier with a single linear layer on the projected training data $\mathbf{z}$. We experiment with several k values and evaluate classification accuracy on a test set. Finally, we take $k = 100$ because LID performance significantly drops on higher components, indicating a major loss of language-specific information. More details on the LDA settings are shown in Appendix D.2

Language Vector Using the LDA-projected space, we construct language vectors by leveraging the neural network's weights to identify active dimensions for each language. For each language l we extract the weight matrix $\mathbf{U} \in \mathbb{R}^{K\times k}$ from the neural network's linear layer, where $u_{l,j}$ represents the contribution of dimension $j \in \{1, \ldots, k\}$ to language l. We define a threshold $\tau = 0.01$ and select active dimensions for language l as $\mathcal{A}_l = \{j \mid |u_{l,j}| > \tau\}$.

The language vector $\mathbf{v}_l \in \mathbb{R}^k$ for language l is computed as the mean of projected hidden states $\mathbf{z}_i$ over samples of language l, restricted to active dimensions:

$$\mathbf{v}_l[j] = \begin{cases} \frac{1}{N_l} \sum_{\mathbf{h}_i \in l} \mathbf{z}_i[j], & \text{if } j \in \mathcal{A}_l, \\ 0, & \text{otherwise,} \end{cases} \quad (360)$$

where N_l is the number of samples for language l, and $\mathbf{z}_i[j]$ is the j-th component of the projected hidden state.

Vector Injection To enable injection, we project the language vector back to the original embedding space using the pseudo-inverse: $\mathbf{v}_l^{\text{orig}} = \mathbf{v}_l\mathbf{W}^\dagger \in \mathbb{R}^d$. By applying this, we retain the original embedding of the input and modify it with the language vector inverse projection. For a source language x (e.g., English) and target language y (e.g., Indonesian), we compute a shift vector:

$$\delta = -\mathbf{v}_x^{\text{orig}} + \mathbf{v}_y^{\text{orig}},$$

which is injected into the hidden states at the middle layer during inference:

$$\mathbf{h}' = \mathbf{h} + \alpha\delta,$$

where $\mathbf{h}$ is the original hidden state, α is a scaling factor, and $\mathbf{h}'$ is the modified hidden state.

Language Shift Strategy We further divide the language vector injection into three strategies based on the temporal scope of application: (1) prompt only, (2) generated tokens only, and (3) both phases. Let $\mathbf{h}_t^{(m)} \in \mathbb{R}^d$ denote the hidden state at position t in the middle layer m, and $\mathbf{h}_t^{(m)\prime}$ denotes its language-shifted counterpart:

- **Prompt-Only** (`prompt-only`): Applies injection exclusively to input prompt processing:

$$\mathbf{h}_t^{(m)\prime} = \begin{cases} \mathbf{h}_t^{(m)} + \alpha\delta, & \forall t \in [1, T_{\text{input}}] \\ \mathbf{h}_t^{(m)}, & \forall t > T_{\text{input}} \end{cases}$$

- **Generated-Only** (`gen-only`): Restricts injection to autoregressive generation:

$$\mathbf{h}_t^{(m)\prime} = \begin{cases} \mathbf{h}_t^{(m)}, & \forall t \in [1, T_{\text{input}}] \\ \mathbf{h}_t^{(m)} + \alpha\delta, & \forall t \in [T_{\text{input}} + 1, T_{\text{total}}] \end{cases}$$

- **Prompt and Generated** (`prompt-and-gen`): Applies injection throughout both phases:

$$\mathbf{h}_t^{(m)\prime} = \mathbf{h}_t^{(m)} + \alpha\delta, \quad \forall t \in [1, T_{\text{total}}]$$

where T_{input} is the input prompt length and $T_{\text{total}} = T_{\text{input}} + N$ the total sequence length after generating N tokens.

5 Implication of `Inference-Time Language Control` (`ITLC`)

We demonstrate the effectiveness of `ITLC` on two scenarios: 1) cross-lingual language control and 2) mitigating language confusion (Marchisio et al., 2024). Cross-lingual language control refers to guiding the model prompted on a source language to switch and generate text in a target language (e.g., EN→XX or XX→EN) by manipulating its latent representation while maintaining semantic relevance and linguistic fidelity across different languages. Mitigating language confusion, on the other hand, focuses on alleviating the limitation of LLMs to consistently generate text in the desired language, which can occur at the word level, line level, or entire response (Marchisio et al., 2024).

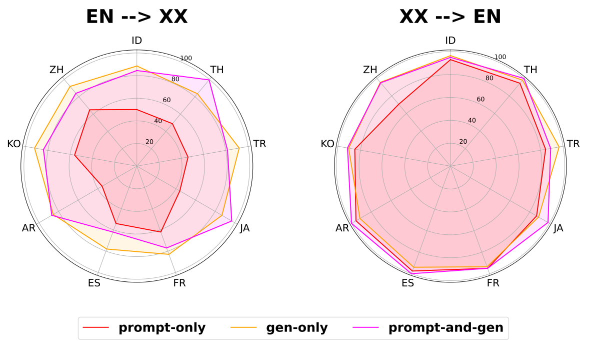


Figure 3: Language correctness (%) for Qwen2.5-0.5B across EN→XX and XX→EN Directions. The result of its instruct version is shown at Appendix D.3.

5.1 Experiment Setting

Cross-lingual Language Control we investigate cross-lingual language control using the Qwen2.5-0.5B model with $\alpha = 1.0$. We utilize the Dolly multilingual dataset subset [2] by taking 200 QA sentences in nine various languages from diverse regions and language families: Indonesian (**ID**), Thai (**TH**), Turkish (**TR**), Japanese (**JA**), French (**FR**), Spanish (**ES**), Arabic (**AR**), Chinese (**ZH**), and Korean (**KO**). We evaluate the performance with multiple automatic metrics: FastText LID (Joulin et al., 2016a,b) for language correctness, while BertScore (Zhang et al., 2020) and SacreBLEU (Papineni et al., 2002; Post, 2018) for generation quality. We conduct a human evaluation with native annotators on English (**EN**), Indonesian (**ID**), and Thai (**TH**) in both EN→XX and XX→EN directions to further validate the generation quality. The human evaluation focuses on 30 samples covering 3 aspects: **naturalness**, prompt-completion **relevance**, and **answer correctness** using likert score ranging from [1. . . 5]. The human annotation guideline is presented in Appendix F.

Mitigating Cross-Lingual Language Confusion We evaluate the effectiveness of `ITLC` in mitigating language confusion on the base and instruct versions of Qwen2.5-0.5B. For the hyperparameter α, we use the optimal scaling parameter identified in our ablation study (see Appendix E for details). We analyze the impact of language shift vectors extracted from both the base and instruct models (base shift vector δbase vs. instruct shift vector δinstruct) on cross-lingual control. We also investigate the influence of chat template formatting

[2] `https://huggingface.co/datasets/CohereLabs/aya_evaluation_suite/viewer/dolly_machine_translated`

Lang Shift	Prompt	Generated
EN → ID	What is investment banking?	Investment bank adalah sebuah perusahaan yang menawarkan layanan investasi dan jasa lainnya kepada para investor atau wirausahawan dalam rangka mendirikan berbagai jenis finansial seperti saham (aset pelanggan) . . .
ID → EN	Tuliskan saya email pengantar ke pengacara untuk mewakili saya di pengadilan untuk tiket lampu merah saya tidak pantas	I'll try to draft an outline for what I might say in my speech as the "tug of war" between two candidates on behalf of people who want their lights out and someone else's lighted street: A good man will tell me that . . .

Table 3: Examples of generated outputs from Qwen2.5-0.5B-Instruct with injection. Overall, it shows that the language has successfully shifted to the desired target language and the answers are generally correct. Additional examples across different language pairs are provided in Appendix D.5.

Target	Baseline		`ITLC`	
Language	BLEU	BERTScore	BLEU	BertScore
ID	**19.29**	62.9	14.3	**63.6**
TH	0.0	62.8	**15.97**	**64.1**
TR	6.05	**60.7**	**15.97**	60.2
JA	0.0	**62.0**	**15.97**	60.2
FR	7.78	**63.3**	**10.97**	63.2
ES	**10.88**	64.0	7.17	**64.4**
AR	7.13	63.8	**11.88**	**65.5**
ZH	0.0	**63.6**	0.0	62.0
KO	**7.54**	63.1	4.11	**63.8**
AVG	6.52	62.91	**10.70**	**63.00**

Table 4: Generation performance for different target languages with Qwen2.5-0.5B-Instruct. **Baseline** denotes the same model prompted in the same language as the desired target language.

Model	Lang Shift	Nat.	Rel.	Cor.
Baseline	ID→ID	1.17	1.17	1.13
	EN→EN	2.80	2.37	2.07
	TH→TH	1.70	1.33	1.13
`ITLC`	ID→EN	4.73	3.17	1.43
	EN→ID	3.43	2.29	1.46
	TH→EN	1.73	1.30	1.27
	EN→TH	1.10	1.07	1.07

Table 5: Human evaluation of `ITLC` response quality. **Nat.**, **Rel.**, and **Cor.** respectively denote naturalness, relevance, and answer correctness ranging from [1 . . . 5]. **Baseline** denotes the same model prompted in the same language as the desired target language.

and few-shot examples on model behavior. Our evaluation focuses on cross-lingual settings where input and target languages differ, and reports the official metrics defined in Marchisio et al. (2024): Language Confusion Pass Rate (LCPR), Line-level Pass Rate (LPR), and Word-level Pass Rate (WPR).

5.2 Results

5.2.1 Cross-Lingual Language Control

Language Vector Impact Our experiments demonstrate that the proposed `ITLC` method enables effective control over cross-lingual generation. As shown in Table 3, both EN→XX and XX→EN directions yield a higher rate of correct language identification. This suggests that manipulating representations of language-specific spaces helps align the source language more closely with the target language in a newly projected representation space. For more details results and comparisons are shown in Appendix D.3 and D.4. Interestingly, this space not only transforms the representation into the desired target language, but also carries rich semantic information that contributes to more meaningful and contextually accurate generation. This is further supported by the results in Table 4 (for qualitative evidence, refer to Table 3), where we compare our method against a monolingual baseline—prompted and completed entirely in the target language—as an upper bound. Notably, `ITLC` achieves comparable and, in many cases, surpasses the baseline in both metrics, indicating closer resulting generations to the ground truth answer in the target language. These findings highlight that the injection strategy enables effective cross-lingual generation while preserving semantic integrity.

Human Evaluation We conducted a human evaluation to validate our findings on language vector injection, with the overall results summarized in Table 5. Our method performs comparably to the Mono Baseline, demonstrating that `ITLC` successfully shifts language and performs cross-lingual generation close to the ideal performance for the target language. Notably, the direction toward Indonesian even outperforms its baseline by 1–2

Method	LCPR	LPR	WPR
Qwen2.5-0.5B			
Baseline	29.41	19.75	73.45
+ Q/A template (0-shot)	44.68	35.36	75.94
+ 5-shot	56.78	50.63	76.16
+ `ITLC` (apply base shift vector)			
+ prompt-only (0.8)	65.71	66.41	74.24
+ gen-only (0.6)	71.35	80.46	67.67
+ prompt-and-gen (0.5)	**78.93**	**85.08**	**77.15**
Qwen2.5-0.5B-Instruct			
Baseline (w/ chat template)	63.00	57.69	79.50
+ 5-shot	63.53	58.79	75.34
+ `ITLC` (apply base shift vector)			
+ seq-only (0.8)	76.05	77.68	**81.11**
+ gen-only (0.6)	75.56	82.42	74.51
+ prompt-and-gen (0.5)	**81.51**	85.32	80.55
+ `ITLC` (apply instruct shift vector)			
+ prompt-only (0.8)	73.26	76.37	79.20
+ gen-only (0.6)	73.95	84.06	71.40
+ seq-and-gen (0.5)	80.96	**86.79**	78.84

Table 6: Cross-lingual language confusion performance (LCPR / LPR / WPR) of Qwen2.5-0.5B models.

points, suggesting particularly strong alignment in that case. Specifically, the XX→EN direction suggests the presence of strong latent representations. In contrast, the EN→XX direction shows reduced performance across both settings, highlighting persistent challenges in generating text for low-resource languages. Nonetheless, the Qwen2.5-0.5B-Instruct model used in this study is a relatively small model, which limits the quality of its language generation. Despite this limitation, our results demonstrate that injecting the language vector into the latent space can effectively guide the model toward cross-lingual generation.

5.2.2 Mitigating Language Confusion

Crosslingual Language Control and Prompting Efficacy As shown in Table 6, our proposed method, `ITLC`, surpasses baseline configurations, including QA/chat templates and 5-shot prompting for both base and instruct models in crosslingual settings. The seq-and-gen strategy with language shift vectors achieves the strongest performance. For the base model, crosslingual performance improves progressively with few-shot [3] examples, as few-shot examples utilize English inputs with explicit target-language, reinforcing input-output alignment. In contrast, the instruct model exhibits minimal variation across few-shot configurations, as its instruction-tuning inherently supports multilingual prompting without dependency on few-shot quantity. These results demonstrate that our approach enhances crosslingual language consistency while accommodating architectural differences between base and instruct models.

Transferability of Language Vector to Post-Trained Models. Interestingly, as shown on the Qwen2.5-0.5B Instruct in Table 6, applying language vectors gathered from the base model to the instruct model achieves comparable performance to its native instruct vectors which suggests the effectiveness of language shift from the base model for crosslingual control even in the instruct model, despite the representation space of the model is already shifted by post-training which covers instruction tuning, preference-tuning and/or RLHF. This transferability indicates that the relative distance between language-specific and that the resulting language-specific features from the pre-training phase is robust to downstream adaptation, including tasks generalization from instruction-tuning and value alignment in RLHF and preference-tuning. This evidence implies that – in the case of the Qwen2.5 model family – the cross-lingual symmetry – i.e., the geometric alignment between language representations – constructed during the fine-tuning is preserved even after various downstream refinement of the model. The preservation of these relationships implies that language-specific cues are retained as invariant properties across model versions, enabling consistent cross-lingual language control through `ITLC` despite parameter updates during downstream fine-tuning, instruction-tuning, preference-tuning, and RLHF.

6 Conclusion

Our work explores the phenomenon of representation alignment in LLMs, confirming its occurrence and elucidating its behavior compared to strictly designed alignment models. We have demonstrated the potential for disentangling language-specific and language-agnostic information, enabling effective language-specific manipulation without semantic loss. Furthermore, we have shown the practical applications of language control manipulation in enhancing language control and mitigating confusion problems. Our findings contribute to a deeper understanding of representation alignment in LLMs and open new avenues for improving their performance in multilingual settings.

[3] Cross-lingual prompts follow the official format: English inputs with instructions like `Respond in <TARGET_LANG>`.

Limitations

The study has several limitations that should be considered when interpreting the results. First, the coverage of LLMs is limited to a specific set of models, particularly Qwen and LaBSE and only one model size (0.5B parameters), which may not be representative of all LLMs. The findings may not generalize to other models with different architectures or training data, as the behavior of representation alignment and language control can vary significantly across different LLMs. Future research should aim to include a more diverse range of models to validate the generalizability of the results.

Second, the evaluation is conducted on a limited number of languages, especially the evaluation of the KNN-based LID method is limited to languages included in the FLORES-200, which may not capture the full spectrum of linguistic diversity. The study focuses on a subset of languages, and the results may not extend to languages with different typological features or those that are underrepresented in the training data. Expanding the evaluation to include a broader range of languages, especially low-resource languages, would provide a more comprehensive understanding of the model's capabilities and limitations.

Additionally, the human evaluation is based on only 30 samples per language, which may not provide a comprehensive assessment of the model's performance. While the sample size is sufficient for preliminary analysis, a larger dataset would be necessary to draw more robust conclusions. Increasing the number of samples and involving a more diverse group of evaluators could enhance the reliability and validity of the findings.

Ethical Considerations

The research involves the use of LLMs, which might raise ethical considerations regarding bias, fairness, and transparency on the generated results. To ensure ethical conduct, the study adheres to the following principles: (1) Bias Mitigation: The models used are evaluated for potential biases, and efforts are made to mitigate any identified biases. (2) Fairness: The evaluation is conducted across multiple languages from diverse regions and language families to ensure fairness and inclusivity. (3) Transparency: The methodology and results are presented transparently to allow for replication and verification. (4) Privacy: No personal data is used in the evaluation, and all data is anonymized to protect privacy. (5) Accountability: The researchers take responsibility for the ethical implications of the study and are committed to addressing any concerns that may arise.

We also acknowledge that our research utilized AI tools for writing, rewriting, and generating code. Although these tools offer significant advantages in terms of efficiency and productivity, their use raises important ethical considerations. We recognize the potential for bias and errors inherent in AI-generated content and have taken steps to mitigate these risks through rigorous human review and validation. Furthermore, we are mindful of the potential impact on the broader software development community, particularly regarding job displacement and the need for upskilling. We believe that responsible AI integration should prioritize transparency, accountability, and the empowerment of human developers, ensuring that these tools augment rather than replace human expertise. This research aims to contribute to the ongoing dialogue on ethical AI development and usage, advocating for a future where AI tools are harnessed responsibly to enhance human creativity and innovation in the field of software engineering.

References

David Ifeoluwa Adelani, Hannah Liu, Xiaoyu Shen, Nikita Vassilyev, Jesujoba O. Alabi, Yanke Mao, Haonan Gao, and En-Shiun Annie Lee. 2024. SIB-200: A simple, inclusive, and big evaluation dataset for topic classification in 200+ languages and dialects. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 226–245, St. Julian's, Malta. Association for Computational Linguistics.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, and 1 others. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Alan Ansell, Edoardo Maria Ponti, Jonas Pfeiffer, Sebastian Ruder, Goran Glavaš, Ivan Vulić, and Anna Korhonen. 2021. MAD-G: Multilingual adapter generation for efficient cross-lingual transfer. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4762–4781, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron

McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional ai: Harmlessness from ai feedback. *Preprint*, arXiv:2212.08073.

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan Xu, and Pascale Fung. 2023. A multitask, multilingual, multimodal evaluation of ChatGPT on reasoning, hallucination, and interactivity. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–718, Nusa Dua, Bali. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Samuel Cahyawijaya. 2024. Llm for everyone: Representing the underrepresented in large language models. *Preprint*, arXiv:2409.13897.

Samuel Cahyawijaya, Delong Chen, Yejin Bang, Leila Khalatbari, Bryan Wilie, Ziwei Ji, Etsuko Ishii, and Pascale Fung. 2025a. High-dimension human value representation in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5303–5330, Albuquerque, New Mexico. Association for Computational Linguistics.

Samuel Cahyawijaya, Delong Chen, Yejin Bang, Leila Khalatbari, Bryan Wilie, Ziwei Ji, Etsuko Ishii, and Pascale Fung. 2025b. High-dimension human value representation in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5303–5330, Albuquerque, New Mexico. Association for Computational Linguistics.

Samuel Cahyawijaya, Holy Lovenia, Fajri Koto, Dea Adhista, Emmanuel Dave, Sarah Oktavianti, Salsabil Akbar, Jhonson Lee, Nuur Shadieq, Tjeng Wawan Cenggoro, Hanung Linuwih, Bryan Wilie, Galih Muridan, Genta Winata, David Moeljadi, Alham Fikri Aji, Ayu Purwarianti, and Pascale Fung. 2023. NusaWrites: Constructing high-quality corpora for underrepresented and extremely low-resource languages. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 921–945, Nusa Dua, Bali. Association for Computational Linguistics.

Lang Cao. 2024. Learn to refuse: Making large language models more controllable and reliable through knowledge scope limitation and refusal mechanism. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3628–3646, Miami, Florida, USA. Association for Computational Linguistics.

Tyler Chang, Zhuowen Tu, and Benjamin Bergen. 2022. The geometry of multilingual language model representations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 119–136, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chao Chen, Kai Liu, Ze Chen, Yi Gu, Yue Wu, Mingyuan Tao, Zhihang Fu, and Jieping Ye. 2024. INSIDE: LLMs' internal states retain the power of hallucination detection. In *The Twelfth International Conference on Learning Representations*.

Rochelle Choenni, Dan Garrette, and Ekaterina Shutova. 2024. How do languages influence each other? studying cross-lingual data sharing during lm fine-tuning. *Preprint*, arXiv:2305.13286.

Team Cohere, :, Aakanksha, Arash Ahmadian, Marwan Ahmed, Jay Alammar, Milad Alizadeh, Yazeed Alnumay, Sophia Althammer, Arkady Arkhangorodsky, Viraat Aryabumi, Dennis Aumiller, Raphaël Avalos, Zahara Aviv, Sammie Bae, Saurabh Baji, Alexandre Barbet, Max Bartolo, Björn Bebensee, and 211 others. 2025. Command a: An enterprise-ready large language model. *Preprint*, arXiv:2504.00698.

Hanyu Duan, Yi Yang, and Kar Yan Tam. 2024. Do llms know about hallucination? an empirical investigation of llm's hidden states. *Preprint*, arXiv:2402.09733.

Christian Federmann, Tom Kocmi, and Ying Xin. 2022. NTREX-128 – news test references for MT evaluation of 128 languages. In *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, pages 21–24, Online. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic bert sentence embedding. *Preprint*, arXiv:2007.01852.

Felix Gaschi, Patricio Cerda, Parisa Rastin, and Yannick Toussaint. 2023. Exploring the relationship between alignment and cross-lingual transfer in multilingual transformers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3020–3042, Toronto, Canada. Association for Computational Linguistics.

Jiyeon Ham and Eun-Sol Kim. 2021. Semantic alignment with calibrated similarity for multilingual sentence embedding. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages

1781–1791, Punta Cana, Dominican Republic. Association for Computational Linguistics.

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 *International Conference on Learning Representations*.

Tianze Hua, Tian Yun, and Ellie Pavlick. 2024. mOthello: When do cross-lingual representation alignment and cross-lingual transfer emerge in multilingual models? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1585–1598, Mexico City, Mexico. Association for Computational Linguistics.

Ziwei Ji, Delong Chen, Etsuko Ishii, Samuel Cahyawijaya, Yejin Bang, Bryan Wilie, and Pascale Fung. 2024. LLM internal states reveal hallucination risk faced with a query. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 88–104, Miami, Florida, US. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016b. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Saurabh Kulshreshtha, Jose Luis Redondo Garcia, and Ching-Yun Chang. 2020. Cross-lingual alignment methods for multilingual BERT: A comparative study. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 933–942, Online. Association for Computational Linguistics.

Jin Myung Kwak, Minseon Kim, and Sung Ju Hwang. 2023. Language detoxification with attribute-discriminative latent space. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10149–10171, Toronto, Canada. Association for Computational Linguistics.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, and 1 others. 2023. Bloom: A 176b-parameter open-access multilingual language model.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. 2023. Inference-time intervention: Eliciting truthful answers from a language model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Jindřich Libovický, Rudolf Rosa, and Alexander Fraser. 2020. On the language neutrality of pre-trained multilingual representations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1663–1674, Online. Association for Computational Linguistics.

Peerat Limkonchotiwat, Wuttikorn Ponwitayarat, Lalita Lowphansirikul, Potsawee Manakul, Can Udomcharoenchaikit, Ekapol Chuangsuwanich, and Sarana Nutanong. 2024. McCrolin: Multi-consistency cross-lingual training for retrieval question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 2780–2793, Miami, Florida, USA. Association for Computational Linguistics.

Peerat Limkonchotiwat, Wuttikorn Ponwitayarat, Can Udomcharoenchaikit, Ekapol Chuangsuwanich, and Sarana Nutanong. 2022. CL-ReLKT: Cross-lingual language knowledge transfer for multilingual retrieval question answering. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2141–2155, Seattle, United States. Association for Computational Linguistics.

Zhaojiang Lin, Zihan Liu, Genta Indra Winata, Samuel Cahyawijaya, Andrea Madotto, Yejin Bang, Etsuko Ishii, and Pascale Fung. 2021. XPersona: Evaluating multilingual personalized chatbot. In *Proceedings of the 3rd Workshop on Natural Language Processing for Conversational AI*, pages 102–112, Online. Association for Computational Linguistics.

Andrea Madotto, Samuel Cahyawijaya, Genta Indra Winata, Yan Xu, Zihan Liu, Zhaojiang Lin, and Pascale Fung. 2020. Learning knowledge bases with parameters for task-oriented dialogue systems. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2372–2394, Online. Association for Computational Linguistics.

Kelly Marchisio, Wei-Yin Ko, Alexandre Berard, Théo Dehaze, and Sebastian Ruder. 2024. Understanding and mitigating language confusion in LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6653–6677, Miami, Florida, USA. Association for Computational Linguistics.

Arya D. McCarthy, Rachel Wicks, Dylan Lewis, Aaron Mueller, Winston Wu, Oliver Adams, Garrett Nicolai, Matt Post, and David Yarowsky. 2020. The Johns Hopkins University Bible corpus: 1600+ tongues for typological exploration. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2884–2892, Marseille, France. European Language Resources Association.

Basel Mousi, Nadir Durrani, Fahim Dalvi, Majd Hawasly, and Ahmed Abdelali. 2024. Exploring alignment in shared cross-lingual spaces. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6326–6348, Bangkok, Thailand. Association for Computational Linguistics.

Kathiravan Pannerselvam, Saranya Rajiakodi, Sajeetha Thavareesan, Sathiyaraj Thangasamy, and Kishore

Ponnusamy. 2024. SetFit: A robust approach for offensive content detection in Tamil-English code-mixed conversations using sentence transfer fine-tuning. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*, pages 35–42, St. Julian's, Malta. Association for Computational Linguistics.

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

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Patomporn Payoungkhamdee, Pume Tuchinda, Jinheon Baek, Samuel Cahyawijaya, Can Udomcharoenchaikit, Potsawee Manakul, Peerat Limkonchotiwat, Ekapol Chuangsuwanich, and Sarana Nutanong. 2025. Towards better understanding of program-of-thought reasoning in cross-lingual and multilingual environments. *Preprint*, arXiv:2502.17956.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

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.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Shivalika Singh, Angelika Romanou, Clémentine Fourrier, David I. Adelani, Jian Gang Ngui, Daniel Vila-Suero, Peerat Limkonchotiwat, Kelly Marchisio, Wei Qi Leong, Yosephine Susanto, Raymond Ng, Shayne Longpre, Wei-Yin Ko, Sebastian Ruder, Madeline Smith, Antoine Bosselut, Alice Oh, Andre F. T. Martins, Leshem Choshen, and 5 others. 2025. Global mmlu: Understanding and addressing cultural and linguistic biases in multilingual evaluation. *Preprint*, arXiv:2412.03304.

Advaith Sridhar, Rohith Gandhi Ganesan, Pratyush Kumar, and Mitesh Khapra. 2020. Include: A large scale dataset for indian sign language recognition. MM '20. Association for Computing Machinery.

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

Yosephine Susanto, Adithya Venkatadri Hulagadri, Jann Railey Montalan, Jian Gang Ngui, Xian Bin Yong, Weiqi Leong, Hamsawardhini Rengarajan, Peerat Limkonchotiwat, Yifan Mai, and William Chandra Tjhi. 2025. Sea-helm: Southeast asian holistic evaluation of language models. *Preprint*, arXiv:2502.14301.

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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5701–5715, Bangkok, Thailand. Association for Computational Linguistics.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

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.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly Universal Dependency parsing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2302–2315, Online. Association for Computational Linguistics.

Nicolas Wagner and Stefan Ultes. 2024. On the controllability of large language models for dialogue interaction. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 216–221, Kyoto, Japan. Association for Computational Linguistics.

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

Bryan Wilie, Samuel Cahyawijaya, Junxian He, and Pascale Fung. 2025. High-dimensional interlingual representations of large language models. *Preprint*, arXiv:2503.11280.

Genta Indra Winata, Alham Fikri Aji, Samuel Cahyawijaya, Rahmad Mahendra, Fajri Koto, Ade Romadhony, Kemal Kurniawan, David Moeljadi, Radityo Eko Prasojo, Pascale Fung, Timothy Baldwin, Jey Han Lau, Rico Sennrich, and Sebastian Ruder. 2023. NusaX: Multilingual parallel sentiment dataset for 10 Indonesian local languages. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 815–834, Dubrovnik, Croatia. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2019. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 833–844, Hong Kong, China. Association for Computational Linguistics.

Shijie Wu and Mark Dredze. 2020. Do explicit alignments robustly improve multilingual encoders? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4471–4482, Online. Association for Computational Linguistics.

Yan Xu, Etsuko Ishii, Samuel Cahyawijaya, Zihan Liu, Genta Indra Winata, Andrea Madotto, Dan Su, and Pascale Fung. 2022. Retrieval-free knowledge-grounded dialogue response generation with adapters. In *Proceedings of the Second DialDoc Workshop on Document-grounded Dialogue and Conversational Question Answering*, pages 93–107, Dublin, Ireland. Association for Computational Linguistics.

Yinfei Yang, Gustavo Hernandez Abrego, Steve Yuan, Mandy Guo, Qinlan Shen, Daniel Cer, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2019a. Improving multilingual sentence embedding using bi-directional dual encoder with additive margin softmax. *arXiv preprint arXiv:1902.08564*.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. 2019b. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China. Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024a. How do large language models handle multilingualism? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

Pierre Zweigenbaum, Serge Sharoff, and Reinhard Rapp. 2017. Overview of the second BUCC shared task: Spotting parallel sentences in comparable corpora. In *Proceedings of the 10th Workshop on Building and Using Comparable Corpora*, pages 60–67, Vancouver, Canada. Association for Computational Linguistics.

A Details of All Evaluation Datasets

The following tables present the full details of dataset sizes used in this study. Refer to Table 7, Table 8, Table 9, Table 10 and Table 11.

B Detail Experiment for Understanding Representation Alignment in LLMs

B.1 Cosine Similarity Distributions Across Datasets

To better understand the representational behavior of the models, we analyzed the distribution of cosine similarity scores across layers. For LaBSE, the average cosine similarity increases from the first layer (mean = 0.6335, std = 0.0920) to the middle layer (mean = 0.7580, std = 0.1182), and remains comparably high in the last layer (mean = 0.7544, std = 0.1150). This trend suggests that semantic alignment becomes stronger toward the middle and final layers, with relatively low variability, indicating consistent behavior across input samples. These observations align with prior findings that intermediate layers in multilingual encoders often capture the most transferable features.

In contrast, Qwen2.5-0.5B exhibits a markedly different pattern. While the middle layer achieves the highest average similarity (mean = 0.9218, std = 0.0871), the first layer has a lower mean and higher variance (mean = 0.5913, std = 0.1650), indicating less stable representations early in the network. Notably, the last layer shows a substantial drop in similarity (mean = 0.3745) and a sharp increase in variability (std = 0.3988), suggesting a divergence in representational behavior, potentially due to task-specific tuning or greater representational fragmentation. This may help explain the weaker correlations between cosine similarity and task performance observed in Qwen's final layers.

These findings reinforce the role of middle layers in capturing semantically meaningful and transferable representations, particularly in instruction-tuned or general-purpose multilingual models. See Figure 2 for the histogram plot and Figure 4 for the bar chart per alignment dataset.

B.2 Additional Analysis For Alignment and Downstream Correlation

As shown in Table 12, the correlation between cosine similarity and downstream performance varies by dataset, layer, and model architecture. The following sections provide detailed interpretations.

SIB200 For LaBSE, correlation values are consistently strong and statistically significant across all layers. The first (Pearson $r = 0.323$), middle (Pearson $r = 0.309$), and last (Pearson $r = 0.210$) layers all demonstrate meaningful positive correlations with performance ($p \approx 0$), indicating that cosine similarity is well-aligned with task accuracy throughout the network. This suggests that SIB200 benefits from LaBSE's cross-lingual representations, especially in the earlier and middle layers. In contrast, Qwen2.5-0.5B shows very weak but statistically significant correlations ($r \leq 0.12$ across all layers). While the trends are consistent, the effect sizes are negligible, suggesting that cosine similarity has limited practical influence on performance for Qwen2.5-0.5B on this dataset.

INCLUDE-BASE For LaBSE, correlations between cosine similarity and performance are negligible and statistically non-significant across all layers, with Pearson r values close to zero (-0.041, 0.005, -0.021). This suggests no meaningful alignment between representational similarity and task accuracy. In contrast, Qwen2.5-0.5B exhibits weak but statistically significant positive correlations (Pearson r range: 0.14–0.18), indicating that higher cosine similarity is marginally associated with improved performance. Despite the small effect sizes, these results highlight a slight but consistent behavioural alignment in Qwen2.5-0.5B on this dataset.

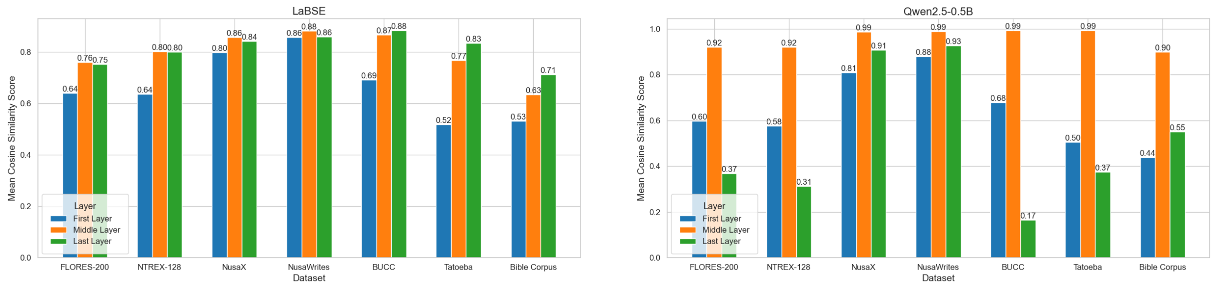
XCOPA For LaBSE, correlation values across layers are weak and statistically insignificant, suggesting minimal alignment between representational similarity and model performance. In contrast, Qwen2.5-0.5B exhibits a strong and statistically significant positive correlation in the last layer (Pearson $r = 0.538$, $p < 0.001$), implying that deeper representations may be more predictive for XCOPA.

PAWS-X LaBSE shows weak, non-significant positive correlations across layers. However, Qwen2.5-0.5B demonstrates a strong positive correlation in the middle layer (Pearson $r = 0.532$, $p \approx 0.004$), suggesting that intermediate representations capture more alignment-relevant features for paraphrase detection.

Downstream Performance Relative to Random Baselines To provide a clearer picture of cross-lingual generalization and behavior alignment, we present a set of bar charts

Dataset	Train	Test	Total	# Languages
SIB200	143,705	41,820	185,525	205
INCLUDE-BASE	890	22,638	23,528	44
XCOPA	1,100	5,500	6,600	11
PAWS-X	345,807	14,000	359,807	7

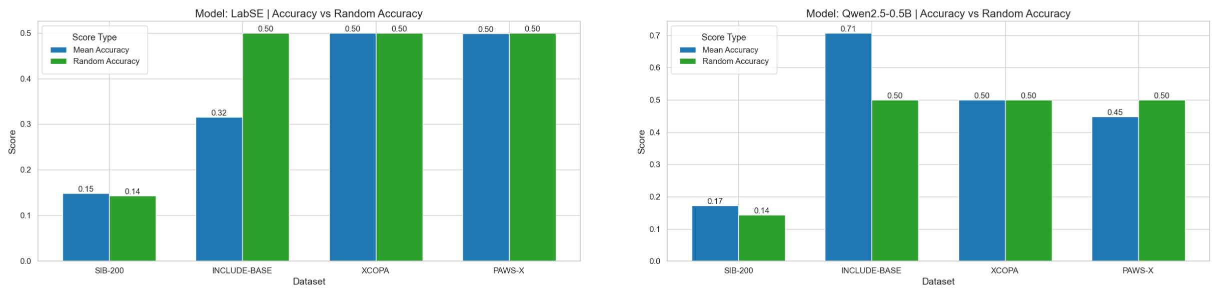
Table 7: Dataset sizes and number of languages for downstream tasks.



(a) Mean Cosine Similarity Score on LaBSE Model

(b) Mean Cosine Similarity Score on Qwen2.5-0.5B Model

Figure 4: Layer-wise cosine similarity distributions of LaBSE and Qwen2.5-0.5B models across different datasets.



(a) Performance of LaBSE across downstream tasks compared to random baselines.

(b) Performance of Qwen2.5-0.5B across downstream tasks compared to random baselines.

Figure 5: Comparison of LaBSE and Qwen2.5-0.5B performance across various downstream tasks and their corresponding random baselines.

Dataset	Total	# Languages
FLORES-200	1,012	204
NTREX-128	1,997	128
NusaX	400	12
NusaWrites	14,800	9 (language pairs)
BUCC	35,000	4 (language pairs)
Tatoeba	88,877	112 (language pairs)
BibleCorpus	85,533	828 (language pairs)

Table 8: Total example counts and number of languages for alignment tasks. We only use test set for this alignment task.

Dataset	Train	Test	Total	# Languages
FLORES-200	997	1012	2,009	204
NTREX-128	-	1,997	1,997	128
NusaX	500	400	400	12

Table 9: Total example counts per language and number of languages for for LID tasks.

Dataset	Train	Test	Total	# Languages
FLORES-200	997	1012	2,009	204
Dolly	-	1,800	-	9

Table 10: Total example counts per language and number of languages for Language Control.

Setting	Total	# Languages
Monolingual		
Aya	100	5
Dolly	100	5
Okapi	100	10
Native prompts	100	4
Crosslingual		
Okapi	100	14
shareGPT	100	14
Complex prompts	99	14

Table 11: Total example counts per language and number of languages for Language Confusion tasks, taken from Language Confusion Benchmark. Only test set is available.

comparing the performance of LaBSE and Qwen2.5-0.5B across four downstream evaluation datasets—SIB200, INCLUDE-BASE, XCOPA, and PAWS-X—relative to their respective random baselines.

On XCOPA and PAWS-X, LaBSE yields near-random or below-random performance, indicating that its fixed representations struggle with cross-lingual commonsense reasoning and paraphrase detection. For SIB200, LaBSE performs slightly above the random baseline, suggesting limited task sensitivity in multilingual sentence similarity settings. However, its performance on INCLUDE-BASE remains weak, staying near or below the random baseline and highlighting deficiencies in broader multilingual alignment.

In contrast, Qwen2.5-0.5B demonstrates stronger generalization on both SIB200 and INCLUDE-BASE, significantly outperforming its baseline and showing evidence of better cross-lingual task adaptation. However, it faces challenges on XCOPA and PAWS-X, where its performance hovers around or falls below baseline, pointing to possible limitations in zero-shot commonsense reasoning and paraphrase understanding across languages.

These comparisons highlight the differing strengths and weaknesses of encoder-only and decoder-only multilingual models across select zero-shot evaluation tasks. See Figure 5.

B.3 Additional Analysis For Alignment and LID Correlation

As shown in Table 13, the correlation between alignment (as measured by cosine similarity) and downstream LID performance varies notably across datasets, model architectures, and transformer layers. The following sections provide detailed interpretations for each dataset to contextualize these trends.

FLORES-200 On the FLORES-200 dataset, we observe a moderate negative correlation between cosine similarity and LID performance for both LaBSE and Qwen2.5-0.5B. The strength of the correlation increases in deeper layers, with the last layer showing the strongest correlation ($r = -0.707$, $p < 10^{-31}$) for LaBSE. Qwen2.5-0.5B, however, exhibits its strongest negative correlation in the middle layer ($r = -0.432$, $p < 10^{-9}$), indicating that as the embeddings become more aligned (i.e., higher cosine similarity), the language identity signal tends to weaken, potentially due to semantic abstraction. The statistically significant p-values across all layers confirm the robustness of this relationship. These findings reinforce the idea that high alignment may come at the cost of LID separability, especially in final layers for LaBSE and middle layer for Qwen2.5-0.5B, where representations are more semantically homogenized.

NTREX-128 For NTREX-128, the correlation trends diverge between the two models. LaBSE

Dataset	Model	Layer	Pearson r	R^2	p-value
SIB200	LaBSE	First	0.323	0.104	$< 10^{-300}$
		Middle	0.309	0.096	$< 10^{-300}$
		Last	0.210	0.044	$< 10^{-205}$
	Qwen2.5-0.5B	First	0.060	0.004	$< 10^{-17}$
		Middle	0.123	0.015	$< 10^{-69}$
		Last	0.043	0.002	$< 10^{-9}$
INCLUDE-BASE	LaBSE	First	-0.041	0.002	0.233
		Middle	0.005	0.000	0.884
		Last	-0.021	0.000	0.545
	Qwen2.5-0.5B	First	0.183	0.034	$< 10^{-7}$
		Middle	0.142	0.020	$< 10^{-4}$
		Last	0.168	0.028	$< 10^{-6}$
XCOPA	LaBSE	First	-0.115	0.013	0.458
		Middle	-0.026	0.001	0.867
		Last	0.144	0.021	0.352
	Qwen2.5-0.5B	First	0.292	0.085	0.055
		Middle	-0.139	0.019	0.368
		Last	0.538	0.289	< 0.001
PAWS-X	LaBSE	First	0.141	0.020	0.484
		Middle	0.270	0.073	0.173
		Last	0.146	0.021	0.467
	Qwen2.5-0.5B	First	0.228	0.052	0.252
		Middle	0.532	0.283	0.004
		Last	0.369	0.136	0.059

Table 12: Pearson correlation coefficients (r), R^2, and p-values for the relationship between cosine similarity and task performance across different transformer layers on LaBSE and Qwen2.5-0.5B. Dashes (–) indicate missing values due to unavailable data.

Dataset	Model	Layer	Pearson r	R^2	p-value
FLORES-200	LaBSE	First	0.024	0.001	0.732
		Middle	-0.122	0.015	0.084
		Last	-0.707	0.500	$< 10^{-31}$
	Qwen2.5-0.5B	First	-0.142	0.020	0.043
		Middle	-0.432	0.186	$< 10^{-9}$
		Last	-0.278	0.077	$< 10^{-4}$
NTREX-128	LaBSE	First	0.254	0.065	0.012
		Middle	-0.173	0.030	0.089
		Last	-0.621	0.385	$< 10^{-11}$
	Qwen2.5-0.5B	First	-0.232	0.054	0.021
		Middle	-0.476	0.226	$< 10^{-6}$
		Last	-0.340	0.115	0.001
NusaX	LaBSE	First	-0.566	0.320	0.112
		Middle	-0.872	0.760	0.002
		Last	–	–	–
	Qwen2.5-0.5B	First	-0.455	0.207	0.218
		Middle	-0.873	0.763	0.002
		Last	-0.045	0.002	0.910

Table 13: Pearson correlation coefficients (r), R^2, and p-values for the relationship between KNN LID F1 score using mean-pooled embedding and alignment cosine similarity across different transformer layers on LaBSE and Qwen2.5-0.5B.

exhibits its strongest negative correlation in the the last layer (Pearson $r = -0.621$, $p < 10^{-11}$), with a positive correlation in the first layer (Pearson $r = 0.254$, $p = 0.012$) and weak negative correlation in the middle (Pearson $r = -0.173$, $p = 0.089$). This suggests that early representations in LaBSE may still retain relatively distinct language features that diminish with depth. In contrast, Qwen2.5-0.5B shows more consistent negative correlations across all layers, particularly in the middle layer (Pearson $r = -0.476$, $p < 10^{-6}$). These results highlight a more uniform degradation of LID-relevant information in Qwen's architecture compared to LaBSE.

NusaX For NusaX, alignment-LID correlations exhibit distinct patterns. LaBSE shows a weak correlation in the first layer (Pearson $r = -0.566$, $p = 0.112$), a highly negative correlation in the middle layer (Pearson $r = -0.872$, $p = 0.002$), and no measurable correlation in the last layer (–), which we assume reflects a perfect inverse relationship (Pearson $r \approx -1$) due to complete LID failure. Qwen2.5-0.5B follows a similar pattern, with its most negative correlation in the middle layer (Pearson $r = -0.873$, $p = 0.002$) and negligible correlations in the first (Pearson $r = -0.455$, $p = 0.218$) and last layers (Pearson $r = -0.045$, $p = 0.910$). The correlations for both models are the most negative observed across all datasets, suggesting alignment disproportionately degrades language signals in low-resource settings. This extreme inverse relationship likely stems from the models' lack of prior exposure to NusaX languages during training, limiting their ability to retain language identity in aligned embeddings.

C LID Methods and Results

D Language Control Results

D.1 Generation Hyperparameter

The generation process for the language control and language confusion results uses specific hyperparameter to balance creativity and control. We set `max_new_tokens=50` for language control and `max_new_tokens=256` for language confusion, and set `top_k` to 50. We apply nucleus sampling with `top_p=0.9`, and use a moderate temperature of `0.7` to encourage focused yet varied outputs. To reduce repetitive phrases, we apply a `repetition_penalty` of 1.5. For input preparation, we follow the formatting conventions and parameters used by Qwen2.5-0.5 models.

D.2 Language Vector Setting

Linear Discriminant Analysis (LDA) is utilized to construct language vectors by extracting language-specific features from the Qwen2.5-0.5B model's scaled hidden states, optimizing crosslingual control through class separability. We evaluate various component sizes (20, 40, 50, 100, 150, 203) to balance LID accuracy and unused variance, fitting an LDA model and training a linear neural network (with 10 epochs, Adam optimizer, and CrossEntropyLoss) to achieve a peak accuracy of approximately 90.63% at 100 components. The unused variance is minimized, ensuring retained discriminative information for injection (δ) with pruning, which enhances language targeting while the Figure 6 visually confirms this optimal trade-off.

D.3 Language Correctness on Different Shift Strategies

Comparing language correctness of Base and Instruct respectively (Table 15 & Figure 3) reveals that the Qwen2.5-0.5B-Instruct model significantly enhances cross-lingual language control. It achieves 100% LID correctness with the Seq + Gen shift strategy in EN→XX direction, compared to the Base model's gen-only average of 87.6%. It suggests the impact of instruct model and our `ITLC` method in improving language separability and semantic transfer, as supported by prior human evaluations Table 5. Both models excel in XX→EN direction (Instruct: 96.7%, Base: 96.0%),

Model	Method	Layer	FLORES-200		NTREX-128		NusaX	
			CLS	Mean	CLS	Mean	CLS	Mean
LaBSE	KNN	First	80.65	88.35	87.02	90.43	64.12	81.78
		Middle	65.11	78.85	71.37	81.30	33.89	45.37
		Last	7.65	3.92	3.45	1.63	0.54	0.00
	Linear Probing	First	93.47	95.13	92.21	93.29	89.16	97.30
		Middle	92.99	94.18	92.33	92.68	88.00	94.51
		Last	30.03	70.89	22.91	74.36	56.00	65.44
Qwen2.5-0.5B	KNN	First	–	83.69	–	86.06	–	65.79
		Middle	–	55.32	–	54.73	–	25.05
		Last	–	71.73	–	81.86	–	29.39
	Linear Probing	First	–	94.21	–	91.42	–	95.55
		Middle	–	91.76	–	90.04	–	87.09
		Last	–	92.46	–	90.27	–	88.77

Table 14: F1 score for KNN and linear classifiers by layer and pooling on FLORES-200, NTREX-128, and NusaX.

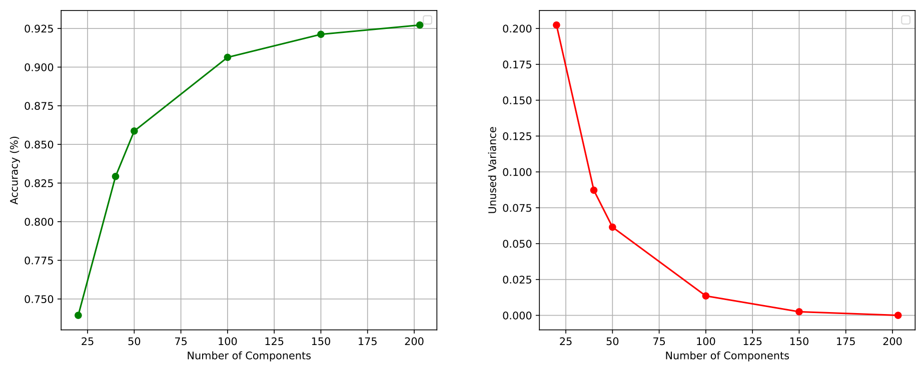


Figure 6: Controlling the number of language feature representations by using LDA performance accuracy (**Left**) and unused variance (**Right**) across number of components.

Language	prompt-only		gen-only		prompt-and-gen	
	EN→XX	XX→EN	EN→XX	XX→EN	EN→XX	XX→EN
ID	90.5	99.0	87.5	100.0	100.0	95.0
TH	99.0	100.0	100.0	100.0	100.0	100.0
TR	76.0	99.0	97.5	100.0	100.0	89.0
JA	85.5	100.0	100.0	100.0	100.0	98.5
FR	71.0	100.0	91.5	100.0	100.0	95.0
ES	88.5	100.0	88.0	100.0	100.0	100.0
ARB	92.5	100.0	91.0	100.0	100.0	100.0
KO	81.0	97.5	86.5	97.0	99.5	91.5
ZH	64.5	99.0	68.5	100.0	100.0	95.0

Table 15: Language correctness (%) for Qwen2.5-0.5B-Instruct across EN→XX and XX→EN Directions.

reflecting English's training dominance (Table 4), though linguistic overlaps (e.g., Korean with Chinese) and weaker EN→XX direction performance for low-resource languages like Chinese (Instruct: 68.5% Generated Only) suggest that Seq + Gen is optimal for Instruct.

D.4 Cross-lingual Generation Performance

As shown in Tables 16 and 17, the Instruct model consistently outperforms the Base model in both EN→XX and XX→EN direction, particularly in semantic relevance (e.g., ROUGE for Indonesian **prompt-and-gen**: 13.6 vs. 14.1 and SacreBLEU: 16.89 vs. 12.20). Despite lower or even zero SacreBLEU in some EN→XX direction cases (e.g., Thai and Korean), the Instruct model shows improved performance in low-resource directions for XX→EN direction, indicating better semantic transfer. This aligns with human evaluations (Table 5) and confirms the effectiveness of our LDA-based injection method in enabling cross-lingual generation that maintains both flexibility and semantic fidelity, as further supported by upperbound comparisons (Table 4).

D.5 Additional Examples of Cross-lingual Generation

Figure 7 and Figure 8 present several examples of generated outputs across multiple source languages (fr, tr, zh, ja, ar) targeting English. Overall, our `ITLC` method successfully shifts to the desired target language and demonstrates effective cross-lingual generation. For instance, in the Japanese

Language	Base									Instruct								
	prompt-only			gen-only			prompt-and-gen			prompt-only			gen-only			prompt-and-gen		
	BS	**Rog**	**SB**	**BS**	**Rog**	**SB**	**BS**	**Rog**	**SB**	**BS**	**Rog**	**SB**	**BS**	**Rog**	**SB**	**BS**	**Rog**	**SB**
ID	51.8	0.9	2.36	57.6	1.3	0.00	53.7	0.3	0.00	63.6	**10.4**	**14.30**	**63.9**	1.2	0.00	63.7	0.9	0.00
TH	58.5	4.9	3.20	52.6	5.6	3.46	58.7	7.6	3.75	**64.1**	1.5	**15.97**	58.0	7.0	**15.97**	58.4	**8.7**	4.93
TR	47.7	1.0	0.00	54.4	3.1	0.00	47.2	2.0	8.12	**60.2**	**7.9**	1.43	58.9	3.0	**9.65**	57.3	4.0	0.00
JA	60.1	9.1	2.22	55.9	9.7	7.05	59.3	**10.8**	7.33	**61.5**	4.4	**15.97**	59.3	9.7	4.82	58.0	8.6	5.37
FR	51.8	9.5	9.54	54.8	11.2	5.86	47.7	10.4	4.62	**63.2**	11.1	10.97	60.1	**12.7**	4.10	59.8	**12.7**	2.74
ES	39.3	0.7	1.41	52.5	0.9	0.00	40.2	0.4	4.97	64.4	**16.0**	7.17	63.6	0.4	**11.88**	**65.5**	0.2	6.57
AR	59.8	0.8	1.45	52.1	1.3	0.00	56.5	1.4	4.20	**63.5**	0.8	**4.75**	62.0	**3.0**	4.11	55.1	0.8	2.63
KO	53.4	0.8	0.00	58.5	1.2	0.00	53.3	0.8	**10.68**	62.0	**4.1**	3.58	**63.8**	1.66	0.00	59.3	0.1	0.00
ZH	53.0	4.9	2.36	57.1	7.7	6.02	55.7	8.2	**10.60**	62.3	1.3	0.00	**63.3**	9.3	6.90	63.1	**10.5**	5.37

Table 16: Comparison of Generative Performance for Base & Instruct in EN→XX Direction. **BS** denote as BertScore (F1), **Rog** is ROUGE-1, and **SB** is for SacreBLEU.

Language	Base									Instruct								
	prompt-only			gen-only			prompt-and-gen			prompt-only			gen-only			prompt-and-gen		
	BS	**Rog**	**SB**	**BS**	**Rog**	**SB**	**BS**	**Rog**	**SB**	**BS**	**Rog**	**SB**	**BS**	**Rog**	**SB**	**BS**	**Rog**	**SB**
ID	59.5	14.1	7.93	59.8	13.8	9.31	59.9	14.1	12.20	62.5	13.9	4.76	**63.9**	**15.7**	7.97	62.2	13.6	**16.89**
TH	62.1	13.7	6.19	60.6	14.3	11.18	62.2	14.4	**14.40**	62.0	13.8	4.93	**62.3**	**15.4**	5.59	62.1	13.8	11.12
TR	53.2	11.1	**14.51**	60.6	13.2	12.67	55.2	11.8	14.18	62.0	12.8	8.45	**62.5**	**14.3**	13.61	61.8	13.1	8.70
JA	62.1	14.1	**15.97**	57.9	14.3	9.84	61.5	15.2	11.91	62.7	14.9	5.73	**63.2**	**17.0**	6.89	62.4	14.7	12.17
FR	61.8	16.7	8.83	58.5	14.7	**9.73**	60.6	15.6	8.66	64.2	17.1	5.68	**64.9**	**18.0**	8.64	64.5	17.2	6.08
ES	63.2	16.8	10.89	60.0	15.4	6.14	63.5	16.7	**11.38**	64.9	18.1	4.38	**65.4**	**18.5**	5.99	64.8	17.6	7.17
AR	62.4	15.0	9.32	56.9	13.3	14.60	**63.1**	15.1	**19.28**	63.3	14.4	6.61	**63.1**	15.3	10.43	62.3	13.7	6.22
ZH	59.3	13.3	15.32	61.1	16.7	11.83	60.7	**16.8**	9.84	**63.2**	16.3	6.73	61.5	14.4	6.27	61.0	13.0	10.96
KO	56.4	13.3	15.32	56.9	13.5	6.43	56.6	13.2	**16.44**	**62.9**	**14.9**	9.59	62.0	14.5	4.89	61.2	13.9	6.44

Table 17: Comparison of Generative Performance for Base & Instruct in XX→EN Direction. **BS** denote as BertScore (F1), **Rog** is ROUGE-1, and **SB** is for SacreBLEU.

Target Language	Monolingual		ITLC	
	SacreBLEU	**BertScore** (F1)	**SacreBLEU**	**BertScore** (F1)
ID	4.58	**60.4**	**10.6**	57.1
TH	0.0	**60.4**	**1.45**	57.6
TR	**8.47**	57.7	3.75	**58.7**
JA	0.0	**57.5**	**8.12**	54.4
FR	**8.61**	58.8	7.33	**60.1**
ES	**12.28**	**60.3**	9.54	54.8
AR	4.90	**57.0**	**4.97**	52.5
ZH	0.0	**59.2**	**10.68**	58.5
KO	**9.55**	57.2	4.2	**59.8**

Table 18: Generation performance for different target languages with Qwen2.5-0.5B. **Mono Baseline** denotes the model prompted in the same language as the desired target language.

Scaling	Monolingual			Crosslingual		
	LCPR	LPR	WPR	LCPR	LPR	WPR
prompt-0.1	64.86	81.01	65.67	33.97	23.75	74.74
prompt-0.2	**66.39**	82.14	**66.75**	38.88	28.91	**75.37**
prompt-0.3	65.59	**82.86**	65.78	46.03	37.86	72.56
prompt-0.4	65.45	82.79	65.53	57.20	51.97	72.27
prompt-0.5	65.87	82.73	62.50	62.93	61.63	73.43
prompt-0.6	64.92	82.64	65.24	63.91	63.83	73.20
prompt-0.7	64.78	81.03	65.52	64.63	66.09	71.74
prompt-0.8	63.69	80.40	65.28	**65.71**	**66.41**	74.24
prompt-0.9	61.25	75.81	64.15	64.59	64.79	73.30
prompt-1.0	60.39	74.98	63.87	62.97	63.35	72.79

Table 19: Performance (LCPR / LPR / WPR) of the Qwen2.5-0.5B model under the prompt-only setting with base shift vector, evaluated across different language vector scaling factors.

example in Figure 7, the input prompt is "Help me come up with three new business ideas." The model's response with "I have already developed some ideas..." (translate with Google Translate) —shows that it semantically understands the question, although the correctness of the content remains somewhat limited.

Another example, such as in the EN→ZH direction, shows that the model generates a well-formed response in Simplified (Hans) Chinese. The output produces: "Dear Mom and Dad, Hello everyone! My dear mother, I am from... I am a member of the research and development..."—demonstrating clear semantic alignment with the prompt. However, as with the previous case, the accuracy and relevance of the content could still be limited.

E Language Confusion Result

Ablation Study of Scaling for Different Language Vector Injection Strategies As shown in Table 19, Table 20 and Table 21 Our analysis reveals distinct optimal scaling factors for crosslingual LCPR across injection strategies: prompt-only achieves peak performance at scaling 0.8 (65.71), gen-only at 0.6 (71.35), and prompt-and-gen at 0.5 (78.93). Notably, prompt-and-gen outperforms other strategies, suggesting combined injection better preserves crosslingual alignment. However, ap-

Lang Shift	Prompt	Generated
FR → EN	Suggérez-moi 5 films d'horreur	Sure! Here I go with five horror movies: 1) "The Conjuring" - This film is famous for its eerie atmosphere and supernatural elements. 2) "Hunger Games: The First Years." It also has strong effects of fear on people ...
TR → EN	Bana ev yapımı biramı daha ilginç hale getirmenin beş yolunu söyle.	A new way to break the cycle of poverty in our society is key for people who can pay more than 10 dollars each year on an average per person –one that could bring those individuals down between $5 and over five times their own cost ...
ZH → EN	我可以送些什么东西给一个好奇的孩子?	It is important to consider what we do and not give things that could hurt or harm others. Let us be mindful of the choices around them before giving anything else as well because ultimately who will know if you truly deserve being treated with care in this case ...
JA → EN	新しいビジネスのアイデアを3つ思いつくのを手伝って下さい	I have already developed some ideas for new businesses that can take place in the 21st century. To be clear to me what exactly you need an idea from this time on? How many times we will use them? What kind of business ...
AR → EN	شرح كيفية عمل ساعة آلية	The "hour clock" in the game is to be placed on board for each player during their turn when they have completed an action against that particular character (e.g., at end of season). A complete stage will then include one 1/3 ...

Figure 7: Examples of generated outputs from Qwen2.5-0.5B-Instruct with injection in XX→EN.

Scaling	Monolingual			Crosslingual		
	LCPR	LPR	WPR	LCPR	LPR	WPR
gen-0.1	64.75	83.99	63.85	35.07	24.79	74.92
gen-0.2	**65.35**	85.09	**65.01**	39.93	28.96	**75.92**
gen-0.3	62.61	86.55	59.29	48.08	38.97	71.16
gen-0.4	59.61	86.23	54.95	57.49	57.82	64.37
gen-0.5	59.64	86.54	54.85	62.62	65.94	65.48
gen-0.6	60.05	**87.49**	58.14	**71.35**	80.46	67.67
gen-0.7	58.01	87.41	55.72	69.39	**80.73**	66.57
gen-0.8	52.45	82.78	52.35	65.84	75.74	65.93
gen-0.9	47.07	75.83	50.58	58.61	68.51	63.73
gen-1.0	40.44	71.15	54.91	51.25	61.85	61.83

Table 20: Performance (LCPR / LPR / WPR) of the Qwen2.5-0.5B model under the generated-only setting with base shift vector, evaluated across different language vector scaling factors.

Scaling	Monolingual			Crosslingual		
	LCPR	LPR	WPR	LCPR	LPR	WPR
prompt-and-gen-0.1	**64.21**	84.27	**63.77**	39.48	28.69	75.74
prompt-and-gen-0.2	63.25	86.34	61.76	50.04	41.18	75.07
prompt-and-gen-0.3	62.94	**88.24**	60.85	64.22	64.18	72.53
prompt-and-gen-0.4	60.79	88.06	59.09	75.88	80.58	75.78
prompt-and-gen-0.5	59.98	87.11	59.41	**78.93**	**85.08**	**77.15**
prompt-and-gen-0.6	57.01	86.37	55.90	77.21	84.13	74.90
prompt-and-gen-0.7	53.56	82.91	53.63	72.57	81.98	71.51
prompt-and-gen-0.8	49.00	77.27	51.33	68.22	76.80	70.08
prompt-and-gen-0.9	40.41	70.51	48.16	60.97	69.07	66.44
prompt-and-gen-1.0	36.60	70.01	51.30	52.51	61.07	63.82

Table 21: Performance (LCPR / LPR / WPR) of the Qwen2.5-0.5B model under the prompt-and-generated setting with base shift vector, evaluated across different language vector scaling factors.

plying language vector shifts to monolingual[4] tasks degrades performance as scaling increases. This suggests that applying shift vectors to monolingual inputs does not amplify language-specific features but instead displaces the original distribution, degrading the performance.

[4]In monolingual (source = target) settings, $\delta = \mathbf{v}_y^{\text{orig}}$ is applied via $\mathbf{h}' = \mathbf{h} + \alpha\delta$.

Crosslingual Language Control via Language Shift Vectors As shown in Table 24 and Table 25, our method ITLC substantially improves cross-lingual language control, as evidenced by enhanced LCPR scores when applying language shift vectors. In crosslingual settings, the seq-and-gen strategy achieves peak performance, with base

Lang Shift	Prompt	Generated
EN → FR	Imagine you have to spend a week on a deserted island. What are 5 items you'd bring with you and why?	Eh bien, Je vais leave comission de monself Comme I não j'ai pas un grand capital à risker je me suis put sortir sans débris que vous couldais take: 1) Alcibiade: Ache ...
EN → TR	Describe the sensation of taking a cold shower and explain the rationale for why someone might take one.	Sırbistan'nde bir aylağda çok güzel su (çocuk) her sabrabındaki kitlemlerde en az 15min de döşmezken bu saymaği sonkardan da ...
EN → ZH	Write a letter to my mum and dad detailing the reasons why they should buy me a pet tortoise	尊敬的妈妈和爸爸，大家好！我亲爱的母亲，我是来自台湾花东海岸南太平洋海域采种选育、研发团队之成员，在1958年3月～4日于台北市选举举行第2 ...
EN → JA	Write a short paragraph on the impact of nutrition on human health.	健康的体重管理を可能にするには、合理的に摺入（食事が量調解）方法は：** "水分充満" と、"筋トレ" 的「効率の良いタンパク質チップレモンフィ ...
EN → AR	Explain how a mechanical watch works	بسم الله وشاعر: "لعلمنا منى""أنت على علم أنوائل : "-" أخبرتنا لا كود": زوعI", قائم السعودة ...

Figure 8: Examples of generated outputs from Qwen2.5-0.5B-Instruct with injection in EN→XX.

shift vectors attaining 78.93% LCPR for the base model and 81.51% for the instruct model. While gen-only and seq-only strategies demonstrate moderate gains of 71.35% and 65.71% respectively for the base model, and 75.56% and 76.05% for the instruct model, they are consistently outperformed by the seq-and-gen approach. Notably, base shift vectors achieve marginally higher LCPR compared to their instruct counterparts across both models, with 78.93% versus 76.06% LCPR for base model configurations and 81.51% versus 80.96% for the instruct model. This consistent advantage suggests that base vectors retain more language-specific information critical for cross-lingual adaptation, likely attributable to their training objectives emphasizing multilingual representation rather than task-specific alignment. A detailed breakdown of the LCPR scores per language for both the base and instruct models is presented in Table 22 adn Table 23

Impact of Few-Shot Prompting on Monolingual and Crosslingual Performance As shown in Table 13 and Table 14, for the base model, monolingual settings exhibit performance degradation with increasing few-shot examples, as LCPR drops from 65.27% to 54.47%. This decline likely arises from the inclusion of multilingual few-shot examples[5], creating conflicting linguistic signals. In cross-lingual settings, LCPR improves progressively from 29.41% to 56.78%, as few-shot examples utilize English inputs with explicit target-language directives[6], reinforcing input-output alignment. This indicates that the model demonstrates stronger cross-lingual adaptation with English-centric prompting. The instruct model exhibits minimal variation across few-shot configurations, with monolingual LCPR ranging between 74.52% and 75.59%, and crosslingual between 63.00% and 64.56%, suggesting its instruction-tuning enables robust multilingual prompting without dependency on few-shot examples. This stability implies that the instruct model's training on aligned multilingual inputs maximizes its crosslingual capability a priori, rendering few-shot augmentation redundant.

Chat/QA Template Efficacy Across Settings As shown in Table 24 and Table 25, structured templates[7] exhibit divergent impacts on monolingual and crosslingual performance. For the base Qwen2.5-0.5B, introducing a QA template (0-shot) degrades monolingual LCPR from 65.27% to 59.26% but improves crosslingual performance from 29.41% to 44.68%, suggesting that task-specific formatting disrupts monolingual focus while aiding cross-lingual alignment. Con-

[5]Few-shot examples are drawn directly from the original benchmark implementation, which includes languages distinct from the target language.

[6]Cross-lingual prompts follow the benchmark's original structure: English inputs with instructions like Respond in <TARGET_LANG>.

[7]QA template: `Q: A:` format; chat template: model-specific structure from instruction-tuning.

	AVG	AR	DE	EN	ES	FR	HI	ID	IT	JA	KO	PT	RU	TR	VI	ZH
Monolingual																
Qwen2.5-0.5B	**65.27**	92.68	63.57	45.29	70.99	59.46	2.02	68.70	60.71	81.21	**85.55**	55.04	**94.27**	72.47	37.96	89.11
+ Q/A template (0-shot)	59.26	59.56	**71.46**	50.98	**74.57**	62.64	0.00	65.58	63.10	46.40	69.64	**61.47**	76.76	42.58	58.63	85.57
+ 1-shot	56.12	72.37	65.25	52.24	60.05	41.22	2.04	67.18	67.88	36.74	78.44	49.31	80.84	50.66	60.78	56.79
+ 2-shot	51.59	66.87	57.53	53.90	64.04	50.61	0.00	69.51	57.44	35.07	42.73	45.60	66.81	48.13	50.90	64.74
+ 3-shot	52.52	65.04	50.91	**55.18**	73.01	62.51	6.00	70.36	53.43	29.06	61.98	48.50	54.37	57.60	44.07	55.76
+ 4-shot	54.16	66.91	50.36	53.80	68.50	**63.79**	6.32	62.89	**72.09**	39.24	59.63	49.35	78.60	34.68	59.03	47.25
+ 5-shot	54.47	68.36	67.17	49.14	71.24	62.67	7.89	69.72	60.08	31.92	53.69	48.92	78.94	37.62	51.87	57.79
+ ITLC (apply base shift vector)																
+ prompt-only (0.8)	63.69	94.24	67.06	32.69	70.51	47.82	0.00	**82.15**	59.35	78.86	71.04	55.92	89.26	71.99	46.62	87.82
+ gen-only (0.6)	60.05	93.86	59.82	6.95	47.89	23.20	**15.24**	74.16	49.98	**88.18**	73.91	35.17	88.42	**82.46**	67.74	93.74
+ prompt-and-gen (0.5)	59.98	94.70	59.59	7.82	55.13	26.13	4.04	74.82	43.01	85.28	83.74	34.48	94.24	80.98	60.93	**94.85**
+ ITLC (apply instruct shift vector)																
+ prompt-only (0.8)	63.11	93.82	59.95	38.97	70.31	48.53	0.00	72.45	63.44	85.45	76.42	52.98	87.61	65.56	44.49	86.68
+ gen-only (0.6)	55.89	**95.67**	57.57	9.76	38.31	18.47	4.04	72.47	43.50	85.43	76.10	27.18	90.30	75.99	57.39	86.19
+ prompt-and-gen (0.5)	58.48	94.94	58.88	3.11	43.66	29.49	6.00	71.94	56.36	85.67	80.18	25.30	89.33	76.51	**67.98**	87.82
Crosslingual																
	AVG	AR	DE	EN	ES	FR	HI	ID	IT	JA	KO	PT	RU	TR	VI	ZH
Qwen2.5-0.5B	29.41	29.98	36.11	–	37.52	35.01	5.48	37.29	34.14	12.45	10.36	32.04	42.23	37.63	33.72	27.75
+ Q/A template (0-shot)	44.68	47.08	49.89	–	58.09	59.10	5.88	57.08	50.16	24.36	17.90	48.78	62.13	48.29	46.28	50.48
+ 1-shot	47.42	43.69	52.73	–	56.13	58.55	10.13	62.77	57.21	25.30	37.61	48.29	66.68	54.92	46.57	43.33
+ 2-shot	49.36	50.88	53.62	–	61.12	63.58	8.93	67.67	60.27	27.93	40.40	52.86	65.56	58.32	48.10	31.48
+ 3-shot	53.16	63.00	56.57	–	62.67	68.08	7.84	65.21	65.78	28.84	38.33	54.44	71.05	65.83	54.10	42.52
+ 4-shot	55.03	61.82	52.35	–	64.14	64.13	12.06	71.80	65.13	30.72	43.88	61.73	77.83	64.57	57.66	42.55
+ 5-shot	56.78	67.70	57.20	–	63.01	62.19	21.43	71.42	67.88	37.83	44.35	57.55	76.36	68.56	58.15	41.21
+ ITLC (apply base shift vector)																
+ prompt-only (0.8)	65.71	83.36	62.48	–	75.77	67.50	10.48	73.30	70.83	60.04	61.90	69.90	83.27	69.29	57.15	74.67
+ gen-only (0.6)	71.35	79.36	**82.60**	–	82.03	73.65	45.68	79.38	71.78	56.47	72.54	71.29	56.98	86.33	66.59	74.16
+ prompt-and-gen (0.5)	**78.93**	**96.23**	79.77	–	**86.94**	**76.30**	**50.05**	**81.14**	**75.33**	**62.18**	78.08	77.27	**90.44**	89.11	79.00	**83.15**
+ ITLC (apply instruct shift vector)																
+ prompt-only (0.8)	63.08	81.38	65.14	–	77.39	66.16	14.03	74.94	66.50	49.36	55.18	66.28	77.59	67.34	60.07	61.79
+ gen-only (0.6)	68.70	82.17	79.48	–	80.57	67.16	37.69	76.92	68.98	56.01	72.72	78.02	44.59	85.17	83.96	48.33
+ prompt-and-gen (0.5)	76.06	94.23	82.17	–	81.49	70.67	41.06	80.42	71.20	61.02	**80.39**	**79.89**	86.78	**89.33**	**84.36**	61.85

Table 22: Language Confusion Pass Rate (LCPR) of the base model across monolingual and crosslingual settings, with a detailed language-wise breakdown.

versely, the instruct model Qwen2.5-0.5B-Instruct maintains stable monolingual LCPR performance with its chat template from 74.79% to 74.52% , while achieving substantial crosslingual gains from 38.75% to 63.00%. This contrast underscores a critical trade-off: task-aligned templates enhance crosslingual consistency for both models but introduce monolingual interference in base architectures. The instruct model's robustness stems from its training on conversational formats, which harmonizes template usage with its intrinsic multilingual capabilities.

E.1 Measuring Language-Specific Information in LLMs

E.1.1 Methods

To investigate language-specific information in multilingual representations, we analyze two distinct paradigms: (1) frozen embeddings from pretrained decoder-only LLMs (Qwen-2.5) and (2) specialized multilingual sentence encoders (LaBSE). We evaluate whether linguistic identity is recoverable from their hidden states and how pooling strategies affect clusterability (via non-parametric KNN retrieval) and linear separability (via supervised classification heads).

KNN-based Language Identification We hypothesize that language identity manifests as separable clusters in the hidden space, which can be detected via non-parametric nearest-neighbor retrieval.

For both Qwen-2.5 and LaBSE, hidden states are extracted from the first ($\ell = 1$), middle ($\ell = m$), and final ($\ell = L$) layers. Let $\mathbf{H}^\ell \in \mathbb{R}^{T \times d}$ denote the hidden states at layer ℓ for a sequence of length T. Sentence-level embeddings are derived as follows:

- Qwen-2.5: Only mean pooling is applied:

$$\mathbf{e}^\ell_{\text{mean}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{H}^\ell_t \in \mathbb{R}^d.$$

- LaBSE: Both CLS and mean pooling are compared:

$$\mathbf{e}^\ell_{\text{CLS}} = \mathbf{H}^\ell_{[CLS]}, \quad \mathbf{e}^\ell_{\text{mean}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{H}^\ell_t \in \mathbb{R}^d.$$

For each layer $\ell \in \{1, m, L\}$ and pooling strategy $\text{pool} \in \{\text{mean}, \text{CLS}\}$, we construct reference

	Monolingual															
	AVG	AR	DE	EN	ES	FR	HI	ID	IT	JA	KO	PT	RU	TR	VI	ZH
Qwen2.5-0.5B-Instruct	74.79	93.58	73.98	**48.80**	85.14	77.26	0.00	**84.86**	68.60	76.09	85.08	81.02	**93.78**	80.58	**84.47**	88.63
+ Chat template (0-shot)	74.52	92.09	58.59	43.26	88.17	78.38	0.00	81.37	**84.41**	79.98	84.94	84.31	87.47	85.37	75.98	93.51
+ 1-shot	74.04	90.73	71.13	40.45	88.07	77.80	4.04	75.02	82.54	81.83	86.58	80.82	88.01	82.06	68.82	92.66
+ 2-shot	74.46	93.32	68.33	44.00	87.13	73.94	7.67	78.43	83.85	81.20	79.84	82.20	87.31	87.13	70.33	92.17
+ 3-shot	74.59	92.80	67.53	39.84	86.74	77.08	4.04	78.88	80.62	79.95	82.77	82.63	91.38	**89.91**	73.06	91.64
+ 4-shot	**75.59**	90.18	65.06	48.57	87.71	77.91	0.00	81.06	82.06	80.39	**89.78**	82.45	92.64	83.44	80.46	92.20
+ 5-shot	74.15	93.91	65.20	47.13	88.29	78.05	3.96	77.72	84.10	78.72	84.82	82.77	88.62	82.06	66.08	90.74
+ ITLC (apply base shift vector)																
+ prompt-only (0.8)	67.33	**95.38**	66.65	26.10	**90.89**	81.03	0.00	83.17	76.72	**85.44**	76.09	**86.10**	16.00	86.38	48.05	91.88
+ gen-only (0.6)	67.00	94.18	66.43	6.82	87.25	68.22	13.08	77.38	70.97	68.02	75.42	84.97	41.77	83.31	70.00	**97.22**
+ prompt-and-gen (0.5)	67.73	94.02	63.95	13.12	89.30	71.87	7.77	80.41	66.67	80.31	65.43	81.93	48.58	83.76	71.83	96.95
+ ITLC (apply instruct shift vector)																
+ prompt-only (0.8)	66.78	94.72	**78.69**	21.89	88.08	**81.47**	1.98	75.28	81.25	81.85	66.88	82.68	15.49	82.22	58.21	91.01
+ gen-only (0.6)	67.42	95.32	55.00	4.95	84.18	65.16	**24.28**	84.21	73.80	75.07	57.39	79.23	59.88	85.18	75.43	92.18
+ prompt-and-gen (0.5)	68.20	94.37	68.64	9.52	84.98	67.53	12.94	79.18	79.52	80.00	69.97	79.83	47.74	85.60	69.94	93.24

	Crosslingual															
	AVG	AR	DE	EN	ES	FR	HI	ID	IT	JA	KO	PT	RU	TR	VI	ZH
Qwen2.5-0.5B-Instruct	38.75	44.30	42.95	–	48.06	43.43	0.66	37.95	38.65	37.06	30.16	45.25	49.84	41.91	44.24	38.10
+ Chat template (0-shot)	63.00	74.74	61.66	–	77.35	70.44	5.86	67.16	70.20	58.01	53.94	72.04	**82.30**	67.27	64.49	56.57
+ 1-shot	63.95	75.31	66.12	–	80.44	72.65	5.25	70.23	70.03	57.57	53.42	74.46	81.45	70.14	61.64	56.62
+ 2-shot	64.56	75.83	67.02	–	78.82	75.76	2.68	67.62	70.97	56.57	54.69	74.87	79.45	71.55	66.39	61.59
+ 3-shot	64.25	74.91	64.20	–	79.27	73.69	4.57	70.46	70.71	60.90	52.55	74.44	78.99	70.51	64.15	60.15
+ 4-shot	63.80	78.38	62.61	–	77.70	71.23	6.51	65.33	68.04	60.90	53.70	74.73	79.07	70.74	63.96	60.28
+ 5-shot	63.53	75.76	63.36	–	75.19	75.44	2.61	69.80	69.20	56.86	51.94	74.81	79.62	68.07	70.17	56.56
+ ITLC (apply base shift vector)																
+ prompt-only (0.8)	76.05	93.60	71.59	–	90.78	82.49	12.44	80.52	83.92	**78.46**	76.41	84.95	67.93	82.36	72.52	86.68
+ gen-only (0.6)	75.56	92.44	82.65	–	86.31	80.93	72.07	72.22	83.15	54.31	75.74	81.85	31.36	87.29	73.29	84.22
+ prompt-and-gen (0.5)	**81.51**	94.49	84.06	–	**91.91**	**85.26**	63.23	**79.22**	84.94	55.01	**80.40**	86.46	80.19	85.92	**82.87**	87.25
+ ITLC (apply instruct shift vector)																
+ prompt-only (0.8)	73.26	91.28	74.62	–	87.32	76.08	6.49	82.48	83.50	71.94	77.84	83.83	54.01	78.22	74.54	83.48
+ gen-only (0.6)	73.95	92.41	84.15	–	82.26	77.59	64.81	71.76	83.04	62.62	73.90	85.09	23.87	84.18	72.57	77.07
+ prompt-and-gen (0.5)	80.96	**94.68**	**86.56**	–	88.33	82.43	**65.21**	74.37	**86.26**	64.68	78.02	**88.78**	65.62	**87.66**	81.74	**89.12**

Table 23: Language Confusion Pass Rate (LCPR) of the instruct model across monolingual and crosslingual settings, with a detailed language-wise breakdown.

Method	Monolingual			Crosslingual		
	LCPR	LPR	WPR	LCPR	LPR	WPR
Qwen2.5-0.5B	**65.27**	81.58	65.15	29.41	19.75	73.45
+ Q/A template (0-shot)	59.26	59.91	73.35	44.68	35.36	75.94
+ 1-shot	56.12	55.38	73.70	47.42	37.95	75.42
+ 2-shot	51.59	49.70	70.98	49.36	41.64	75.03
+ 3-shot	52.52	51.51	72.07	53.16	46.65	77.07
+ 4-shot	54.16	52.95	**74.15**	55.03	48.23	**77.60**
+ 5-shot	54.47	53.62	70.40	56.78	50.63	76.16
+ ITLC (apply base shift vector)						
+ prompt-only (0.8)	63.69	80.40	65.28	65.71	66.41	74.24
+ gen-only (0.6)	60.05	**87.49**	58.14	71.35	80.46	67.67
+ prompt-and-gen (0.5)	59.98	87.11	59.41	**78.93**	**85.08**	77.15
+ ITLC (apply instruct shift vector)						
+ prompt-only (0.8)	63.11	79.95	64.18	63.08	63.77	73.04
+ gen-only (0.6)	55.89	86.38	55.32	68.70	78.99	65.36
+ prompt-and-gen (0.5)	58.48	87.24	57.21	76.06	82.31	75.74

Table 24: Performance (LCPR / LPR / WPR) of base model under monolingual and crosslingual settings.

Method	Monolingual			Crosslingual		
	LCPR	LPR	WPR	LCPR	LPR	WPR
Qwen2.5-0.5B-Instruct	74.79	82.61	**77.94**	38.75	27.22	78.40
+ Chat template (0-shot)	74.52	83.66	77.12	63.00	57.69	79.50
+ 1-shot	74.04	82.75	76.52	63.95	59.24	79.50
+ 2-shot	74.46	83.47	74.07	64.56	59.86	78.76
+ 3-shot	74.59	84.11	76.27	64.25	59.74	78.45
+ 4-shot	**75.59**	**84.45**	77.52	63.80	58.89	79.52
+ 5-shot	74.15	82.87	76.37	63.53	58.79	75.34
+ ITLC (apply base shift vector)						
+ prompt-only (0.8)	67.33	74.82	76.35	76.05	77.68	**81.11**
+ gen-only (0.6)	67.00	84.07	65.83	75.56	82.42	74.51
+ prompt-and-gen (0.5)	67.73	81.70	68.96	**81.51**	85.32	80.55
+ ITLC (apply instruct shift vector)						
+ prompt-only (0.8)	66.78	74.96	73.08	73.26	76.37	79.20
+ gen-only (0.6)	67.42	83.64	65.46	73.95	84.06	71.40
+ prompt-and-gen (0.5)	68.20	82.20	68.05	80.96	**86.79**	78.84

Table 25: Performance (LCPR / LPR / WPR) of instruct model under monolingual and crosslingual settings.

sets:

$$\mathcal{R}^{\ell}_{\text{pool}} = \left\{ \left(\mathbf{e}^{\ell,(i,j)}_{\text{pool}}, y^{(j)} \right) \right\}_{i=1,j=1}^{200,204},$$

where $y^{(j)}$ is the language label for the j-th language in FLORES-200, and i indexes the examples within each language. This results in a total of $200 \times 204 = 40,800$ reference embeddings. For Qwen-2.5, only $\mathcal{R}^{\ell}_{\text{mean}}$ is used, while LaBSE employs both $\mathcal{R}^{\ell}_{\text{CLS}}$ and $\mathcal{R}^{\ell}_{\text{mean}}$.

We evaluate on three test sets: Flores-200, NTREX-128, and NusaX. To ensure fair comparison, we retain only languages overlapping with the FLORES-200 train set:

$$\mathcal{L}_{\text{overlap}} = \mathcal{L}_{\text{test}} \cap \mathcal{L}_{\text{FLORES-train}},$$

where $\mathcal{L}_{\text{test}}$ is the language set of the test dataset, and $\mathcal{L}_{\text{FLORES-train}}$ contains the 204 languages in the FLORES-200 train set. For a test embedding $\mathbf{e}^{\ell}_{\text{test,pool}}$, we compute its L2 distance to all reference embeddings in $\mathcal{R}^{\ell}_{\text{pool}}$:

$$d\left(\mathbf{e}^{\ell}_{\text{test,pool}}, \mathbf{e}^{\ell,(i,j)}_{\text{ref,pool}}\right) = \left\|\mathbf{e}^{\ell}_{\text{test,pool}} - \mathbf{e}^{\ell,(i,j)}_{\text{ref,pool}}\right\|_2^2, \quad \forall i \in \{1,\ldots,200\}, \quad \forall j \in \{1,\ldots,204\}.$$

The predicted language $\hat{y}_{\text{test}}$ is obtained via majority vote over the $k = 256$ nearest neighbors:

$$\hat{y}_{\text{test}} = \underset{l \in \mathcal{L}_{\text{overlap}}}{\arg\max} \sum_{(i,j) \in \mathcal{N}_k} \mathbf{1}(y^{(j)} = l),$$

where $\mathcal{N}_k$ denotes the set of indices for the top-k neighbors, and $\mathbf{1}$ is the indicator function.

Linear Classification Head To complement our non-parametric analysis, we probe the linear separability of language identity in Qwen-2.5 and LaBSE embeddings. This evaluates whether linguistic boundaries are geometrically aligned with hyperplanes in the hidden space, which would suggest that language control can be achieved through simple affine transformations.

Similar to the KNN-based approach, embeddings are extracted identically. For each dataset $\mathcal{D} \in$ {FLORES-200, NTREX-128, NusaX} and each layer $\ell \in \{1, m, L\}$ representing early, middle, and last layers respectively, we train a separate linear layer to map embeddings $\mathbf{e}^{\ell} \in \mathbb{R}^d$ to language logits $\mathbf{z}^{\ell} \in \mathbb{R}^C$, where C is the number of languages. The classifier for each layer is defined as:

$$\mathbf{z}^{\ell} = \mathbf{W}^{\ell}\mathbf{e}^{\ell} + \mathbf{b}^{\ell}, \quad \mathbf{W}^{\ell} \in \mathbb{R}^{C\times d}, \mathbf{b}^{\ell} \in \mathbb{R}^C,$$

with cross-entropy loss minimized during training.

E.1.2 Results

Our analysis reveals distinct layer-wise behaviors in language identification (LID) performance across LaBSE and Qwen2.5-0.5B models, focus on mean-pooled embedding.

KNN-based Language Identification The KNN method highlights significant performance variations across layers. As shown in Table 2, for LaBSE, the first layer achieves robust results, with mean F1 scores of 88.35% on FLORES-200, 90.43% on NTREX-128, and 81.78% on NusaX. Performance declines moderately in the middle layer, yielding 78.85% for FLORES-200, 81.30% for NTREX-128, and 45.37% for NusaX. The last layer exhibits catastrophic degradation, collapsing to 3.92%, 1.63%, and 0.00% on the respective datasets. This suggests that deeper LaBSE layers lose language-discriminative features critical for KNN classification.

For Qwen2.5-0.5B, the first layer similarly outperforms middle layers, with mean F1 scores of 83.69% on FLORES-200, 86.06% on NTREX-128, and 65.79% on NusaX. The middle layer shows the weakest results across all datasets: 55.32%, 54.73%, and 25.05%, respectively, while the last layer partially recovers to 71.73%, 81.86%, and 29.39%. This non-monotonic trend suggests limited retention of language-specific signals in the middle layer of Qwen2.5-0.5B.

LaBSE, trained for semantic alignment, shows severe degradation in its final layer, with near-zero F1 scores across datasets, as deeper layers erase language-specific signals required for KNN classification. In contrast, Qwen2.5-0.5B, a standard pretrained LLM, experiences a performance dip in its middle layer but recovers partially in the final layer, retaining sufficient linguistic discriminability. This divergence underscores a key architectural trade-off: contrastive models like LaBSE discard lexical or syntactic patterns in deeper layers to prioritize semantic invariance, while standard LLMs preserve partial language-identifying features across layers despite progressive abstraction.

Linear Classification Head For LaBSE, the First Layer consistently achieves the highest LID F1 scores across all datasets, with a significant drop in performance observed in the Last Layer. The NusaX dataset delivers the best overall results, particularly in the First Layer, where it reaches 97.30% F1 score. However, the Last Layer shows notably weaker performance, especially for the FLORES-200 and NusaX datasets. These findings suggest that earlier layers of LaBSE retain more language-identification-relevant features, such as surface-level linguistic cues, compared to deeper layers (see Table 2).

Similarly, in the Qwen2.5-0.5B model, the First Layer consistently outperforms the Middle Layer in LID F1 scores across all datasets. The NusaX dataset again produces the best results, with 95.55% F1 score, while NTREX-128 exhibits the lowest performance across all layers. These results indicate that the shallow First Layer of Qwen2.5-0.5B is more effective for language identification tasks than deeper layers, such as the Middle Layer, which shows weaker performance (refer to Table 2).

Overall, both models show that their highest LID performance occurs in the First Layer, with F1 scores declining as the layers get deeper. The NusaX dataset consistently yields the best performance, while the Last Layer in LaBSE and the Middle Layer in Qwen2.5-0.5B exhibit the weakest results. These trends suggest that shallow layers retain more language-specific information, which is crucial for language identification, likely due to their greater focus on surface-level features and general linguistic patterns. Table 14 further illustrate the comparative performance across layers and pooling techniques for both LaBSE and Qwen2.5-0.5B models.

Classifier Comparison: KNN vs. Linear Head As shown in Table 14, linear classifiers achieve superior F1 scores compared to KNN across layers, suggesting their ability to identify language-discriminative features within linearly separable subspaces. However, linear methods exhibit attenuated performance gaps between layers, for instance, the difference between first and middle layers in Qwen2.5-0.5B is less than 5% with linear classifiers, while KNN reveals differences exceeding 30%. Similarly, LaBSE's linear classifier reduces the last-layer performance gap to under 25%, whereas KNN shows near-complete degradation. This contrast implies that parametric linear methods, while more accurate overall, may obscure layer-specific language information degradation due to their reliance on learned projections. In contrast, KNN's non-parametric nature might more directly reflect the geometric structure of embeddings, amplifying sensitivity to layer-wise shifts in language information quality.

Pooling Method Comparison: CLS Token vs. Mean As shown in Table 14, the effectiveness of pooling strategies varies across layers. In first and middle layers, mean pooling achieves superior performance, with F1 margins exceeding 10% over CLS token pooling under KNN. However, in last layers, CLS token pooling shows limited resilience under KNN, marginally outperforming mean pooling in isolated cases despite near-random overall performance. Linear classifiers amplify mean pooling's advantage across all layers, suggesting its robustness to layer-specific degradation.

This suggests that mean pooling better preserves language-discriminative signals across layers, likely due to its aggregation of token-level features. In contrast, the CLS token, optimized for semantic tasks, exhibits sharper performance declines in deeper layers, particularly under non-parametric methods like KNN. These observations highlight the interplay between pooling strategy, layer depth, and classification method in language identification tasks.

F Annotation Guideline

F.1 Context of the Annotation Task

The annotation task involves evaluating the quality of cross-lingual language generation, where a model generates responses in a target language based on input prompts in a source language. The goal is to assess how well the model performs in terms of naturalness, relevance, and answer correctness. This evaluation is crucial for understanding the model's capabilities and identifying areas for improvement.

F.2 Detailed Scoring Guidelines

F.2.1 Naturalness (1-5):

- **1:** The response sounds very unnatural, robotic, or translated. It lacks fluency and typical language patterns of the target language, making it sound artificial and unnatural.
- **2:** The response is somewhat unnatural, with noticeable awkwardness or unnatural word choices. It may sound stilted or forced.
- **3:** The response is moderately natural, with some minor awkwardness but generally understandable. It flows reasonably well but has room for improvement.
- **4:** The response is mostly natural, with only slight deviations from typical language use. It sounds almost native-like but may have minor imperfections.
- **5:** The response is completely natural, indistinguishable from text written by a native speaker. It flows smoothly and uses language patterns typical of the target language.

F.2.2 Relevance (1-5):

- **1:** The response is completely irrelevant to the input prompt. It fails to address the topic or question posed.
- **2:** The response is somewhat relevant but misses key points or goes off-topic. It may touch on related ideas but does not fully address the prompt.

- **3:** The response is moderately relevant, addressing some aspects of the prompt but lacking completeness. It covers some key points but omits important details.
- **4:** The response is highly relevant, addressing most key points of the prompt. It provides a comprehensive answer but may miss minor details.
- **5:** The response is completely relevant, fully addressing all aspects of the prompt. It covers all key points and provides a thorough answer.

F.2.3 Correctness (1-5):

- **1:** The response contains major factual errors or inaccuracies. It provides incorrect information or contradicts known facts.
- **2:** The response contains some factual errors or inaccuracies. It may be partially correct but includes misleading or incorrect details.
- **3:** The response is mostly correct but may have minor inaccuracies or omissions. It is generally accurate but requires minor corrections.
- **4:** The response is highly accurate, with only minor details potentially incorrect. It is reliable and trustworthy but may have small errors.
- **5:** The response is completely accurate and factually correct. It provides precise and reliable information without any errors.

F.3 Additional Notes

- **Contextual Understanding:** Annotators should consider the context of the input prompt and the intended audience when evaluating naturalness and relevance. A response that is natural and relevant in one context may not be in another.
- **Consistency:** Annotators should strive for consistency in their annotations across different examples. This helps ensure that the evaluation is fair and reliable.
- **Examples:** Providing clear examples of each rating level for each category can help annotators understand the expected standards and make consistent judgments.
- **Feedback:** Encourage annotators to provide feedback on ambiguous cases or areas where the guidelines could be improved. This can help refine the annotation process and improve the quality of the evaluations.