LEAD: MODELS KNOW THE LANGUAGE, BUT TOKEN PREFERENCES DO NOT ALWAYS FOLLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

Multilingual large language models often experience *language confusion*, failing to consistently generate responses in the intended language. Existing mitigation approaches often require costly model retraining or inference-time interventions that assume access to target language labels. Moreover, they do not distinguish whether language confusion stems from a failure to encode the intended language or from a failure to utilize the encoded language information during generation. This paper introduces **Language Evidence-Aware Decoding** (LEAD), a lightweight method that exploits language-subspace logits from the final hidden state to selectively intervene only at decoding time. Our method is motivated by the $\mathcal{V}$-info theory and three empirical findings: linear probes identify languages with high accuracy, the logit contributions from language identity are weak, and causal interventions on the LM head reduce language confusion. Experiments on multiple multilingual LLMs across diverse languages demonstrate that LEAD substantially reduces language confusion while preserving general task performance.

1 INTRODUCTION

Recent advances in large language models (LLMs) have substantially improved their multilingual capabilities (Meta AI, 2025; Grattafiori et al., 2024; Yang et al., 2025). Modern LLMs support an increasingly broad range of languages and exhibit strong cross-lingual transfer across reasoning, creative writing, machine translation, and open-ended question answering (Achiam et al., 2023; Zhao et al., 2023; Team et al., 2025; Jiang et al., 2024). These advances have made multilingual interaction an increasingly important component of real-world LLM deployment. Despite this progress, multilingual LLMs still suffer from *language confusion: a model may unexpectedly insert tokens from an unintended language or even switch to a different language entirely during generation*. Recent studies (Zhang et al., 2026; Choo et al., 2026) have shown that it persists across increasingly capable LLMs, which is challenging and has attracted considerable attention from the research community.

Existing approaches mitigate language confusion primarily through model retraining or inference-time intervention. Retraining approaches explicitly penalize language-confused generations through preference optimization (Lee et al., 2025; Choo et al., 2026). While effective, these methods require additional training and may adversely affect the model's original performance on normal generation tasks. Intervention methods (Sterz et al., 2025; Xie et al., 2025; Lopo et al., 2025) instead construct language-identity directions from multilingual or parallel corpora and inject them into model activations at inference time. However, they assume that the desired target language is already known when selecting the corresponding steering direction during inference time. Although the recent norm-adjusted method is considerably lighter than model fine-tuning methods (Zhang et al., 2026), it still requires training an additional language classifier, and an incorrect language decision can lead to aggressive masking of otherwise valid candidates. Such dependency can become particularly restrictive under complex cross-lingual instructions, where the intended response language cannot always be reliably inferred from a fixed prompt pattern.

More fundamentally, prior work leaves an important mechanistic question unresolved: why does a model that is already highly multilingual still generate tokens in the wrong language? From the perspective of the data processing inequality, Shannon mutual information (Shannon, 1948) and $\mathcal{V}$-information theory (Xu et al., 2020) appear to yield conflicting conclusions. This discrepancy

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

reflects a fundamental distinction: having sufficient information does not imply that the underlying information can be fully identified and exploited. [1] Motivated by this difference, we argue that two distinct failure modes must be disentangled. First, the model may fail to infer the intended output language, such that its final representation encodes an incorrect language intent. Alternatively, the model may correctly represent the target-language intent, yet fail to assign this evidence sufficient influence over the final token ranking. Existing mitigation methods largely treat these two cases as a single problem, leaving unclear where language confusion actually emerges along the representation-to-logit pathway.

To investigate this question, we trace language information from the model's final hidden representation to its output logits. We first use linear probing to examine whether the final hidden state reliably encodes the intended response language. We then use LLMR and RLDR to quantify the extent to which language-identity evidence contributes to the ranking of candidate tokens. Finally, we conduct causal interventions on the language modeling head to verify its effect on generation. Together, these analyses reveal a consistent pattern: **The model typically retains a clear representation of the intended language, but the LM head does not fully utilize this language evidence when ranking candidate tokens.** Consequently, tokens from unintended languages can remain competitive in the candidate set despite the presence of the correct language intent, eventually giving rise to language confusion. These findings motivate **Language Evidence-Aware Decoding (LEAD)**, a lightweight intervention that mitigates language confusion directly at the output layer. LEAD uses SVD to isolate language-identity evidence already encoded in the model and traces its contribution to candidate-token logits, masking tokens in the decoding pool only when they receive insufficient support from the inferred target-language evidence.

Our main contributions are summarized as follows:

- First, through linear probing, language-identity contribution analysis, and causal intervention, we provide a mechanistic analysis of language confusion that explicitly distinguishes *target-language intent recognition* from *target-language evidence utilization*.
- Second, motivated by this insight, we introduce LEAD, a lightweight decoding-time intervention that extracts and exploits the model's intrinsic language-identity evidence to suppress confusion tokens.
- Third, we conduct extensive experiments across diverse multilingual LLMs and evaluation settings. LEAD consistently reduces language confusion compared with strong baselines, while preserving downstream task performance, demonstrating that language consistency can be significantly improved through targeted intervention on the model's existing language evidence.

2 Related work

2.1 Language Confusion

Language confusion has emerged as a critical challenge in multilingual large language models. Marchisio et al. (2024) formalized this phenomenon and introduced the Language Confusion Benchmark (LCB) for standardized evaluation. Subsequent studies linked language confusion to reduced language-identity information in intermediate representations (Lopo et al., 2025) and identified the final layers as critical for transitioning to the target language (Nie et al., 2025). However, these findings leave open whether language identity is lost or insufficiently used during output generation.

Existing mitigation methods broadly fall into training-based alignment and representation steering. Training-based approaches encourage language-consistent outputs through preference optimization (Lee et al., 2025; Choo et al., 2026). Steering methods intervene across layers or at selected intermediate layers, typically selecting language-identity vectors based on the labels (Sterz et al., 2025; Xie et al., 2025; Lopo et al., 2025). At the logit level, Zhang et al. (2026) use norm-based self-distillation to train a coarse four-class language-confusion gate.

[1] We provide detailed theoretical results in Appendix A.

2.2 Disentangled Representation Learning

Disentangling language-identity information from language-agnostic semantics is a long-standing challenge in multilingual language models. Prior work has explored this problem from several perspectives. Zhao et al. (2021) obtain language-agnostic representations from mBERT and XLM-R by explicitly aligning cross-lingual word pairs, followed by normalization of the latent spaces to zero mean and unit variance. From a subspace perspective, Yang et al. (2021) identify the dominant principal components of each language-identity embedding space as a major source of language bias and project them out to improve language invariance. Related studies further seek language-agnostic representations by identifying shared subspaces within multilingual language models (Gonen et al., 2020; Liang et al., 2021; Chang et al., 2022). Moving beyond language-wise subspace estimation, Xie et al. (2022) jointly model language-identity signals in the shared multilingual representation space, allowing language-dependent structure to be captured across languages rather than being identified independently for each language.

3 Lens into the Language Confusion

3.1 Language-Identity Decoding with Linear Probes

To examine whether language identity is encoded in the model's hidden states, we train a lightweight linear probe (Alain & Bengio, 2016) to predict the target language from the hidden representation at each layer. To reduce the possibility that the probe exploits prompt-template artifacts rather than target-language intent, we deliberately evaluate it on out-of-distribution data. Specifically, we train on monolingual data and test on cross-lingual data, and vice versa. In the monolingual setting, we omit explicit language-specific instructions. [2]

Figure 1 reports the layer-wise language identification accuracy. At the final layer, the probe achieves nearly perfect accuracy across both settings, with English being the main exception. These results indicate that the final hidden states preserve highly discriminative information about language identity.

To investigate whether language identity remains identifiable immediately before a confusion point, we further track the linear probe's predictions at each generation step. As shown in Figures 2a and 2b, the linear probe remains highly confident in the target language at the step preceding a confusion point. In addition, we compute the average probabilities assigned by the linear probe to the target language and the unintended language immediately before the confusion point, obtaining 0.937 and 0.008, respectively. These results suggest that the hidden states continue to encode the correct target-language intent even when language confusion occurs.

Our linear-probing analysis shows that target-language information remains linearly decodable in the final hidden states even under distribution shift. However, linear decodability alone does not imply that the LM head assigns sufficient ranking advantage to target-language tokens, which suggests a potential mismatch between the information encoded in the hidden states and the information used by the LM head for final token ranking.

3.2 Language-Identity Logit Contribution Analysis

At each generation step, the language model maps the hidden state h to vocabulary-sized logits through the unembedding matrix $W_{out} = [e_0, e_1, \ldots, e_{|V|-1}] \in \mathbb{R}^{d_{\text{model}} \times |V|}$, where e_i is the output embedding of token i. The logit of token i is therefore $\text{logit}_i = h^\top e_i$.

To quantify how much the LM head utilizes language-identity information, we project h onto the language-identity subspace constructed in Section 4.1. The logit decomposes accordingly:

$$h_a = h - h_s, \qquad \text{logit}_i = \underbrace{h_a^\top e_i}_{\text{logit}_{i,h_a}} + \underbrace{h_s^\top e_i}_{\text{logit}_{i,h_s}}. \tag{1}$$

Building on the decomposition, we introduce two metrics that quantify the role of language identity in the output distribution. For each token i, let c denote a confusion token and t the non-confusion

[2] Detailed descriptions of the training procedure and neuron-level analysis are provided in Appendix C.

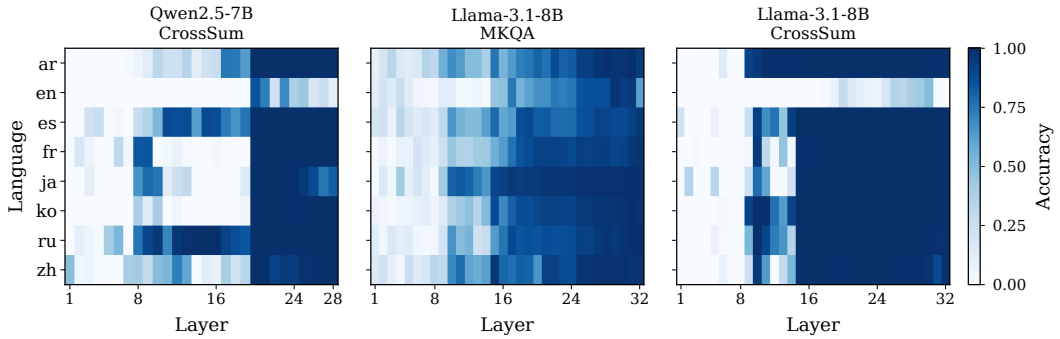
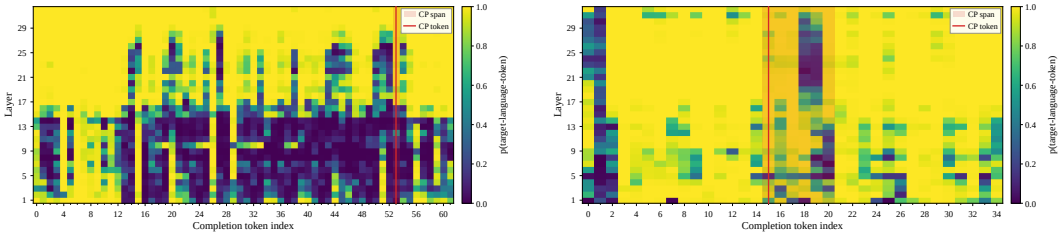


Figure 1: Per-language probe accuracy by layer on monolingual and cross-lingual generations. Left: Qwen2.5-7B linear probe trained on MKQA, then evaluated on CrossSum. Middle and right: Llama3.1-8B, evaluated out of domain on MKQA and CrossSum, respectively, with probes trained on the opposite dataset.



(a) MKQA (target language = Japanese)

(b) MKQA (target language = German)

Figure 2: Case studies on the confidence of target language prediction before the confusion point.

token ranked below c. We define

$$\mathrm{LLMR}(i) = \frac{\mathrm{logit}_{i,h_s}^2}{\mathrm{logit}_i^2}, \qquad \mathrm{RLDR}(c,t) = \frac{\mathrm{logit}_{c,h_a} - \mathrm{logit}_{t,h_a}}{\mathrm{logit}_{c,h_s} - \mathrm{logit}_{t,h_s}}. \tag{2}$$

Language Logit Magnitude Ratio (LLMR) quantifies the squared magnitude of the language-identity logit component relative to the total logit: Values near 0 imply a weak contribution from the language-identity component to the total logit.

Residual-to-Linguistic Delta Ratio (RLDR) compares the residual and linguistic logit gaps between a confusion token and its subsequent normal token. When $\mathrm{RLDR}(c,t) < -1$, the residual component is the dominant factor that elevates the confusion token above its neighbour, which means that the confusion occurs because the residual signal overwhelms the linguistic identity signal.

As shown in Table 1, although language identity is encoded in the hidden state, its contribution to the logits is relatively small. During token ranking, the LM head jointly reflects language identity, residual compatibility, and other factors. The relatively weak contribution of language identity may therefore allow tokens from a non-target language to obtain high ranks, eventually leading to language confusion.

Table 1: Average LLMR(RLDR) across different languages.

Model	ar	hi	ja	ko	ru	zh
Qwen3-8B	2.78%(-1.12)	10.15%(-1.07)	4.19%(-1.08)	5.32%(-1.09)	2.69%(-1.23)	3.17%(-1.14)
Qwen3-30B-A3B	2.05%(-1.15)	4.96%(-1.10)	2.44%(-1.08)	3.28%(-1.09)	1.35%(-1.24)	0.82%(-1.22)
Llama3.1-8B	3.20%(-1.14)	4.10%(-1.15)	3.94%(-1.13)	3.26%(-1.16)	2.35%(-1.41)	2.97%(-1.10)

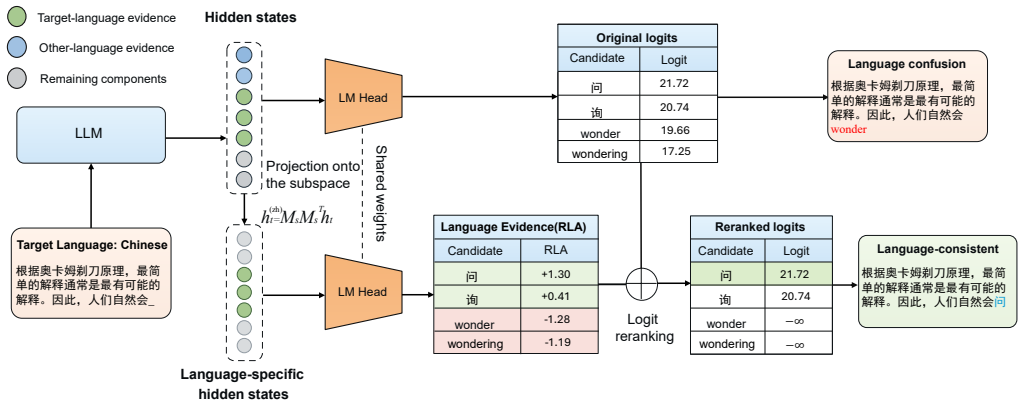


Figure 3: Overview of the Language Evidence-Aware Decoding (LEAD) framework.

4 LEAD: Language Evidence-Aware Decoding

We propose a lightweight inference-time intervention method for mitigating language confusion without modifying the representation or requiring model retraining. Our method consists of three steps. Figure 3 illustrates the framework. First, we construct a language-identity subspace from multilingual hidden representations. Second, at each decoding step, we project h_t onto this subspace and use the projected component to measure the language-identity logit contribution of each candidate token. Third, we map these contributions into relative advantages within the current top-k candidate set. Tokens with insufficient language evidence are masked before sampling. Further methodological details and hyperparameter settings are provided in Appendix D.

4.1 Language-Identity Logit Contribution

Assume that the model processes inputs written in L languages. For each language l, we collect the final-token representations from multilingual inputs and compute their mean representation:

$$m_l = \frac{1}{n_l} \sum_{j=1}^{n_l} r_l^j, \tag{3}$$

where $r_l^j \in \mathbb{R}^d$ is the final-token representation of the j-th sample written in language l, and n_l is the number of samples for language l. By concatenating the mean vectors $\boldsymbol{m}_l$ for all L languages along the column axis, we construct a mean embedding matrix $\boldsymbol{M} \in \mathbb{R}^{d \times L}$ that characterizes the multilingual latent space. Following the formulation of Piratla et al. (2020) and Xie et al. (2022), we decompose the multilingual mean-representation matrix $\boldsymbol{M}$ into a language-shared component $\boldsymbol{M}_a$ and a language-identity component $\boldsymbol{M}_s$. We obtain this decomposition by solving

$$\begin{aligned} \min_{\boldsymbol{M}_a, \boldsymbol{M}_s, \boldsymbol{\Gamma}} \quad & \left\| \boldsymbol{M} - \boldsymbol{M}_a \mathbb{1}^\top - \boldsymbol{M}_s \boldsymbol{\Gamma}^\top \right\|_F^2 \\ \text{s.t.} \quad & \mathrm{Span}(\boldsymbol{M}_a) \perp \mathrm{Span}(\boldsymbol{M}_s), \end{aligned} \tag{4}$$

where $\boldsymbol{M}_a \in \mathbb{R}^{d \times 1}$, $\boldsymbol{M}_s \in \mathbb{R}^{d \times r}$, and the coefficient matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{L \times r}$ specifies the language-dependent coordinates in this subspace.

The optimization can be efficiently solved via Singular Value Decomposition. At generation step t, we project the final hidden state onto this subspace and compute its norm-adjusted logit contribution to token i:

$$\boldsymbol{h}_{t,s} = \boldsymbol{M}_s \boldsymbol{M}_s^\top \boldsymbol{h}_t, \qquad \boldsymbol{S}_{t,i} = \frac{\boldsymbol{h}_{t,s}^\top \boldsymbol{e}_i}{\|\boldsymbol{e}_i\|}, \tag{5}$$

where we adjust the contribution by the token embedding norm, in order to remove the systematic advantage of high norm tokens (Zhang et al., 2026). In practice, however, the benefits of this normalization are very modest, further supporting the effectiveness of our language-identity subspaces, which can mitigate the advantage of high resource languages without relying on token norm adjustment. For completeness, we provide a more detailed explanation and ablation study in Appendix D.4.

4.2 Relative Language Advantage

We observe that the distribution of $\boldsymbol{S}_{t,i}$ can vary substantially across decoding steps. To account for this variation, we compute the relative language advantage of each token with respect to the top-k candidate set at the current step. Specifically, let μ_t and σ_t denote the mean and standard deviation of the language-evidence scores within this candidate set. The relative language advantage of token i is defined as

$$\boldsymbol{r}_{t,i} = \frac{\boldsymbol{S}_{t,i} - \boldsymbol{\mu}_t}{\boldsymbol{\sigma}_t}. \tag{6}$$

This within-step normalization removes the scale variation across decoding steps while preserving the relative ordering of language-identity evidence among candidate tokens. Further analysis suggests that the relative language-evidence scores of target-language and language-confused tokens exhibit a bimodal distribution that can be approximated by a two-component Gaussian mixture. We provide this analysis in Appendix B.

4.3 Masking Workflow

During inference, we apply a relative advantage masking method through three steps:

Intervention Criterion We classify tokens by language using their Unicode names to determine whether to activate the intervention.[3] At each decoding step, we consider two candidate sets obtained using $(\text{top-}k = 5,\ \text{top-}p = 0.999)$ and $(\text{top-}k = 20,\ \text{top-}p = 0.95)$, respectively. We activate the intervention only if at least one of these candidate sets contains tokens from more than one language.

Candidate Masking When the intervention criterion is met, we evaluate the relative language-evidence score $\boldsymbol{r}_{t,i}$ for each candidate. Candidates with $\boldsymbol{r}_{t,i} < 0$ are treated as having insufficient support for the target language and masked before sampling.

Nucleus Expansion Masking can leave a small nucleus empty when it contains only confusion tokens. We therefore progressively expand the candidate pool by admitting additional tokens in descending order of their original model probabilities. After each expansion, we reapply the masking rule, computing language-evidence scores relative to the fixed top-k candidate set, and stop once at least one candidate remains.

5 Experiments

5.1 Experimental Setup

Target Languages and Base Models We evaluate the effectiveness of LEAD in mitigating *language confusion* across six target languages: Chinese, Arabic, Korean, Japanese, Russian and Hindi. The base models employed in our experiments are **Qwen3-30B-A3B** (Yang et al., 2025), **Qwen3-8B**, **Llama3.1-8B** (Grattafiori et al., 2024).

Baselines and Evaluation Benchmarks For comparative analysis, we employ in-context learning (3-shot), greedy decoding and language confusion gate (Zhang et al., 2026) as strong baseline methods.

Our evaluation is twofold: language confusion assessment and general accuracy assessment. We evaluate language confusion on **LCB-crosslingual**, **LCB-monolingual** (Marchisio et al., 2024),

[3] A detailed description of the classification workflow is provided in Appendix D.1.

Table 2: Average Response Pass Rate (RPR) and Word Pass Rate (WPR). Values are presented as RPR(WPR). All values are in percentages.

Model	Method	LCB (cross-lingual)	LCB (monolingual)	CrossSum (cross-lingual)	FLORES-NO-LATIN (cross-lingual)
Llama3.1-8B	**Baseline**	71.77(91.85)	94.28(97.99)	58.52(91.24)	87.75(96.43)
	3-shot	75.35(94.83)	94.75(98.85)	86.62(97.69)	92.57(97.79)
	greedy-decoding	75.92(94.32)	96.44(98.56)	67.27(94.26)	93.14(98.05)
	LCG	77.80(95.45)	97.69(99.16)	78.14(98.46)	94.03(98.67)
	LEAD	**87.75**(**97.41**)	**98.33**(**99.42**)	**87.87**(**99.20**)	**97.51**(**99.43**)
Qwen3-8B	**Baseline**	77.79(94.05)	95.72(98.77)	76.78(98.06)	95.02(99.25)
	3-shot	84.02(**95.61**)	85.64(98.28)	84.24(98.52)	94.90(98.97)
	greedy-decoding	77.96(94.15)	96.72(99.15)	78.29(98.13)	96.19(99.32)
	LCG	79.46(94.57)	96.83(99.13)	88.49(99.01)	98.17(99.62)
	LEAD	**84.33**(95.11)	**97.44**(**99.51**)	**93.16**(**99.05**)	**98.85**(**99.86**)
Qwen3-30B	**Baseline**	85.12(94.83)	96.50(98.42)	85.15(98.06)	98.21(99.55)
	3-shot	84.77(95.70)	87.53(**99.44**)	**94.34**(99.36)	98.78(99.72)
	greedy-decoding	84.50(94.80)	96.31(98.44)	85.45(98.30)	98.75(99.72)
	LCG	85.57(95.23)	96.67(99.00)	91.00(99.27)	99.39(**99.98**)
	LEAD	**88.58**(**95.92**)	**97.06**(99.18)	93.56(**99.75**)	**99.65**(99.88)

CrossSum (Bhattacharjee et al., 2023), and **FLORES+** (Team et al., 2022). We evaluate general task performance across three major benchmarks, including **MGSM** (Shi et al., 2023): mathematical reasoning, **XWinograd** (Muennighoff et al., 2023): commonsense reasoning, and **MMMLU** (Hendrycks et al., 2021): knowledge-intensive question answering. Further details on evaluation settings are described in Appendix F.

5.2 Definition of Language Confusion Metrics

To quantitatively evaluate language confusion, we employ four metrics: Word Pass Rate (WPR), Response Pass Rate (RPR)[4], Latin confusion Rate (Latin) and CJ confusion Rate (CJ). **WPR** (Marchisio et al., 2024) measures language confusion at the word level. **RPR** (Choo et al., 2026) measures the proportion of non-confused responses out of the total responses generated by the LLM. A "non-confused response" is defined as a response sequence that consists entirely of the target language. **CJ** measures the percentage of responses in low-resource language settings that contain at least one Chinese character or Japanese kana. **Latin** measures the percentage of responses containing at least one Latin-script character that is inconsistent with the intended output language.

5.3 Evaluation Results

Comparison with Baseline Methods Table 2 shows that LEAD consistently improves both RPR and WPR over the baseline. On Llama3.1-8B, RPR (WPR) increases from 71.77 (91.85) to 87.75 (97.41) on LCB-crosslingual and from 58.52 (91.24) to 87.87 (99.20) on CrossSum. Compared with LCG, LEAD achieves higher average RPR across all reported settings and higher WPR in most settings. On CrossSum with Llama3.1-8B, the gains over LCG are 9.73 percentage points in RPR. This advantage is consistent with LEAD's finer-grained language evidence, which can help distinguish languages grouped together by LCG's four-class gate and mitigate intra-class confusion. In contrast, greedy decoding and 3-shot prompting exhibit more task-dependent behavior. Greedy decoding does not explicitly enforce language consistency, while few-shot prompting can provide strong gains in some settings but remains less consistent across benchmarks and incurs additional context overhead. Table 3 further reports Latin and CJ confusion rates. On Llama3.1-8B, LEAD achieves the lowest rates in all settings except CJ confusion on LCB-monolingual. On Qwen3-8B, it consistently reduces Latin confusion relative to LCG, by an average of 6.44 percentage points across the two LCB settings and CrossSum. On Qwen3-30B, LEAD reduces CJ confusion to zero in all settings except CrossSum, where it achieves a CJ confusion rate of 1.73%, compared with 1.61% for LCG. Overall, LEAD generally achieves the best results in reducing language confusion.

[4]Following Choo et al. (2026), we remove units, e-mail addresses, and proper nouns before computing metrics to avoid penalizing legitimate code-switching.

Table 3: Average Latin confusion rate (Latin) and CJ confusion rate (CJ). Values are presented as Latin(CJ). All values are in percentages.

Model	Method	LCB (cross-lingual)	LCB (monolingual)	CrossSum (cross-lingual)	FLORES-NO-LATIN (cross-lingual)
Llama3.1-8B	**Baseline**	55.10(3.09)	15.19(1.92)	65.71(18.99)	12.62(3.17)
	3-shot	45.37(4.94)	15.83(1.50)	33.16(5.47)	6.39(2.57)
	greedy-decoding	53.07(1.83)	14.64(0.67)	60.88(13.31)	7.59(1.73)
	LCG	46.36(0.34)	11.81(**0.00**)	46.68(1.60)	6.39(0.39)
	LEAD	**28.95**(**0.17**)	**7.50**(0.25)	**23.46**(**1.33**)	**2.87**(**0.14**)
Qwen3-8B	**Baseline**	52.51(0.34)	23.50(0.33)	55.38(14.25)	4.25(1.52)
	3-shot	41.98(1.01)	25.61(1.42)	37.90(9.36)	4.33(2.15)
	greedy-decoding	52.45(0.67)	23.17(0.25)	53.64(12.01)	3.29(1.27)
	LCG	48.22(**0.00**)	21.00(**0.00**)	40.48(**0.93**)	1.08(0.18)
	LEAD	**40.96**(0.08)	**16.86**(0.17)	**32.56**(1.69)	**0.99**(**0.14**)
Qwen3-30B	**Baseline**	45.20(0.08)	23.25(**0.00**)	53.15(5.05)	1.67(0.21)
	3-shot	42.35(0.42)	24.28(0.25)	**24.38**(2.54)	1.34(0.07)
	greedy-decoding	45.47(0.25)	23.75(0.25)	54.19(4.53)	1.20(0.18)
	LCG	40.62(0.17)	21.64(**0.00**)	39.88(**1.61**)	**0.35**(0.04)
	LEAD	**36.94**(**0.00**)	**17.31**(**0.00**)	34.58(1.73)	0.40(**0.00**)

Table 4: Evaluation results on normal task performance. We report the average accuracy and standard deviation over three runs with different random seeds. These results suggest that LEAD does not adversely affect task performance.

Model	Method	MGSM	M-MMLU	XWinograd
Llama3.1-8B	**Baseline**	54.08 ± 1.00	39.47 ± 1.16	59.06 ± 1.79
	LEAD	54.12 ± 1.31	39.07 ± 1.00	61.69 ± 1.90
Qwen3-8B	**Baseline**	60.29 ± 0.39	41.69 ± 0.66	74.38 ± 2.62
	LEAD	60.45 ± 0.56	42.07 ± 0.22	74.75 ± 1.43
Qwen3-30B	**Baseline**	60.63 ± 0.48	73.05 ± 1.11	75.50 ± 1.14
	LEAD	60.87 ± 0.18	71.98 ± 0.36	76.65 ± 1.27

Effects On Task Performance Preserving task performance is essential when mitigating language confusion. We evaluate LEAD on MGSM, M-MMLU, and XWinograd to assess its impact on standard task performance. As shown in Table 4, LEAD achieves performance comparable to the baseline across different tasks, with improvements on some benchmarks. This indicates that LEAD mitigates language confusion without compromising task performance.

Additional Baseline Comparisons Existing activation steering methods require knowing the target language of each prompt in advance so that the matching steering vector can be applied, making them impractical in real-world serving where no oracle language label is available. For completeness, we provide a direct comparison with ITLC (Lopo et al., 2025) and the training-based method ORPO (Lee et al., 2025) in Appendix G.1. The results show that LEAD still outperforms these methods.

5.4 Causal Intervention Directly On the LM-HEAD

To further investigate whether the LM head underweights language-identity evidence, we explicitly strengthen the language-identity component of the LM head. We first identify tokens associated with Chinese, Arabic and Russian using Unicode names, and extract language-identity subspaces from the corresponding LM-head row vectors. Within each language-identity token set, we replace the projected language component of each token with the one having the largest ℓ_2 norm, thereby equalizing language-identity strength while leaving the residual component unchanged. We evaluate the modified LM head on LCB using Qwen3-8B and Llama3.1-8B, with RPR as the metric. To determine whether the observed effect is driven by strengthening the language-identity direction, we introduce two controls: *norm scaling*, which increases the norm of the language-identity component to the maximum value, and *random direction*, which adds a random component while keeping the final norm of the language-identity direction unchanged.

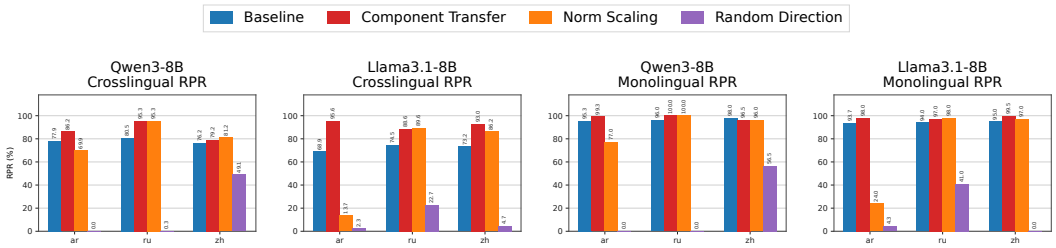


Figure 4: Comparison of the baseline LM head with three interventions: component transfer, norm scaling, and random direction. Component transfer most consistently alleviates language confusion.

As shown in Figure 4, component transfer substantially improves RPR, especially on the LCB-*cross* setting, where RPR for Arabic increases from 68.9% to 95.6% on Llama3.1-8B. In contrast, the random-direction control causes a substantial decline, while norm scaling produces mixed effects, improving performance in some settings but reducing it in others. We speculate that the gains from norm scaling arise because increasing token norms can enlarge the corresponding dot products, thereby improving token ranking in some settings. These results provide causal evidence that language-identity strength is unevenly distributed across token representations in the LM head: tokens with weaker language-identity components receive insufficient language-consistent support during ranking, confirming our hypothesis that language identity, although clearly encoded in final hidden states, does not reliably dominate the token-selection decision.

5.5 Verifying the Linguistic Nature of the Subspace

As shown in Figure 5, we visualize the representations of the final token at the last layer across languages, both before and after ablating the language-identity subspace M_s. Prior to ablation, representations form distinct per-language clusters, confirming that the hidden states encode a clear language-identity signal. After projection, these clusters collapse into a substantially tighter, language-agnostic grouping: representations that previously occupied well-separated regions of the embedding space now overlap, indicating that the removed subspace captured the dominant axes of cross-lingual variation.

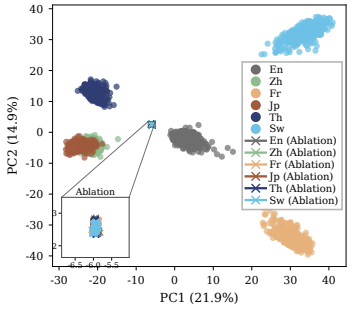


Figure 5: PCA of final-token hidden states before and after projection in Llama3.1-8B (last layer).

6 Conclusion and Limitations

In this paper, we present **Language Evidence-Aware Decoding (LEAD)**, a lightweight decoding framework that mitigates language confusion by leveraging the language evidence already encoded in the model's hidden states. Unlike existing approaches, LEAD requires no additional neural network training and no modification to model parameters; it relies solely on a simple language subspace to read the model's language evidence at inference time, without degrading the model's original performance. Furthermore, we provide a deeper mechanistic analysis of language confusion, revealing that the model encodes sufficient language-identity information in its final hidden states, yet the LM head fails to fully exploit this evidence, thereby giving rise to language confusion. Although our method can distinguish confusion across different languages, we do not further investigate confusion among languages within the Latin-script family. We leave it to future work to extend LEAD to this finer-grained scenario.

AI USE STATEMENT

In this work, we used generative AI tools just for refining grammar and improving readability of the writing. We have not used generative AI tools for formulating research questions, designing methodologies, generating results, or drawing conclusions, and tasks such as data collection and annotation are not applicable to this work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Once the discussion forums open, we will make a comment directed to the reviewers and area chairs and put a link to an anonymous repository. In addition, the main paper and appendix provide sufficient details to reproduce our results, with further methodological details available in Section 4 and Appendix D.

REFERENCES

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

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Uddin Ahmad, Yuan-Fang Li, Yong-Bin Kang, and Rifat Shahriyar. CrossSum: Beyond English-centric cross-lingual summarization for 1,500+ language pairs. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2541–2564, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.143. URL https://aclanthology.org/2023.acl-long.143/.

Tyler A. Chang, Zhuowen Tu, and Benjamin K. Bergen. The geometry of multilingual language model representations. *CoRR*, abs/2205.10964, 2022. doi: 10.48550/arXiv.2205.10964. URL https://doi.org/10.48550/arXiv.2205.10964.

Jinho Choo, JunSeung Lee, Jimyeong Kim, Yeeho Song, S. K. Hong, and Yeong-Dae Kwon. TLPO: Token-level policy optimization for mitigating language confusion in large language models. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42670–42690, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1976. URL https://aclanthology.org/2026.acl-long.1976/.

Hila Gonen, Shauli Ravfogel, Yanai Elazar, and Yoav Goldberg. It's not Greek to mBERT: Inducing word-level translations from multilingual BERT. In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 45–56, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.blackboxnlp-1.5. URL https://aclanthology.org/2020.blackboxnlp-1.5.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand,

Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Nahyun Lee, Yeongseo Woo, Hyunwoo Ko, and Guijin Son. Controlling language confusion in multilingual llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. Association for Computational Linguistics, 2025.

Haonan Li, Fajri Koto, Minghao Wu, Alham Fikri Aji, and Timothy Baldwin. Bactrian-x : A multilingual replicable instruction-following model with low-rank adaptation. *CoRR*, abs/2305.15011, 2023. doi: 10.48550/ARXIV.2305.15011. URL https://doi.org/10.48550/arXiv.2305.15011.

Sheng Liang, Philipp Dufter, and Hinrich Schütze. Locating language-specific information in contextualized embeddings. *CoRR*, abs/2109.08040, 2021. URL https://arxiv.org/abs/2109.08040.

Shayne Longpre, Yi Lu, and Joachim Daiber. MKQA: A linguistically diverse benchmark for multilingual open domain question answering. *Trans. Assoc. Comput. Linguistics*, 9:1389–1406, 2021. doi: 10.1162/TACL_A_00433. URL https://doi.org/10.1162/tacl_a_00433.

Joanito Agili Lopo, Muhammad Ravi Shulthan Habibi, Tack Hwa Wong, Muhammad Ilham Ghozali, Fajri Koto, Genta Indra Winata, Peerat Limkonchotiwat, Alham Fikri Aji, and Samuel Cahyawijaya. Language surgery in multilingual large language models. In David Ifeoluwa Adelani, Catherine Arnett, Duygu Ataman, Tyler A. Chang, Hila Gonen, Rahul Raja, Fabian Schmidt, David Stap, and Jiayi Wang (eds.), *Proceedings of the 5th Workshop on Multilingual Representation Learning (MRL 2025)*, pp. 438–467, Suzhuo, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-345-6. doi: 10.18653/v1/2025.mrl-main.30. URL https://aclanthology.org/2025.mrl-main.30/.

Kelly Marchisio, Wei-Yin Ko, Alexandre Berard, Théo Dehaze, and Sebastian Ruder. Understanding and mitigating language confusion in LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 6653–6677, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.380. URL https://aclanthology.org/2024.emnlp-main.380/.

Geoffrey J McLachlan, Sharon X Lee, and Suren I Rathnayake. Finite mixture models. *Annual review of statistics and its application*, 6(1):355–378, 2019.

Meta AI. The llama 4 herd: The beginning of a new era of natively multimodal intelligence. https://ai.meta.com/blog/llama-4-multimodal-intelligence/, 2025. Meta AI Blog.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, et al. Crosslingual generalization through multitask finetuning. In *The 61st Annual Meeting Of The Association For Computational Linguistics*, 2023.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pp. 807–814, Madison, WI, USA, 2010. Omnipress. ISBN 9781605589077.

Ercong Nie, Helmut Schmid, and Hinrich Schütze. Mechanistic understanding and mitigation of language confusion in english-centric large language models, 2025. URL https://arxiv.org/abs/2505.16538.

Vihari Piratla, Praneeth Netrapalli, and Sunita Sarawagi. Efficient domain generalization via common-specific low-rank decomposition. In *International conference on machine learning*, pp. 7728–7738. PMLR, 2020.

Shauli Ravfogel, Yoav Goldberg, and Ryan Cotterell. Log-linear guardedness and its implications. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9413–9431, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.523. URL https://aclanthology.org/2023.acl-long.523/.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, et al. Language models are multilingual chain-of-thought reasoners. In *The Eleventh International Conference on Learning Representations*, 2023.

Hannah Sterz, Fabian David Schmidt, Goran Glavaš, and Ivan Vulić. ReCoVeR the target language: Language steering without sacrificing task performance. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 19390–19405, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1056. URL https://aclanthology.org/2025.findings-emnlp.1056/.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. Language-specific neurons: The key to multilingual capabilities in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5701–5715, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.309. URL https://aclanthology.org/2024.acl-long.309/.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

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, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. No language left behind: Scaling human-centered machine translation, 2022. URL https://arxiv.org/abs/2207.04672.

Martin B Wilk and Ram Gnanadesikan. Probability plotting methods for the analysis of data. *Biometrika*, 55(1):1–17, 1968.

Yunfan Xie, Lixin Zou, Dan Luo, Min Tang, Chenliang Li, Liming Dong, and Xiangyang Luo. Mitigating language confusion through inference-time intervention. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics, COLING 2025, Abu Dhabi, UAE, January 19-24, 2025*, pp. 8418–8431. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.coling-main.563/.

Zhihui Xie, Handong Zhao, Tong Yu, and Shuai Li. Discovering low-rank subspaces for language-agnostic multilingual representations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 5617–5633, 2022.

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. *arXiv preprint arXiv:2002.10689*, 2020.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Ziyi Yang, Yinfei Yang, Daniel Cer, and Eric Darve. A simple and effective method to eliminate the self language bias in multilingual representations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5825–5832, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.470. URL https://aclanthology.org/2021.emnlp-main.470.

Collin Zhang, Fei Huang, Chenhan Yuan, and Junyang Lin. Language confusion gate: Language-aware decoding through model self-distillation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 75989–76007, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/7afa43ddf76300bd086de27dc591555f-Paper-Conference.pdf.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

Wei Zhao, Steffen Eger, Johannes Bjerva, and Isabelle Augenstein. Inducing language-agnostic multilingual representations. In *Proceedings of *SEM 2021: The Tenth Joint Conference on Lexical and Computational Semantics*, pp. 229–240, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.starsem-1.22. URL https://aclanthology.org/2021.starsem-1.22.

A $\mathcal{V}$-INFO THEORY

In this section, following prior work (Xu et al., 2020; Ravfogel et al., 2023), we use the $\mathcal{V}$-info theory (Xu et al., 2020) to show why the data processing inequality does not necessarily hold in the $\mathcal{V}$-information framework.[5]

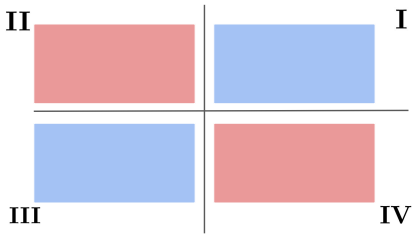


Figure 6: Construction of a log-linear model that breaks data processing inequality.

A.1 PRELIMINARY

We follow prior work (Xu et al., 2020) and formally define predictive $\mathcal{V}$-information.

Definition 1 (Predictive Family). *Let $\Omega = \{f : \mathcal{X} \cup \{\varnothing\} \to \mathcal{P}(\mathcal{Y})\}$. We say that $\mathcal{V} \subseteq \Omega$ is a predictive family if it satisfies*

$$\forall f \in \mathcal{V}, \forall P \in \text{range}(f), \quad \exists f' \in \mathcal{V}, \quad s.t. \quad \forall x \in \mathcal{X}, f'[x] = P, f'[\varnothing] = P \tag{7}$$

A predictive family is a set of predictive models, and only models belonging to this family are permitted for prediction. We refer to the property captured by Equation 7 as *optional ignorance*. Intuitively, it means that the available side information may be ignored when making predictions.

Definition 2 (Predictive conditional $\mathcal{V}$-entropy). *Let X, Y be two random variables taking values in $\mathcal{X} \times \mathcal{Y}$, and $\mathcal{V}$ be a predictive family. Then the predictive conditional $\mathcal{V}$-entropy is defined as*

$$H_{\mathcal{V}}(Y|X) = \inf_{f \in \mathcal{V}} \mathbb{E}_{x,y \sim X,Y} \left[-\log f[x](y)\right]$$
$$H_{\mathcal{V}}(Y|\varnothing) = \inf_{f \in \mathcal{V}} \mathbb{E}_{y \sim Y} \left[-\log f[\varnothing](y)\right]$$

We call $H_{\mathcal{V}}(Y|\varnothing)$ the $\mathcal{V}$-entropy, and also denote it as $H_{\mathcal{V}}(Y)$.

Shannon mutual information is a measure of changes in entropy when conditioning on new variables:

$$I(X;Y) = H(Y) - H(Y|X) = H_{\Omega}(Y) - H_{\Omega}(Y|X) \tag{8}$$

Here, we will use $I_{\mathcal{V}}(X \to Y)$ to represent the *change* in predictability of an output variable Y when given side information X.

Definition 3 (Predictive $\mathcal{V}$-information). *Let X, Y be two random variables taking values in $\mathcal{X} \times \mathcal{Y}$, and $\mathcal{V}$ be a predictive family. The predictive $\mathcal{V}$-information from X to Y is defined as*

$$I_{\mathcal{V}}(X \to Y) = H_{\mathcal{V}}(Y|\varnothing) - H_{\mathcal{V}}(Y|X) \tag{9}$$

A.2 DATA PROCESSING INEQUALITY

Notably, the data processing inequality does not generally hold for predictive $\mathcal{V}$-information. In contrast to Shannon mutual information, predictive $\mathcal{V}$-information is defined relative to a restricted family of predictors. Although a transformation $t(X)$ is computed from X and therefore cannot

[5] In this work, the $\mathcal{V}$-predictor family is understood specifically as the family of binary log-linear classifiers.

increase the underlying information about Y, it may make task-relevant information more accessible to predictors in $\mathcal{V}$. In particular, the transformed representation can expose patterns that are difficult to exploit directly from X, so that $I_{\mathcal{V}}(t(X) \to Y)$ may exceed $I_{\mathcal{V}}(X \to Y)$. Whether such an increase is possible depends on the closure properties of the predictive family under composition with t. In the following section, we employ the instance constructed by Ravfogel et al. (2023) to show the failure mode of data processing inequality.

A.3 Construction of a Log-linear Model that Breaks Data Processing Inequality

Consider the data in $\mathbb{R}^2$ presented in Figure 6, where the distribution $p(\mathcal{X}, \mathcal{Z})$ has 4 distinct clusters, each with a different label from $\mathcal{Z}$, corresponding to Voronoi regions formed by the intersection of the axes. The red clusters correspond to $\mathcal{Z} = \top$ and the blue clusters correspond to $\mathcal{Z} = \bot$. The data is taken to be log-linearly guarded with respect to $\mathcal{Z}$. The XOR labels cannot be recovered by a binary affine log-linear classifier. Hence, for the family $\mathcal{V}$ of binary log-linear models, $\mathrm{I}_{\mathcal{V}}(\mathcal{X} \to \mathcal{Z}) = 0$.

Now consider a four-class softmax classifier whose columns are

$$\begin{aligned} \boldsymbol{\theta}_1^\star &= \alpha[1,1]^\top, & \boldsymbol{\theta}_2^\star &= \alpha[-1,1]^\top, \\ \boldsymbol{\theta}_3^\star &= \alpha[-1,-1]^\top, & \boldsymbol{\theta}_4^\star &= \alpha[1,-1]^\top, \end{aligned} \qquad \alpha > 0. \tag{10}$$

As α increases, the softmax output identifies the quadrant of each input:

$$\lim_{\alpha \to \infty} p(\widehat{\mathcal{Y}} = i \mid \mathcal{X} = \boldsymbol{x}_i) = 1, \qquad i \in \{1,2,3,4\}. \tag{11}$$

A subsequent binary linear classifier can then assign the same label to classes 1 and 3, and the other label to classes 2 and 4. Consequently, $\mathrm{I}_{\mathcal{V}}(\widehat{\mathcal{Y}} \to \mathcal{Z}) \to 1$ while $\mathrm{I}_{\mathcal{V}}(\mathcal{X} \to \mathcal{Z}) = 0$. Thus, the multi-class softmax layer converts information that is not linearly usable from the original representation into a discrete representation that can be exploited by a subsequent linear classifier. This provides a concrete counterexample to the data-processing inequality for $\mathcal{V}$-information. A detailed theoretical treatment of this construction is provided in Ravfogel et al. (2023); we present the XOR example here only for illustration.

B Two Component Gaussian model

Empirically, the relative language advantages in the top-20 candidates exhibit a bimodal tendency. We therefore approximate their distribution using two Gaussian components (McLachlan et al., 2019):

$$p(r) = \pi_+ \mathcal{N}\left(r; \mu_+, \sigma_+^2\right) + \pi_- \mathcal{N}\left(r; \mu_-, \sigma_-^2\right), \tag{12}$$

where

$$\mu_+ > \mu_-. \tag{13}$$

The high-mean component, denoted by $c = +$, corresponds to candidates with relatively strong language evidence, whereas the low-mean component, denoted by $c = -$, corresponds to candidates with relatively weak language evidence. The two components are fitted on a calibration set using the relative language-advantage scores. For fitting the Gaussian model, we utilize the training split of Bactrian-X, a multilingual instruction-following dataset (Li et al., 2023). The empirical validity of Gaussian approximation is evaluated using Q–Q plots (Wilk & Gnanadesikan, 1968).

The Validity of Gaussian Approximation We next examine the distributional structure of the relative language evidence after normalization. We separately collect the relative language scores of target-language and confusion tokens and assess their normality using Q–Q plots.

As shown in Figure 8, both token groups exhibit a strong linear correspondence with the theoretical Gaussian quantiles across two model families. For Llama3.1-8B, the Q–Q correlations are 0.999 and 0.995 for target-language and confusion tokens, with KS statistics of 0.013 and 0.038. Similarly, Qwen3-8B achieves correlations of 0.999 and 0.990, with KS statistics of 0.022 and 0.043. Only mild deviations are observed in the extreme tails.

These results suggest that, after relative normalization, the class-conditional language evidence of both target-language and confusion tokens can be well approximated by Gaussian distributions.

C Linear Probe Experiments

C.1 Linear Probe Training Details

Training Data The probe training set is constructed using examples from MKQA (Longpre et al., 2021) and CrossSum (Bhattacharjee et al., 2023).

Feature Extraction For each prompt, we perform a single forward pass through the frozen causal language model and extract the hidden representation at the last token. A separate probe is trained independently for each layer, allowing us to measure how linearly accessible language identity becomes throughout the network depth.

Optimization The linear classifier contains no dropout. Each probe is trained with the cross-entropy objective using AdamW. We use a batch size of 256, weight decay of 10^{-4}, and evaluate two learning rates, 10^{-4} and 10^{-5}. Training runs for at most 50 epochs, with early stopping based on validation loss using a patience of 3 epochs. We retain the checkpoint with the lowest validation loss.

Models and Layer Coverage We conduct probing experiments on two models: Llama3.1-8B-Instruct and Qwen2.5-7B-Instruct. For Llama3.1-8B-Instruct, we train probes on all 32 layers. For Qwen2.5-7B-Instruct, we train probes on all 28 layers.

C.2 Training Results

The final-layer Llama probe with learning rate 10^{-4} achieves about 99.1% accuracy on the test sets. Qwen exhibits a similar pattern in deeper layers, reaching approximately 99% test accuracy, whereas the shallow layers encode language identity substantially less explicitly. For example, the first-layer Qwen probe achieves only approximately 63–77% accuracy depending on the learning rate.

C.3 FFN Neuron-Level Analysis of Probe Accuracy

Analyzing the accuracy of the linear probes, we find that the accuracies for languages other than English are consistently high. As shown in Figure 1, there is a substantial accuracy gap between English and the other languages. This gap is particularly pronounced in the cross-lingual setting, where the accuracy of identifying English is nearly zero. Since the model does not exhibit evident confusion when generating English, we hypothesize that processing English may not require a strong, explicitly encoded language-identity signal. To examine this hypothesis, we identify language-specific neurons using LAPE (Tang et al., 2024) and interpret this phenomenon from a neuron-level perspective. [6]

To identify language-specific neurons, we follow LAPE. For the j-th neuron in the i-th layer, we define its activation probability on language k as

$$p_{i,j}^{k} = \mathbb{E}\left[\mathbb{I}\left(\text{act_fn}(\tilde{\boldsymbol{h}}^{i}\boldsymbol{W}_{1}^{i})_{j} > 0\right) \mid \text{language } k\right], \tag{14}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. The resulting vector $\boldsymbol{p}_{i,j} = (p_{i,j}^{1}, \ldots, p_{i,j}^{\ell})$ describes the neuron's activation tendency across languages. After applying L1 normalization to obtain $\boldsymbol{p}'_{i,j}$, we quantify its language selectivity using the language activation probability entropy:

$$\text{LAPE}_{i,j} = -\sum_{k=1}^{\ell} p'^{k}_{i,j} \log p'^{k}_{i,j}. \tag{15}$$

[6] Following prior work (Nair & Hinton, 2010), we define a *neuron* as a linear transformation of a single column of $\boldsymbol{W}_1^i$ followed by a nonlinear activation. Accordingly, the FFN module is formulated as $\boldsymbol{h}^i = \text{act_fn}(\tilde{\boldsymbol{h}}^i\boldsymbol{W}_1^i) \cdot \boldsymbol{W}_2^i$.

Neurons with low LAPE values are regarded as language-specific neurons, since their activations are concentrated on one or a small number of languages.

We then examine the number of language-specific neurons. Figure 9 demonstrates that English has the fewest such neurons. This observation suggests that processing English relies less on language-identity features; instead, language information may be entangled with semantic and other general representations, thereby hindering the linear separability of English-specific information.

D METHOD DETAILS

D.1 CLASSIFY TOKENS BY LANGUAGES

Before decoding, we assign a script label to every token in the vocabulary. For each token, we first decode it from its byte-pair encoding (BPE) into Unicode characters. We then determine the script of each character from its Unicode properties. Whitespace, punctuation, symbols, numbers, control and format characters, and combining marks are treated as non-linguistic and ignored. Each remaining character is mapped to a script according to its Unicode name: **Latin**; Chinese and Japanese (**CJ**), for characters whose name contains CJK, IDEOGRAPHIC, HIRAGANA, or KATAKANA; **Hangul**, **Cyrillic**, **Arabic**, **Devanagari**, **Thai**, **Bengali**, and **Telugu**; and **Other** for characters that match none of these.

Rather than resolving each token to a single family through a prioritized heuristic, we encode the set of scripts present in a token as a bitmask. A token that contains no linguistic characters is labeled **None**; a token containing exactly one script receives that script; and a token containing two or more scripts is labeled **Mixed**.

A known challenge with BPE is that some tokens may represent incomplete Unicode characters. Following Zhang et al. (2026), we infer the family of such tokens from the continuous block structure of the Unicode standard. This step applies to tokens whose decoded string contains a replacement character, excluding the 256 single-byte base tokens:

1. **Byte Sequence Decomposition.** The token is converted back into its raw byte sequence, which is fed into a UTF-8 state machine that breaks it down into logically complete or partial UTF-8 units. We identify *left partial* units: byte sequences that correctly start a multi-byte character but are incomplete.

2. **Inferring Code Point Bounds.** The first two bytes of a left partial unit determine the range of code points it could represent if completed. For a 3-byte sequence, we obtain the full range of possibilities by setting the missing third byte to its minimum valid value (`0x80`) and then to its maximum valid value (`0xBF`); for a 4-byte sequence, both missing bytes range over all valid continuation values.

3. **Overlapping with Language Blocks.** If the inferred range overlaps any Han ideograph block, namely CJK Unified Ideographs (`U+4E00–U+9FFF`), Extensions B through I, or the CJK Compatibility Ideographs Supplement, the **CJ** bit is added to the token's bitmask.

D.2 LANGUAGE SUBSPACE CONSTRUCTION

We construct the language-identity subspace using examples from Bactrian-X andvCrossSum, covering 19 languages. For each example, we extract the hidden state at the final input position from the last layer. We then apply SVD and retain an orthonormal basis $\boldsymbol{M}_s$ of rank 18. As shown in Appendix G.3, performance generally improves with the subspace rank, but the differences become marginal at higher ranks.

The optimal solution of Equation 4 can be computed efficiently via Singular Value Decomposition (SVD). Algorithm 1 presents the detailed procedure. The only hyperparameter $r < L$ controls the amount of language-identity information captured by the identified subspace. Readers interested in more details can consult the proof provided in Xie et al. (2022). Besides, we provide the rank scaling law experiments in Appendix G.3.

Algorithm 1: Construction of the Language-Identity Subspace

In: Languages' mean embeddings $\boldsymbol{M}$, subspace rank r
Out: shared component $\boldsymbol{M}_a$, language-identity orthonormal basis $\boldsymbol{M}_s$, coordinates $\boldsymbol{\Gamma}$

/* 1) Approximate $\boldsymbol{M}$ in low rank */

$\boldsymbol{M}'_a \leftarrow \frac{1}{L}\boldsymbol{M}\mathbb{1}_L$;

$\boldsymbol{M}'_s, \boldsymbol{\Sigma}', \boldsymbol{V}' \leftarrow \text{Top-}r \text{ SVD}\left(\boldsymbol{M} - \boldsymbol{M}'_a\mathbb{1}_L^\top\right)$;

$\boldsymbol{\Gamma}' \leftarrow \boldsymbol{V}'\boldsymbol{\Sigma}'$;

$\boldsymbol{M}' \leftarrow \boldsymbol{M}'_a\mathbb{1}_L^\top + \boldsymbol{M}'_s(\boldsymbol{\Gamma}')^\top$;

/* 2) Enforce orthogonality */

$\boldsymbol{M}_a \leftarrow \frac{\left((\boldsymbol{M}')^\top\right)^+\mathbb{1}_L}{\left\|((\boldsymbol{M}')^\top)^+\mathbb{1}_L\right\|_2^2}$;

$\boldsymbol{M}_s, \boldsymbol{\Sigma}, \boldsymbol{V} \leftarrow \text{Top-}r \text{ SVD}\left(\boldsymbol{M}' - \boldsymbol{M}_a\mathbb{1}_L^\top\right)$;

$\boldsymbol{\Gamma} \leftarrow \boldsymbol{V}\boldsymbol{\Sigma}$;

D.3 RLA Hyperparameter Choice

We use the top-20 tokens as the relative language advantage calculation set in our experiments. This empirically chosen cutoff filters out noisy tokens in the tail of the logit distribution.

Table 5: Percentage of tokens in each language family with embedding norms among top 5% of all token embedding norms.

Model	CJ%	Latin%	Low-Res%
Qwen3-8B	10.74	4.61	0.14
Qwen3-30B-A3B	6.52	5.50	0.07
Llama3.1-8B	4.38	5.95	1.34
Gemma3-12B	0.94	5.04	2.40

D.4 Token Norm Analysis

D.4.1 Norm-bias in LM Head

Zhang et al. (2026) report a pronounced imbalance in the distribution of high-norm tokens across language families: Latin and CJ tokens are overrepresented among the top 5% of tokens by embedding norm, whereas low-resource language tokens are substantially underrepresented. As shown in Table 5, the results confirm a significant imbalance: high-resource languages like Latin and CJ consistently dominate the high-norm group, while low-resource languages are heavily underrepresented.

D.4.2 Norm Ablation Study

In Tables 6 and 7, we compare our default setting, which divides the logits by the token norm (+NORM), with the variant that disables this normalization (− NORM). Overall, removing token-norm normalization produces only minor changes across both models and all evaluation settings, with no consistent advantage for either variant. This indicates that explicitly dividing the logits by token norms has a limited effect on the final results. More importantly, our subspace-based method can already push high-resource Latin and CJ languages toward lower-ranked positions without relying on token-norm normalization, demonstrating the effectiveness of the proposed approach.

Table 6: Token-norm ablation on Qwen3-8B. +NORM represents the logits divided by the token norm; −NORM disables ℓ_2 token-norm.

	FLORES-NO-LATIN				LCB (Cross)				LCB (Mono)			
	Latin	CJ	WPR	RPR	Latin	CJ	WPR	RPR	Latin	CJ	WPR	RPR
+NORM	0.99	0.14	99.86	98.85	40.96	0.08	95.11	84.33	16.86	0.17	99.51	97.44
−NORM	0.87	0.21	99.88	98.99	41.41	0.25	94.76	84.60	17.25	0.08	99.50	97.22

Table 7: Token-norm ablation on Llama3.1-8B. +NORM represents the logits divided by the token norm; −NORM disables ℓ_2 token-norm.

	FLORES-NO-LATIN				LCB (Cross)				LCB (Mono)			
	Latin	CJ	WPR	RPR	Latin	CJ	WPR	RPR	Latin	CJ	WPR	RPR
+NORM	2.87	0.14	99.43	97.51	28.95	0.17	97.41	87.75	7.50	0.25	99.42	98.33
−NORM	3.43	1.45	99.27	96.43	28.34	0.75	97.45	87.13	6.64	1.25	99.50	97.50

E FEW-SHOT PROMPTING

We adopt the few-shot examples from Marchisio et al. (2024).[7] Figure 7 illustrates the prompting template used for the base model.

Q: Write your answer in French. How should I choose what cheese to buy?
A: Il existe de nombreux types de fromages différents, donc le choix du fromage à acheter dépend des préférences personnelles, de la disponibilité et de l'utilisation prévue. [. . .]
Q: What is the difference between pets and cattle? Reply in Arabic.
A: [. . .]

Figure 7: Template used for few-shot prompting the base models. The model's answers are truncated to prevent the generation of new questions.

F MORE DETAILS ON EVALUATION SETTINGS

Baselines We compare our LEAD approach with five baseline intervention mechanisms: in-context learning (3-shot), greedy decoding, ITLC (Lopo et al., 2025), ORPO (Lee et al., 2025) and LCG (Zhang et al., 2026). For the 3-shot baseline, we provide the prompt template used for constructing few-shot demonstrations in Appendix E. The ITLC baseline is representative of activation-steering methods, while the LCG baseline employs norm-based self-distillation to train a four-class language-confusion gate that masks tokens not predicted by the gate when necessary.

Datasets Table 8 summarizes the tasks used to evaluate WPR, RPR, CJ, and Latin, along with the number of instances and languages covered by each task. Unless otherwise specified, all evaluations are conducted in a generative setting with zero-shot Chain-of-Thought (CoT) prompting, except for the 3-shot setting. For cross-lingual evaluation, LCB-cross primarily consists of complex prompts, where instructions involving multiple languages may appear at different positions within the prompt, thereby providing a more flexible and challenging cross-lingual setting. In contrast, the CrossSum and FLORES+ series provide explicit cross-lingual instructions. We further divide FLORES+ into NO-LATIN and WITH-LATIN subsets, and only use the NO-LATIN to measure language confusion. To evaluate language consistency under monolingual contexts, we use LCB-mono as a controlled same-language setting.

[7]The few-shot demonstrations are presented in languages different from the target language. They are intended to clarify the task rather than specify the output language.

Table 8: Tasks used for confusion-rate and accuracy evaluation. When a single #Instances value is shown, it is the count *per language*; slash-separated counts follow the listed language order.

Task	Confusion-rate	Acc.	Languages	#Instances
LCB (cross-lingual)	O		zh, ja, ar, ru, ko, hi	299
LCB (monolingual)	O		zh, ja, ar, ru, ko, hi	900
FLORES-NO-LATIN	O		zh, ja, ar, ru, ko, hi	709
CrossSum	O		zh, ja, ar, ru, ko	1255
MGSM		O	bn, de, es, fr, ja, ru, sw, te, th, zh	250
MMMLU		O	bn, de, es, fr, ja, sw, zh	200
XWinograd		O	fr, ja, ru, zh	83/200/200/200

Implementation Details All experiments were conducted on a single node equipped with one RTX PRO6000 GPU.

G Additional Experimental Results

G.1 Comparison with Other Methods

For completeness, we compare our method with the steering method ITLC and the training-based method ORPO. As shown in Tables 9 and 10, our method substantially outperforms ITLC on Llama3.1-8B and Qwen3-8B, while achieving competitive performance on Qwen3-30B. Thus, even though ITLC requires the language label to be known in advance and requires model-specific hyper-parameter tuning, our method still performs better overall.

We further compare LEAD with the training-based method ORPO on Llama3.1-8B and Qwen3-8B. As shown in Tables 11 and 12, LEAD consistently achieves higher RPR and lower Latin and CJ confusion rates across all evaluated benchmarks, often by substantial margins. WPR is also competitive, with only slightly lower scores in a few settings.

G.2 Natural Code-Switching

We evaluate whether LEAD preserves the model's natural code-switching capability on **FLORES-WITH-LATIN**, a subset of FLORES in which the reference translations contain Latin characters and therefore provide contexts where code-switching may naturally occur.

We use Qwen3-30B and Qwen3-30B+LCG as reference systems. Prior work has shown that LCG retains a degree of natural code-switching capability (Zhang et al., 2026), making it a useful reference for examining whether LEAD excessively suppresses such behavior.

As shown in Table 13, LEAD reduces the overall code-switching rate for both base models. For example, the rate of Qwen3-8B decreases from 34.98% to 22.77%. Importantly, the resulting rates remain comparable to those of Qwen3-30B (24.15%) and Qwen3-30B+LCG (20.30%). Similarly, Llama3.1-8B with LEAD achieves a rate of 22.44%. In addition, LEAD achieves very low Latin intrusion rates on **FLORES-NO-LATIN** (2.87% for Llama3.1-8B and 0.99% for Qwen3-8B), while the rates exceed 20% on **FLORES-WITH-LATIN**. The contrast suggests that the higher Latin-character rate in **FLORES-WITH-LATIN** reflects LEAD's ability to preserve natural code-switching. These results indicate that LEAD does not indiscriminately eliminate code-switching. Instead, while reducing excessive language mixing, it retains code-switching rates comparable to the stronger reference systems.

G.3 Rank Scaling Experiment

To better understand how the rank of the language subspace affects performance, we conduct a scaling experiment on three models by progressively increasing the rank and measuring the resulting change in confusion metrics.

As shown in Figure 10, language confusion generally decreases as the rank increases. This trend is consistent with the intuition that a higher-rank subspace captures richer language-identity information, thereby providing a more complete basis for removing linguistic bias.

G.4 Detailed Experimental Results by Target Language

The results demonstrate that in most configurations, LEAD mitigates language confusion more effectively than other comparative methods. Tables 14 through 19 provide detailed RPR, WPR, CJ and Latin broken down by model and target language. Missing entries in the table indicate that the corresponding language is not present in the dataset or that Chinese and Japanese were excluded from the CJ confusion rate.

Table 9: Direct comparison between LEAD and ITLC on language-confusion benchmarks in terms of RPR and WPR. Higher values are better. All values are percentages.

Model	Method	LCB (cross-lingual) RPR (WPR)	LCB (monolingual) RPR (WPR)	CrossSum RPR (WPR)	FLORES-NO-LATIN RPR (WPR)
Llama3.1-8B	**LEAD (ours)**	**87.75(97.41)**	**98.33(99.42)**	**87.87(99.20)**	**97.51(99.43)**
	ITLC	77.43(92.81)	95.72(98.91)	63.21(91.05)	87.80(96.21)
Qwen3-8B	**LEAD (ours)**	**84.33(95.11)**	**97.44(99.51)**	**93.16(99.05)**	**98.85(99.86)**
	ITLC	82.81(94.21)	96.31(98.90)	75.78(97.32)	94.92(99.04)
Qwen3-30B	**LEAD (ours)**	**88.58(95.92)**	97.06(99.18)	**93.56(99.75)**	**99.65(99.88)**
	ITLC	87.40(95.90)	**97.11(99.83)**	93.11(99.26)	98.85(99.69)

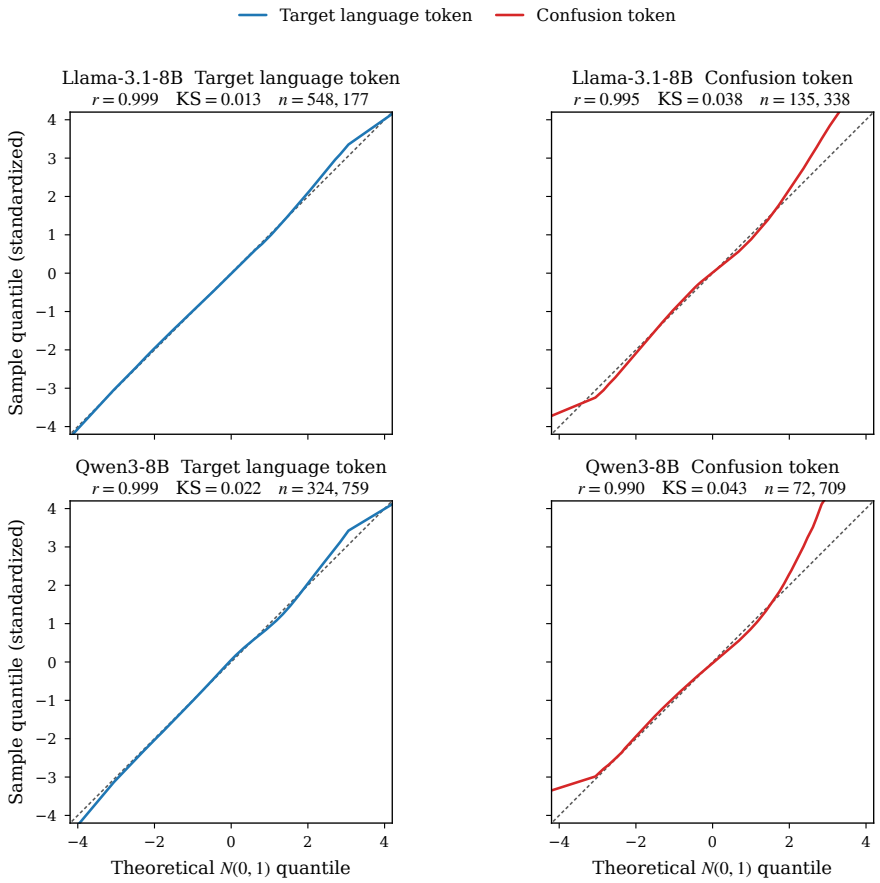


Figure 8: Q–Q analysis of relative language evidence for target-language and confusion tokens on Llama3.1-8B and Qwen3-8B.

Table 10: Direct comparison between LEAD and ITLC on language-confusion benchmarks in terms of Latin and CJ confusion rates. Lower values are better. All values are percentages.

Model	Method	LCB (cross-lingual) Latin (CJ)	LCB (monolingual) Latin (CJ)	CrossSum Latin (CJ)	FLORES-NO-LATIN Latin (CJ)
Llama3.1-8B	**LEAD (ours)**	**28.95(0.17)**	**7.50(0.25)**	**23.46(1.33)**	**2.87(0.14)**
	ITLC	51.27(2.42)	15.78(1.75)	67.50(17.24)	13.92(4.30)
Qwen3-8B	**LEAD (ours)**	**40.96(0.08)**	**16.86(0.17)**	**32.56(1.69)**	**0.99(0.14)**
	ITLC	48.72(0.67)	24.22(0.83)	56.01(14.91)	4.37(1.69)
Qwen3-30B	**LEAD (ours)**	**36.94(0.00)**	**17.31(0.00)**	34.58(1.73)	**0.40(0.00)**
	ITLC	39.60(0.34)	20.58(0.33)	**30.13(1.59)**	1.15(0.14)

Table 11: Comparison of LEAD and ORPO on language-confusion benchmarks in terms of RPR and WPR. Higher values are better. All values are percentages.

Model	Method	LCB (cross-lingual) RPR (WPR)	LCB (monolingual) RPR (WPR)	CrossSum RPR (WPR)	FLORES-NO-LATIN RPR (WPR)
Llama3.1-8B	**LEAD**	**87.75(97.41)**	**98.33**(99.42)	**87.87(99.20)**	**97.51(99.43)**
	ORPO	79.26(95.93)	94.30(**99.80**)	50.46(89.24)	91.18(99.34)
Qwen3-8B	**LEAD**	**84.33(95.11)**	**97.44**(99.51)	**93.16**(99.05)	**98.85(99.86)**
	ORPO	81.33(94.10)	96.06(**99.88**)	75.39(**99.29**)	94.24(99.56)

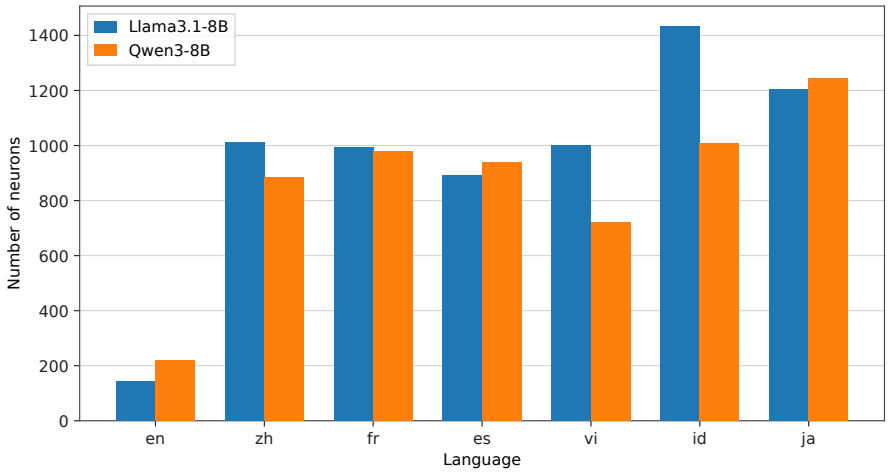


Figure 9: Number of language-specific neurons in Llama3.1-8B-Instruct and Qwen3-8B. English has substantially fewer specialized neurons than other languages.

Table 12: Comparison of LEAD and ORPO on language-confusion benchmarks in terms of Latin and CJ confusion rates. Lower values are better. All values are percentages.

Model	Method	LCB (cross-lingual) Latin (CJ)	LCB (monolingual) Latin (CJ)	CrossSum Latin (CJ)	FLORES-NO-LATIN Latin (CJ)
Llama3.1-8B	**LEAD**	**28.95(0.17)**	**7.50(0.25)**	**23.46(1.33)**	**2.87(0.14)**
	ORPO	46.80(4.09)	14.25(2.00)	62.16(23.59)	8.13(3.60)
Qwen3-8B	**LEAD**	**40.96(0.08)**	**16.86(0.17)**	**32.56(1.69)**	**0.99(0.14)**
	ORPO	47.77(1.34)	21.89(1.42)	55.29(15.84)	4.70(2.08)

Table 13: Impact of LEAD on legitimate code-switching behavior.

Model	No LEAD	LEAD	Qwen3-30B	Qwen3-30B+LCG
Llama3.1-8B	45.76	22.44	24.15	20.30
Qwen3-8B	34.98	22.77	24.15	20.30

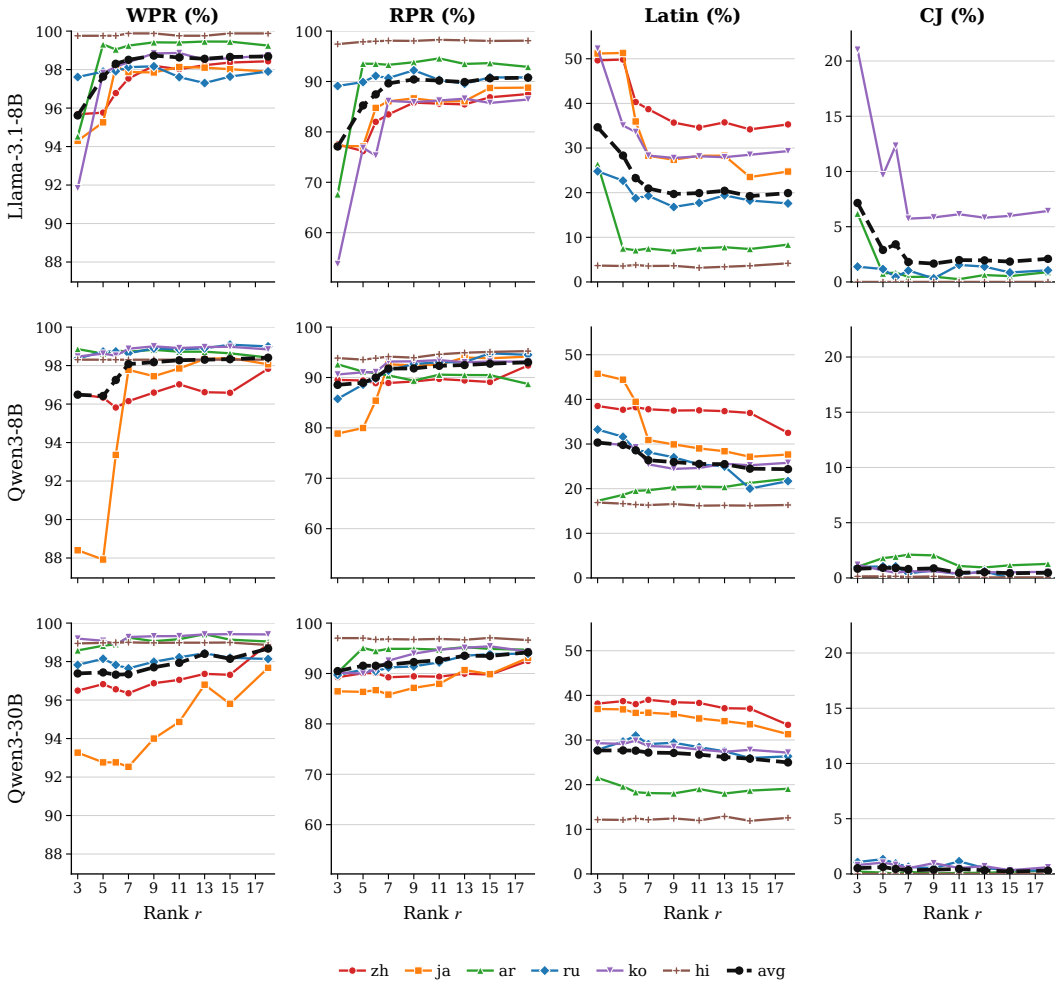


Figure 10: Rank Scaling Experiments. As the rank increases, RPR and WPR exhibit upward trends, whereas Latin and CJ decline.

Lang.	Method	LCB (cross-lingual)	LCB (monolingual)	CrossSum (cross-lingual)	FLORES-NO-LATIN (cross-lingual)
zh	Baseline	62.24(–)	18.50(–)	72.40(–)	14.25(–)
	3-shot	56.22(–)	16.00(–)	47.20(–)	5.50(–)
	Greedy decoding	57.58(–)	15.50(–)	66.80(–)	11.00(–)
	ITLC	63.27(–)	16.00(–)	77.20(–)	15.51(–)
	LCG	55.89(–)	12.50(–)	61.20(–)	13.40(–)
	LEAD (ours)	47.50(–)	8.00(–)	39.20(–)	4.37(–)
ar	Baseline	49.50(3.00)	10.67(0.67)	56.98(10.47)	15.66(1.97)
	3-shot	43.21(1.33)	13.00(1.00)	16.67(1.55)	8.89(1.55)
	Greedy decoding	46.86(2.67)	8.33(0.67)	46.12(5.43)	9.59(1.13)
	ITLC	43.83(2.00)	11.67(0.00)	44.57(10.47)	14.95(3.81)
	LCG	38.49(0.33)	8.33(0.00)	29.46(0.39)	3.10(0.14)
	LEAD (ours)	18.40(0.00)	2.00(0.00)	5.81(0.39)	1.13(0.00)
ko	Baseline	65.93(7.70)	29.00(5.00)	75.50(43.37)	20.45(9.87)
	3-shot	55.59(17.41)	28.00(4.00)	40.56(14.86)	12.55(8.46)
	Greedy decoding	62.27(3.67)	28.00(1.00)	70.28(33.33)	11.28(5.50)
	ITLC	68.28(5.68)	26.00(7.00)	84.34(37.75)	25.39(12.98)
	LCG	52.56(1.01)	17.00(0.00)	53.82(4.42)	9.31(1.27)
	LEAD (ours)	34.82(0.67)	15.00(1.00)	28.92(3.21)	3.95(0.28)
ja	Baseline	65.26(–)	23.00(–)	61.57(–)	11.42(–)
	3-shot	48.88(–)	26.00(–)	36.36(–)	1.41(–)
	Greedy decoding	64.94(–)	26.00(–)	58.68(–)	6.63(–)
	ITLC	57.57(–)	26.00(–)	66.53(–)	13.12(–)
	LCG	62.93(–)	24.00(–)	47.93(–)	9.45(–)
	LEAD (ours)	41.14(–)	16.00(–)	21.49(–)	4.80(–)
hi	Baseline	40.50(0.33)	3.00(0.00)	– (–)	3.39(0.14)
	3-shot	32.14(1.01)	5.00(0.00)	– (–)	2.40(0.00)
	Greedy decoding	42.53(0.67)	1.00(0.00)	– (–)	1.41(0.14)
	ITLC	30.13(1.34)	4.00(0.00)	– (–)	5.36(0.28)
	LCG	32.12(0.00)	3.00(0.00)	– (–)	0.85(0.14)
	LEAD (ours)	9.72(0.00)	0.00(0.00)	– (–)	0.14(0.00)
ru	Baseline	47.20(1.34)	7.00(2.00)	62.11(3.12)	10.58(0.71)
	3-shot	36.18(0.00)	7.00(1.00)	25.00(0.00)	7.62(0.28)
	Greedy decoding	44.22(0.33)	9.00(1.00)	62.50(1.17)	5.64(0.14)
	ITLC	44.54(0.67)	11.00(0.00)	64.84(3.52)	9.17(0.14)
	LCG	36.19(0.00)	6.00(0.00)	41.02(0.00)	2.26(0.00)
	LEAD (ours)	22.09(0.00)	4.00(0.00)	21.88(0.39)	2.82(0.28)

Table 14: Latin (CJ) confusion rate on Llama3.1-8B. Values are presented as Latin (CJ) in %.

Lang.	Method	LCB (cross-lingual)	LCB (monolingual)	CrossSum (cross-lingual)	FLORES-NO-LATIN (cross-lingual)
zh	Baseline	73.20(92.52)	95.00(95.95)	58.40(91.87)	91.11(97.64)
	3-shot	74.86(94.57)	98.50(99.48)	89.20(96.40)	96.33(97.74)
	Greedy decoding	79.87(96.32)	96.00(97.42)	69.20(95.56)	93.94(97.13)
	ITLC	78.87(92.80)	97.00(98.48)	73.60(87.60)	92.67(94.77)
	LCG	75.19(96.79)	97.50(97.98)	69.60(97.32)	91.26(96.05)
	LEAD (ours)	86.60(97.07)	99.00(99.49)	71.89(99.52)	98.03(99.43)
ar	Baseline	68.85(92.50)	93.67(97.99)	42.25(89.58)	81.66(93.14)
	3-shot	69.47(95.14)	92.00(98.62)	86.05(97.29)	88.58(96.33)
	Greedy decoding	74.87(94.14)	96.67(98.99)	60.47(92.86)	89.56(96.46)
	ITLC	72.54(91.51)	94.33(98.99)	52.71(91.41)	80.68(95.63)
	LCG	80.22(96.84)	97.67(100.00)	71.71(99.16)	92.81(99.28)
	LEAD (ours)	91.27(98.85)	100.00(100.00)	92.64(100.00)	98.31(99.86)
ko	Baseline	63.82(87.43)	87.00(99.00)	34.94(93.44)	76.02(96.85)
	3-shot	64.47(94.21)	90.00(97.00)	71.49(99.20)	81.38(98.03)
	Greedy decoding	73.20(93.81)	92.00(98.00)	45.78(97.15)	87.31(99.15)
	ITLC	65.17(92.98)	90.00(98.99)	35.34(91.57)	71.51(96.19)
	LCG	74.88(92.80)	97.00(100.00)	68.67(99.18)	91.82(98.71)
	LEAD (ours)	87.60(97.17)	95.00(99.00)	92.37(100.00)	97.60(99.72)
ja	Baseline	66.65(89.83)	98.00(98.99)	85.54(92.34)	90.83(97.94)
	3-shot	78.84(94.82)	95.00(100.00)	93.39(97.93)	98.03(99.15)
	Greedy decoding	64.13(89.84)	98.00(98.98)	85.12(93.16)	94.78(98.70)
	ITLC	80.87(92.99)	97.00(100.00)	80.58(92.56)	93.94(96.19)
	LCG	67.41(93.39)	97.00(100.00)	90.50(97.05)	91.26(98.81)
	LEAD (ours)	76.79(93.88)	99.00(100.00)	93.80(100.00)	94.08(99.41)
hi	Baseline	83.55(96.76)	98.00(100.00)	– (–)	97.04(98.73)
	3-shot	80.54(95.54)	98.00(98.99)	– (–)	97.60(98.73)
	Greedy decoding	82.87(97.44)	100.00(100.00)	– (–)	98.59(99.72)
	ITLC	90.60(97.13)	100.00(100.00)	– (–)	95.20(99.15)
	LCG	88.24(98.43)	100.00(100.00)	– (–)	98.45(99.58)
	LEAD (ours)	94.29(99.29)	100.00(100.00)	– (–)	99.86(100.00)
ru	Baseline	74.52(92.07)	94.00(96.00)	71.48(88.98)	89.84(94.27)
	3-shot	83.90(94.68)	95.00(98.98)	92.97(97.65)	93.51(96.76)
	Greedy decoding	80.55(94.36)	96.00(97.98)	75.78(92.55)	94.64(97.16)
	ITLC	76.55(89.45)	96.00(96.97)	73.83(92.13)	92.81(95.35)
	LCG	80.87(94.45)	97.00(96.97)	90.23(99.61)	98.59(99.57)
	LEAD (ours)	89.95(98.22)	97.00(98.00)	88.67(96.47)	97.18(98.16)

Table 15: Response Pass Rate (RPR) and Word Pass Rate (WPR) on Llama3.1-8B. Values are presented as RPR (WPR) in %.

Lang.	Method	LCB (cross-lingual)	LCB (monolingual)	CrossSum (cross-lingual)	FLORES-NO-LATIN (cross-lingual)
zh	Baseline	60.58(–)	30.00(–)	52.00(–)	1.69(–)
	3-shot	50.57(–)	37.00(–)	41.20(–)	1.69(–)
	Greedy decoding	61.92(–)	31.00(–)	52.00(–)	1.69(–)
	ITLC	56.24(–)	31.00(–)	51.60(–)	1.55(–)
	LCG	60.24(–)	32.00(–)	48.80(–)	0.99(–)
	LEAD (ours)	55.23(–)	26.50(–)	41.20(–)	0.56(–)
ar	Baseline	45.55(0.67)	18.00(0.33)	45.35(15.50)	7.90(2.12)
	3-shot	33.84(1.35)	20.67(0.67)	25.58(10.47)	7.90(3.39)
	Greedy decoding	47.55(1.35)	20.00(0.00)	44.96(12.02)	5.64(1.83)
	ITLC	42.20(0.67)	20.33(0.33)	47.29(15.12)	7.62(1.97)
	LCG	45.21(0.00)	18.00(0.00)	21.32(0.78)	0.71(0.00)
	LEAD (ours)	39.86(0.00)	14.67(0.67)	27.91(3.88)	2.54(0.42)
ko	Baseline	59.60(0.67)	27.00(1.00)	61.45(24.90)	5.78(3.10)
	3-shot	47.55(2.35)	32.00(5.00)	46.99(16.06)	6.35(4.80)
	Greedy decoding	58.92(1.34)	26.00(1.00)	58.23(21.29)	4.94(3.10)
	ITLC	57.25(2.01)	31.00(3.00)	60.64(26.10)	5.92(4.51)
	LCG	53.25(0.00)	20.00(0.00)	38.96(2.01)	2.40(0.71)
	LEAD (ours)	39.85(0.33)	19.00(0.00)	30.12(1.20)	1.27(0.14)
ja	Baseline	59.25(–)	32.00(–)	59.92(–)	5.22(–)
	3-shot	51.23(–)	33.00(–)	41.74(–)	6.06(–)
	Greedy decoding	59.25(–)	32.00(–)	59.09(–)	4.09(–)
	ITLC	56.59(–)	33.00(–)	61.16(–)	6.91(–)
	LCG	53.55(–)	30.00(–)	49.17(–)	1.41(–)
	LEAD (ours)	42.85(–)	22.00(–)	33.47(–)	0.28(–)
hi	Baseline	44.86(0.00)	22.00(0.00)	– (–)	1.83(0.28)
	3-shot	31.82(0.00)	21.00(0.00)	– (–)	1.13(0.00)
	Greedy decoding	42.84(0.00)	22.00(0.00)	– (–)	0.56(0.14)
	ITLC	37.83(0.00)	20.00(0.00)	– (–)	1.27(0.14)
	LCG	41.18(0.00)	18.00(0.00)	– (–)	0.42(0.00)
	LEAD (ours)	36.82(0.00)	16.00(0.00)	– (–)	0.71(0.00)
ru	Baseline	45.24(0.00)	12.00(0.00)	58.20(2.34)	3.10(0.56)
	3-shot	36.87(0.34)	10.00(0.00)	33.98(1.56)	2.82(0.42)
	Greedy decoding	44.23(0.00)	8.00(0.00)	53.91(2.73)	2.82(0.00)
	ITLC	42.23(0.00)	10.00(0.00)	59.38(3.52)	2.96(0.14)
	LCG	35.88(0.00)	8.00(0.00)	44.14(0.00)	0.56(0.00)
	LEAD (ours)	31.19(0.00)	3.00(0.00)	30.08(0.00)	0.56(0.00)

Table 16: Latin (CJ) confusion rate on Qwen3-8B. Values are presented as Latin (CJ) in %.

Lang.	Method	LCB (cross-lingual)	LCB (monolingual)	CrossSum (cross-lingual)	FLORES-NO-LATIN (cross-lingual)
zh	Baseline	76.17(92.30)	98.00(99.40)	95.20(98.79)	98.73(99.29)
	3-shot	81.85(93.82)	37.50(97.46)	97.60(99.20)	98.73(99.29)
	Greedy decoding	75.51(93.26)	96.00(100.00)	93.20(98.36)	98.73(99.58)
	ITLC	82.20(92.78)	96.50(99.43)	94.80(99.19)	98.73(99.58)
	LCG	75.50(92.21)	98.00(98.82)	93.20(99.59)	99.29(99.44)
	LEAD (ours)	81.87(94.26)	99.00(99.42)	94.80(98.79)	99.72(99.86)
ar	Baseline	77.85(94.53)	95.33(99.32)	55.81(98.05)	89.42(99.86)
	3-shot	83.40(95.84)	96.33(99.33)	70.16(99.22)	89.14(99.58)
	Greedy decoding	76.51(94.54)	97.33(100.00)	60.08(98.05)	92.67(99.86)
	ITLC	83.21(95.81)	95.33(99.00)	55.43(98.45)	89.56(99.72)
	LCG	80.86(95.25)	99.00(100.00)	81.78(99.22)	96.05(100.00)
	LEAD (ours)	81.54(94.26)	97.67(99.67)	82.95(97.67)	95.91(99.72)
ko	Baseline	71.16(95.11)	94.00(98.99)	59.04(98.79)	90.97(98.87)
	3-shot	82.21(95.77)	93.00(100.00)	71.89(99.20)	90.13(98.87)
	Greedy decoding	73.84(95.89)	94.00(98.98)	64.26(98.80)	92.24(98.87)
	ITLC	78.86(93.98)	94.00(98.99)	59.04(98.39)	90.41(99.29)
	LCG	75.50(94.02)	97.00(100.00)	84.74(99.60)	96.33(99.15)
	LEAD (ours)	82.55(95.95)	97.00(100.00)	95.98(99.60)	98.73(99.86)
ja	Baseline	77.84(93.10)	94.00(96.91)	86.36(95.45)	96.19(97.88)
	3-shot	81.53(93.12)	92.00(96.94)	91.74(95.38)	94.92(96.47)
	Greedy decoding	77.16(90.86)	97.00(98.98)	87.19(95.45)	96.61(98.17)
	ITLC	81.18(90.72)	95.00(97.96)	87.60(92.15)	94.36(96.47)
	LCG	78.52(92.33)	92.00(96.97)	90.91(96.64)	98.87(99.15)
	LEAD (ours)	84.89(95.22)	94.00(97.98)	97.52(99.59)	99.72(99.86)
hi	Baseline	83.21(94.53)	97.00(100.00)	– (–)	97.88(99.86)
	3-shot	88.57(98.35)	97.00(97.98)	– (–)	98.87(99.86)
	Greedy decoding	84.21(95.95)	98.00(99.00)	– (–)	98.87(100.00)
	ITLC	86.57(97.25)	99.00(100.00)	– (–)	99.01(99.86)
	LCG	83.21(96.92)	97.00(100.00)	– (–)	98.87(100.00)
	LEAD (ours)	88.24(94.92)	97.00(100.00)	– (–)	99.44(100.00)
ru	Baseline	80.53(94.70)	96.00(98.00)	87.50(99.22)	96.90(99.72)
	3-shot	86.56(96.76)	98.00(98.00)	89.84(99.61)	97.60(99.72)
	Greedy decoding	80.53(94.43)	98.00(97.96)	86.72(100.00)	98.03(99.44)
	ITLC	84.88(94.72)	98.00(98.00)	82.03(98.44)	97.46(99.29)
	LCG	83.20(96.71)	98.00(99.00)	91.80(100.00)	99.58(100.00)
	LEAD (ours)	86.89(96.03)	100.00(100.00)	94.53(99.61)	99.58(99.86)

Table 17: Response Pass Rate (RPR) and Word Pass Rate (WPR) on Qwen3-8B. Values are presented as RPR (WPR) in %.

Lang.	Method	LCB (cross-lingual)	LCB (monolingual)	CrossSum (cross-lingual)	FLORES-NO-LATIN (cross-lingual)
zh	Baseline	54.55(–)	32.50(–)	60.80(–)	0.99(–)
	3-shot	57.58(–)	32.00(–)	34.00(–)	0.71(–)
	Greedy decoding	58.57(–)	34.50(–)	60.00(–)	0.85(–)
	ITLC	56.23(–)	29.50(–)	44.00(–)	1.41(–)
	LCG	57.90(–)	31.50(–)	55.60(–)	0.42(–)
	LEAD (ours)	54.88(–)	31.50(–)	50.00(–)	0.28(–)
ar	Baseline	39.51(0.00)	20.00(0.00)	30.23(2.33)	3.10(0.28)
	3-shot	31.80(0.00)	22.67(0.00)	12.02(0.00)	2.96(0.00)
	Greedy decoding	35.82(0.00)	21.00(0.00)	33.72(1.16)	1.97(0.28)
	ITLC	30.12(0.67)	12.00(0.33)	15.89(1.55)	2.54(0.00)
	LCG	36.49(0.00)	20.33(0.00)	16.67(0.00)	0.28(0.00)
	LEAD (ours)	29.79(0.00)	12.33(0.00)	10.47(0.00)	0.42(0.00)
ko	Baseline	50.54(0.34)	26.00(0.00)	56.22(12.05)	2.12(0.56)
	3-shot	48.85(1.68)	26.00(1.00)	20.48(7.63)	1.55(0.28)
	Greedy decoding	53.21(1.00)	30.00(1.00)	56.22(12.05)	1.97(0.42)
	ITLC	49.85(0.67)	29.00(1.00)	24.90(3.21)	0.56(0.56)
	LCG	43.18(0.67)	26.00(0.00)	37.75(4.82)	0.71(0.14)
	LEAD (ours)	40.84(0.00)	18.00(0.00)	32.53(4.42)	0.28(0.00)
ja	Baseline	53.90(–)	31.00(–)	59.50(–)	1.55(–)
	3-shot	51.88(–)	35.00(–)	34.30(–)	1.27(–)
	Greedy decoding	53.89(–)	32.00(–)	62.81(–)	0.56(–)
	ITLC	49.20(–)	31.00(–)	33.06(–)	1.27(–)
	LCG	45.86(–)	29.00(–)	48.76(–)	0.28(–)
	LEAD (ours)	43.19(–)	24.00(–)	39.26(–)	0.56(–)
hi	Baseline	36.15(0.00)	23.00(0.00)	– (–)	0.71(0.00)
	3-shot	29.48(0.00)	17.00(0.00)	– (–)	0.14(0.00)
	Greedy decoding	36.83(0.00)	16.00(0.00)	– (–)	0.56(0.00)
	ITLC	22.07(0.00)	17.00(0.00)	– (–)	0.28(0.00)
	LCG	30.13(0.00)	17.00(0.00)	– (–)	0.14(0.00)
	LEAD (ours)	23.77(0.00)	12.00(0.00)	– (–)	0.56(0.00)
ru	Baseline	36.53(0.00)	7.00(0.00)	58.98(0.78)	1.55(0.00)
	3-shot	34.53(0.00)	13.00(0.00)	21.09(0.00)	1.41(0.00)
	Greedy decoding	34.53(0.00)	9.00(0.00)	58.20(0.39)	1.27(0.00)
	ITLC	30.15(0.00)	5.00(0.00)	32.81(0.00)	0.85(0.00)
	LCG	30.19(0.00)	6.00(0.00)	40.62(0.00)	0.28(0.00)
	LEAD (ours)	29.18(0.00)	6.00(0.00)	40.62(0.78)	0.28(0.00)

Table 18: Latin (CJ) confusion rate on Qwen3-30B. Values are presented as Latin (CJ) in %.

Lang.	Method	LCB (cross-lingual)	LCB (monolingual)	CrossSum (cross-lingual)	FLORES-NO-LATIN (cross-lingual)
zh	Baseline	81.89(94.79)	95.00(98.24)	93.20(99.59)	99.15(99.29)
	3-shot	77.83(94.06)	38.50(100.00)	97.20(98.80)	99.58(99.72)
	Greedy decoding	80.88(94.41)	95.50(99.38)	90.80(98.79)	99.44(99.58)
	ITLC	84.57(95.89)	98.00(100.00)	93.60(99.19)	98.73(99.15)
	LCG	81.55(94.94)	94.00(99.37)	91.20(99.19)	99.72(100.00)
	LEAD (ours)	82.90(97.05)	95.00(99.43)	91.60(100.00)	99.72(99.86)
ar	Baseline	88.58(96.86)	98.00(99.33)	77.91(97.67)	96.05(99.72)
	3-shot	89.92(96.92)	96.67(99.66)	94.19(99.22)	96.47(99.58)
	Greedy decoding	88.24(97.56)	98.33(99.33)	82.17(98.45)	97.18(99.86)
	ITLC	88.26(97.89)	97.67(100.00)	89.15(100.00)	96.47(99.72)
	LCG	88.59(97.97)	99.00(99.67)	89.53(99.61)	98.31(100.00)
	LEAD (ours)	91.93(97.29)	99.33(99.66)	94.96(100.00)	99.86(100.00)
ko	Baseline	84.55(95.72)	93.00(98.98)	75.90(99.17)	97.46(99.86)
	3-shot	84.90(96.62)	96.00(100.00)	86.75(100.00)	98.59(100.00)
	Greedy decoding	83.22(95.52)	94.00(98.99)	77.51(100.00)	97.60(99.86)
	ITLC	86.83(96.19)	93.00(100.00)	93.98(100.00)	99.15(100.00)
	LCG	83.89(96.29)	96.00(98.99)	85.94(100.00)	99.15(100.00)
	LEAD (ours)	87.57(94.38)	95.00(100.00)	91.16(99.18)	99.72(100.00)
ja	Baseline	83.20(91.28)	99.00(98.98)	90.08(94.63)	98.59(99.15)
	3-shot	82.19(93.60)	97.00(98.98)	96.69(99.17)	99.01(99.58)
	Greedy decoding	80.86(90.94)	97.00(98.99)	89.26(95.04)	99.58(99.58)
	ITLC	83.22(90.78)	96.00(99.00)	95.45(97.11)	99.15(99.29)
	LCG	81.21(92.05)	94.00(97.96)	93.39(97.92)	99.58(100.00)
	LEAD (ours)	88.58(96.21)	97.00(98.99)	96.69(99.58)	99.29(99.86)
hi	Baseline	88.25(96.07)	95.00(95.96)	– (–)	99.01(99.72)
	3-shot	88.24(96.73)	98.00(99.00)	– (–)	99.86(99.86)
	Greedy decoding	88.58(96.15)	96.00(96.97)	– (–)	99.44(99.72)
	ITLC	94.29(97.89)	99.00(100.00)	– (–)	100.00(100.00)
	LCG	89.92(96.34)	98.00(98.99)	– (–)	99.72(100.00)
	LEAD (ours)	91.59(96.57)	97.00(98.00)	– (–)	99.44(99.72)
ru	Baseline	84.23(94.28)	99.00(99.00)	88.67(99.22)	99.01(99.58)
	3-shot	85.55(96.27)	99.00(99.00)	96.88(99.61)	99.15(99.58)
	Greedy decoding	85.23(94.20)	97.00(97.00)	87.50(99.21)	99.29(99.72)
	ITLC	87.24(96.74)	99.00(100.00)	93.36(100.00)	99.58(100.00)
	LCG	88.24(93.81)	99.00(99.00)	94.92(99.61)	99.86(99.86)
	LEAD (ours)	88.91(94.03)	99.00(99.00)	93.36(100.00)	99.86(99.86)

Table 19: Response Pass Rate (RPR) and Word Pass Rate (WPR) on Qwen3-30B. Values are presented as RPR (WPR) in %.