Enhancing Low-Resource Language Reasoning via High-Resource Language Feature Transfer

Minju Song[1] **Hyeon Hwang**[1] **Junhyun Lee**[2,3*] **Jaewoo Kang**[1,4*]
Korea University[1] Hankuk University of Foreign Studies[2]
Noah's Farm[3] AIGEN Sciences[4]
{minjusong, hyeon-hwang, kangj}@korea.ac.kr junhyun.lee@hufs.ac.kr

Abstract

Large language models exhibit substantial performance variation across languages, even when solving semantically equivalent tasks. Existing analyses often treat this phenomenon as an observational disparity caused by differences in pretraining data, tokenization, or benchmark coverage. We study a complementary hypothesis: high-resource languages (HRLs) may more reliably elicit latent computations useful for task-specific (i.e. mathematical) reasoning, while lower-resource languages (LRLs) may under-activate those computations despite expressing the same task. To test this hypothesis, we introduce a mechanistic intervention framework for identifying and transferring task-relevant sparse latent features across languages. Using sparse autoencoders over residual-stream activations, we isolate features enriched in successful HRL task-specific reasoning while filtering out source-language and generic-generation features. We then construct steering directions from these features and inject them during LRL inference. The resulting interventions test whether the selected features are functionally involved in the observed reasoning gap: suppressing them should impair source-language reasoning, while activating them should partially recover target-language reasoning beyond random and non-task controls. Our framework reframes some cross-lingual reasoning gaps as failures of mechanism elicitation rather than capability absence, and offers a causally testable route to feature-mediated transfer without translation, fine-tuning, or changing the user-facing language.

1 Introduction

Large language models (LLMs) have become increasingly capable multilingual systems (Xue et al., 2021; Workshop et al., 2023), yet their reasoning abilities remain uneven across languages (Bang et al., 2023; Zhu et al., 2024; Kang et al., 2026). For instance, a model that reliably solves a mathematical problem in a high-resource language (HRL), such as English or Spanish, may fail on an equivalent problem expressed in Thai, Bengali, Swahili, or another low-resource language (LRL). This gap is often attributed to data imbalance: HRLs are better represented during pretraining, instruction tuning, and evaluation, leading to more robust behavior in those languages (Joshi et al., 2020; Nguyen et al., 2023). For example, English alone makes up

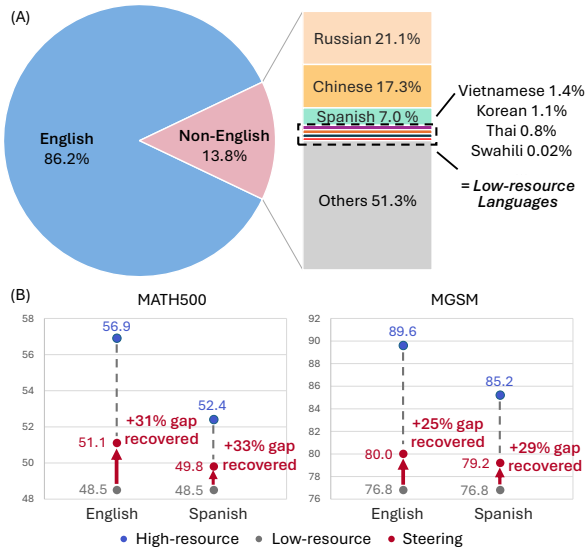


Figure 1: (A) In pretraining corpora FineWeb (Penedo et al., 2024), English dominates the pretraining corpus (86.2%), while target low-resource languages (Vietnamese, Korean, Thai, Swahili) together account for under 4%. (B) On MATH500 and MGSM, Gemma-2-9B-it scores substantially higher when prompted in a high-resource reference language (blue: English or Spanish) than in low-resource targets (grey: averaged over Thai/Korean/Vietnamese for MATH500, and Thai/Korean/Swahili for MGSM). Steering with contrastively selected HRL reasoning features (red) recovers 25–33% of the gap across all four (reference, benchmark) settings, without training or translation.

*Corresponding author

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 48447–48461
October 24-29, 2026 ©2026 Association for Computational Linguistics

most of the pretraining data while low-resource languages get only a tiny share, and this gap shows up in their reasoning accuracy too (Figure 1). While important, this explanation leaves a deeper question unresolved: *Does language merely change the surface form of an input, or can it change which internal reasoning mechanisms a model uses?*

We study the hypothesis that language can modulate the elicitation of latent computation. Under this view, semantically equivalent prompts in different languages need not merely differ in surface form; they may induce different patterns of internal feature activation. An HRL may more reliably elicit computations useful for a task, such as symbolic decomposition or stepwise arithmetic reasoning, whereas an LRL may express the same problem while eliciting those computations more weakly, at different positions, or not at all (Wendler et al., 2024; Etxaniz et al., 2024). If this hypothesis is correct, cross-ingual performance gaps reflect not only differences in input distribution, but also differences in the accessibility of task-associated internal mechanisms.

We frame this problem as one of mechanistic intervention (Elhage et al., 2021; Meng et al., 2022; Tang et al., 2024; Li et al., 2023). Rather than asking only whether a model performs better in one language than another, we ask whether task-relevant latent features activated during successful HRL reasoning can be identified, tested, and reused during LRL inference. This perspective turns cross-lingual performance gaps into a causal question: *which internal computations are activated, suppressed, or transferable across languages?*

To answer this question, we propose a sparse-feature intervention framework for cross-lingual reasoning. Using sparse autoencoders (SAEs) (Ng et al., 2011; Chalnev et al., 2024; Templeton et al., 2024), we represent model activations in an interpretable latent basis and construct task-specific feature libraries from successful HRL reasoning traces. Crucially, feature discovery is contrastive: selected features must be enriched during successful task execution while being less characteristic of generic HRL behavior. This helps distinguish reusable task mechanisms from features that merely encode language identity or broad generation patterns.

At inference time, we use the selected sparse features to construct a residual-stream steering direction from activation differences between HRL and LRL reasoning traces. The resulting vector is injected during LRL inference, leaving the prompt prefill unchanged while amplifying computations that are more strongly expressed in successful HRL reasoning. This allows us to test whether features associated with HRL success can causally improve LRL inference.

Our evaluation separates observational feature discovery from causal validation (Pearl, 2001). The contrastive procedure identifies candidate feature sets associated with successful reference-language (HRL) reasoning, but this association alone does not establish causality. We therefore evaluate the selected features through interventions in a fixed trained model: if the features are functionally involved in the relevant computation, activating them should improve target-language (LRL) inference, suppressing them should impair reference-language inference, and matched random or excluded-feature controls should not produce the same pattern. This design does not estimate a formal natural indirect effect, but it provides interventional evidence that the selected features participate in a mechanism consistent with partial cross-lingual transfer. The resulting method requires no parameter updates, no translation at inference time, and no change to the user-facing language, though feature discovery uses benchmark correctness signals.

2 Preliminaries

2.1 Sparse Autoencoders (SAEs)

SAEs provide a sparse decomposition of transformer activations into latent features. Given a residual-stream activation $h \in \mathbb{R}^d$ from a decoder-only language model, an SAE with F latent features encodes h into a sparse latent representation

$$z = \mathrm{Enc}(h) = \phi\left(W_{\mathrm{enc}}(h - b_{\mathrm{dec}}) + b_{\mathrm{enc}}\right) \in \mathbb{R}^F, \tag{1}$$

where $W_{\mathrm{enc}} \in \mathbb{R}^{F \times d}$, $b_{\mathrm{enc}} \in \mathbb{R}^F$, $b_{\mathrm{dec}} \in \mathbb{R}^d$, and $\phi(\cdot)$ is a sparsity-inducing nonlinearity. The latent code is reconstructed through a linear decoder,

$$\hat{h} = \mathrm{Dec}(z) = \sum_{f=1}^{F} z_f w_{\mathrm{dec}}^{(f)} + b_{\mathrm{dec}}, \tag{2}$$

where $w_{\mathrm{dec}}^{(f)} \in \mathbb{R}^d$ denotes the f-th column of the decoder matrix $W_{\mathrm{dec}} \in \mathbb{R}^{d \times F}$. Each decoder column corresponds to the residual-stream direction associated with latent feature f, and the coefficient z_f determines the contribution of that feature to the reconstruction.

2.2 Problem Setup and Notation

Let M be a decoder-only language model and let ℓ denote a residual-stream layer of interest. We attach a pretrained SAE to layer ℓ and analyze its latent representations during chain-of-thought (CoT) reasoning. We consider a reference-language (HRL) b and a target-language (LRL) a.

Given a reasoning benchmark $\mathcal{D}$, the model produces a reasoning trace $r_i^{(k)}$ of length $T_i^{(k)}$ for problem $i \in \mathcal{D}$ in language $k \in \{a, b\}$. For each generation token position $t \in \{1, \ldots, T_i^{(k)}\}$, let $h_{i,t}^{(k)} \in \mathbb{R}^d$ denote the residual-stream activation at layer ℓ and $z_{i,t}^{(k)} = \mathrm{Enc}(h_{i,t}^{(k)}) \in \mathbb{R}^F$ its SAE representation, with $z_{i,t,f}^{(k)}$ being the activation of latent feature f at that token.

Our goal is to identify latent features associated with successful reasoning in the reference language b and use them to steer the model toward similar behavior when solving problems in the target language a.

2.3 Interventional Causal Effects

Our causal claims are restricted to interventions within a fixed trained model. Feature selection from naturally generated traces is observational: it identifies candidate internal variables associated with successful reasoning, but does not by itself establish causality.

We therefore evaluate causal involvement by directly modifying residual-stream activations during the forward pass. For an internal activation $h_{\ell,t}$ and an intervention vector Δh, we write $Y(do(h_{\ell,t} \leftarrow h_{\ell,t} + \Delta h))$ for the model's outcome under the intervention, where Y denotes final-answer correctness or a logit-based answer score.

A feature set is treated as *sufficient* if activating its associated residual direction improves target-language (LRL) reasoning, *necessary* if suppressing it degrades reference-language (HRL) reasoning, and *specific* if matched random or excluded feature interventions do not produce the same effect. We use the terms partial sufficiency and functional necessity operationally. A positive activation effect suggests that the corresponding residual directions are partially sufficient under the specified intervention policy; a negative suppression effect suggests functional dependence on the evaluated distribution. These terms do not imply logical sufficiency or necessity in the counterfactual-cause sense.

3 Method

This section details our method in two parts. First, we describe how we identify task-relevant sparse latent features through a contrastive comparison between reasoning traces of a high-resource reference language and a low-resource target language (LRL). Second, using those features, we intervene on the low-resource forward pass at a single residual stream layer to transfer reasoning behavior from the reference language.

3.1 Task-Relevant Sparse Latent Feature Identification.

Paired contrast set and per-token features. Let $\mathcal{D}$ be a reasoning benchmark on which the model produces a chain-of-thought response $r_i^{(k)}$ for each problem i and language k, scored as correct or incorrect. To isolate features that drive successful reasoning in a high-resource reference language b but fail to engage in a low-resource target language (LRL) a, we restrict attention to the set of paired problems on which the reference succeeds and the target fails on the same question:

$$\mathcal{P}_{b \to a} = \left\{ i \;\middle|\; r_i^{(b)} \text{ correct} \wedge r_i^{(a)} \text{ incorrect} \right\}. \quad (3)$$

For each $i \in \mathcal{P}_{b \to a}$ and each side $k \in \{a, b\}$, we record the identity of the strongest-activating feature at every generation token of the corresponding trace,

$$f_{i,t}^{*(k)} = \arg \max_{f \in \{1, \ldots, F\}} z_{i,t,f}^{(k)}, \quad (4)$$

and define the per-response feature set $\mathcal{F}_i^{(k)} = \{ f_{i,t}^{*(k)} : t = 1, \ldots, T_i^{(k)} \}$ as the set of distinct top features across all generation tokens of trace $r_i^{(k)}$.

Set-difference candidate pool. For each feature f and each side $k \in \{a, b\}$, we count the number of contrast responses in which it appears as the per-token argmax:

$$c_f^{(k)} = \left| \{ i \in \mathcal{P}_{b \to a} : f \in \mathcal{F}_i^{(k)} \} \right|. \quad (5)$$

The contrastive candidate pool is then defined as the set-difference between the reference and target sides, retaining features that appear in the reference-correct responses but are absent from the matched target-incorrect ones:

$$\mathcal{C}_{\text{cand}} = \left\{ f \;\middle|\; c_f^{(b)} \geq 1 \wedge c_f^{(a)} = 0 \right\}, \quad (6)$$

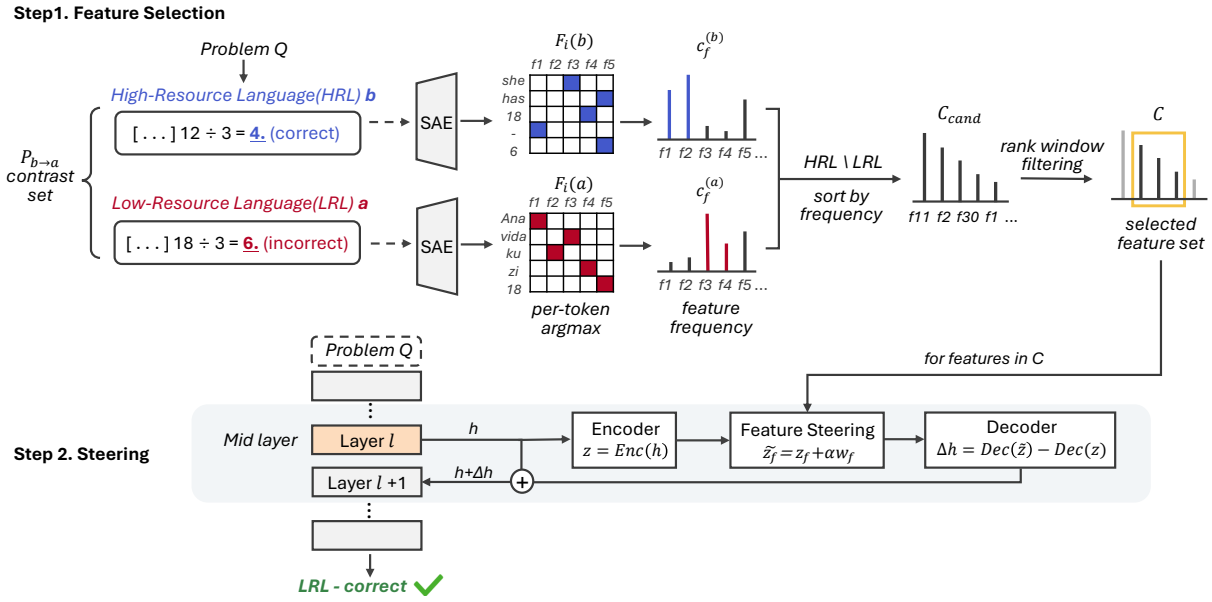


Figure 2: **Method overview. Step 1.** We collect the paired contrast set $\mathcal{P}_{b\to a}$ of problems on which the model is correct in the reference language b but incorrect in the target language a. Encoding both traces through the SAE and counting the per-token argmax feature across the set gives per-side frequencies $c_f^{(b)}, c_f^{(a)}$; their set-difference followed by rank-window filtering yields the selected feature set $\mathcal{C}$. **Step 2.** At inference time on target-language inputs, the layer-ℓ residual stream h is encoded to z, the features in $\mathcal{C}$ are bumped by αw_f where $w_f = \bar{z}_f^{(b)} - \bar{z}_f^{(a)}$, and the decoded difference Δh is added back to h at generation tokens only.

ranked in descending order of $c_f^{(b)}$ so that higher ranks correspond to features that recur consistently across reference-correct traces.

Rank-window filtering. Features with the largest values in $\mathcal{C}_{\text{cand}}$ often correspond to highly frequent generation patterns that appear across reasoning traces regardless of their underlying reasoning content, providing limited signal for identifying reasoning-relevant latent dimensions. To reduce their influence, we retain only features whose ranks fall within a predefined window (s, n), where $0 \le s < n$:

$$\mathcal{C} = \left\{ \mathcal{C}_{\text{cand}}^{(s+1)}, \mathcal{C}_{\text{cand}}^{(s+2)}, \ldots, \mathcal{C}_{\text{cand}}^{(n)} \right\}. \tag{7}$$

where $\mathcal{C}_{\text{cand}}^{(r)}$ denotes the feature at rank r in the sorted candidate list.

3.2 Transfer via Steering at the Residual Stream

Per-feature steering coefficients. For each language $k \in \{a, b\}$, let $\mathcal{D}_k^+$ denote the set of correctly solved problems. We compute the mean SAE activation vector over all generated tokens from correctly solved reasoning traces:

$$\bar{z}^{(k)} = \frac{1}{N_k} \sum_{i \in \mathcal{D}_k^+} \sum_{t=1}^{T_i^{(k)}} z_{i,t}^{(k)} \in \mathbb{R}^F, \tag{8}$$

where $N_k = \sum_{i \in \mathcal{D}_k^+} T_i^{(k)}$ is the total number of generated tokens across all correctly solved traces in language k, and $z_{i,t}^{(k)}$ is the SAE representation defined in §3.1. For each selected feature $f \in \mathcal{C}$, we define its steering coefficient as the difference in mean activation between the reference language b and the target language a:

$$w_f = \bar{z}_f^{(b)} - \bar{z}_f^{(a)}. \tag{9}$$

Intuitively, w_f measures how much more strongly feature f is expressed in the reference language relative to the target language on successful reasoning traces.

Steering intervention. Let $h \in \mathbb{R}^d$ denote the layer-ℓ residual stream activation at a generation token, and let $z = \text{Enc}(h) \in \mathbb{R}^F$ be its SAE representation. We construct a modified latent code by shifting the selected features according to their steering coefficients, where $\alpha > 0$ controls the steering strength:

$$\tilde{z}_f = \begin{cases} z_f + \alpha w_f, & f \in \mathcal{C}, \\ z_f, & \text{otherwise}, \end{cases} \tag{10}$$

The resulting intervention in residual space is given by

$$\Delta h = \mathrm{Dec}(\tilde{z}) - \mathrm{Dec}(z) = \alpha \sum_{f \in \mathcal{C}} w_f \, W_{\mathrm{dec}}[f], \tag{11}$$

where the final equality follows from the linearity of the decoder. We register this intervention as a forward hook on the output of layer ℓ and apply it only to generation tokens, leaving the prompt prefill unchanged. The same steering vector Δh is added at every generation step across all problems.

4 Experiments

4.1 Experimental Setup

We evaluate two instruction-tuned language models (Gemma Team, 2024; Qwen et al., 2025) paired with pretrained layer-20 Sparse Autoencoders (SAEs). For `gemma-2-9b-it`, we use the canonical Gemma-Scope SAE (Lieberum et al., 2024) (`width_16k`; $F = 16{,}384$). For `Qwen2.5-7B-Instruct`, we use the released layer-20 matryoshka SAE [1] (`lmsys/matryoshka/k-100`; $F = 65{,}536$, $k = 100$). Both SAEs employ the JumpReLU architecture (Rajamanoharan et al., 2024). All steering interventions are applied at layer-20 residual stream.

4.2 Datasets and Languages

We evaluate on three multilingual reasoning benchmarks: **MATH500** (Hendrycks et al., 2021; Lightman et al., 2023) ($N = 311$), the MATH500 split of the multilingual MMATH benchmark (Luo et al., 2025), requiring multi-step symbolic reasoning; **MGSM** (Shi et al., 2022) ($N = 250$), a multilingual benchmark of grade-school mathematical word problems; and **MMLU-ProX (Psychology)** (Xuan et al., 2025) ($N = 798$), a 10-way multiple-choice benchmark covering undergraduate-level psychology questions.

Each problem is presented in a single language using a native-language chain-of-thought (Wei et al., 2022) prompt. We consider English (en) and Spanish (es) as high-resource reference languages (HRL), and Korean (ko), Thai (th), Swahili (sw), and Vietnamese (vi) as target languages (LRL). Depending on the benchmark, translations are obtained from publicly available multilingual benchmark releases.

4.3 Feature Selection and Steering

For each reference–target language pair and dataset, we construct a contrastive feature pool using the procedure described in Section 3.1. Candidate features are identified from problems that are solved correctly in the reference language but incorrectly in the target language. Following rank-window filtering, we discard the top 50% of candidate features ranked by occurrence count and retain features between the 50th and 90th percentiles. Only features that survive this filtering stage are used for steering.

During inference, we apply residual-stream steering at every generation step while leaving prompt-prefill activations unchanged.

4.4 Evaluation Metrics

Accuracy. Accuracy is computed using the native evaluation protocol of each benchmark. For MATH500 and MGSM, answers are evaluated using `math_verify`; for MMLU-ProX, predictions are scored by matching the final selected answer option.

Recovery Rate. To quantify how much steering closes the performance gap between a target language and a reference language, we define the recovery rate

$$\mathrm{Recovery}_{a \to b} = \frac{\mathrm{acc}_{a \to b} - \mathrm{acc}_a}{\mathrm{acc}_b - \mathrm{acc}_a}, \tag{12}$$

where acc_a denotes the baseline accuracy of target language a, acc_b denotes the accuracy of reference language b, and $\mathrm{acc}_{a \to b}$ denotes the steered target-language accuracy.

5 Discussion

5.1 Causal Evidence for Feature-Mediated Transfer

The contrastive feature selection procedure in Section 3 identifies sparse features that are more salient in successful reference-language traces than in matched target-language failures. We treat the selected set $\mathcal{C}$ as a set of candidate intervention features: task-associated features that may transfer task-relevant computations from the reference language reasoning to the target language. We evaluate this interpretation with three intervention tests.

[1] https://huggingface.co/chanind/qwen2.5-7B-it-layer-20-saes

Reference language: English														
Model		MATH500				MGSM				MMLU-ProX (psychology)				
		en	th	ko	vi	en	th	ko	sw	en	th	ko	sw	vi
Gemma-2-9B	Baseline	56.91	48.23	49.20	48.23	89.60	78.40	76.40	75.60	57.27	32.71	17.67	8.27	40.73
	+Steering		49.20	52.41	51.77		82.40	78.40	79.20		33.30	23.31	10.30	42.36
	Recovery		+11%	+42%	+41%		+36%	+15%	+26%		+2%	+14%	+4%	+10%
Qwen2.5-7B	Baseline	73.63	60.77	64.95	63.67	95.20	80.40	79.60	15.20	59.90	39.35	42.48	15.29	49.50
	+Steering		63.70	66.60	66.56		82.00	80.00	19.60		41.60	44.11	17.42	53.13
	Recovery		+23%	+19%	+29%		+11%	+3%	+6%		+11%	+9%	+5%	+35%
Reference language: Spanish														
Model		MATH500				MGSM				MMLU-ProX (psychology)				
		es	th	ko	vi	es	th	ko	sw	es	th	ko	sw	vi
Gemma-2-9B	Baseline	52.41	48.23	49.20	48.23	85.20	78.40	76.40	75.60	45.49	32.71	17.67	8.27	40.73
	+Steering		48.87	50.16	50.48		82.40	78.00	77.20		33.50	19.70	10.00	41.35
	Recovery		+15%	+30%	+54%		+59%	+18%	+17%		+6%	+7%	+5%	+13%
Qwen2.5-7B	Baseline	67.85	60.77	64.95	63.67	84.00	80.40	79.60	15.20	53.01	39.35	42.48	15.29	49.50
	+Steering		63.34	65.90	66.56		82.00	82.00	19.60		41.10	43.61	15.54	51.38
	Recovery		+36%	+33%	+69%		+44%	+55%	+6%		+13%	+11%	+1%	+54%

Table 1: **Main results: multilingual reasoning accuracy and cross-lingual gap closure.** Accuracy (%) on MATH500, MGSM, and MMLU-ProX (psychology subset) for Gemma-2-9B-it and Qwen2.5-7B-Instruct, with English (top) and Spanish (bottom) as reference languages. For each (reference, target) pair, **Baseline** reports vanilla CoT accuracy, **+Steering** reports accuracy after applying our residual-stream intervention at layer 20, and **Recovery** is the fraction of the reference gap closed, Reference-language scores (en, es columns of the Baseline row) serve as ceilings and are not steered. Target language coverage varies by benchmark availability.

Method	**EN**	**TH**	**KO**	**VI**
Baseline	56.91	48.23	49.20	48.23
Ours	–	49.20	52.41	51.77
Top-k feature steering	–	45.98	45.66	47.91
Random steering	–	47.80	46.84	52.30
Negative steering vector	–	46.62	47.27	46.95
HRL ablation	–	54.02	50.16	52.09
HRL ablation (random)	–	55.16	55.09	55.52

Table 2: **Control and ablation experiments.** We evaluate the contrastively selected feature set $\mathcal{C}$ against alternative steering choices and against a source-side suppression test. **Ours**: steering with $\mathcal{C}$ (target-language accuracy). **Top-k feature steering**: steering with the most frequent features in $\mathcal{C}_{\text{cand}}$, skipping rank-window filtering. **Random steering**: steering with a size-matched random feature set. **Negative steering vector**: applying $-\Delta h$ instead of $+\Delta h$. **HRL ablation**: suppressing $\mathcal{C}_{\text{EN}\to a}$ during English inference (rows labeled by target language a, but the score is on English). All target-language steering results use Gemma-2-9B-it on MATH500 with English as reference.

Partial sufficiency. For a target language a, we test whether activating a feature set $\mathcal{S}$ improves target-language reasoning:

$$\tau_S^a(\alpha) = \mathbb{E}_i\left[Y_i^a\big(\text{do}(h_{\ell,t} \leftarrow h_{\ell,t} + \Delta h_S^+(\alpha))\big) - Y_i^a\right], \quad (13)$$

where Y_i^a denotes final-answer correctness or a logit-based answer score. When $\mathcal{S}$ is instantiated as the contrastively selected feature set $\mathcal{C}$, a positive value of $\tau_C^a(\alpha)$ indicates that activating these features improves target-language reasoning. Table 1 provides this target-side test: activating the pair-specific feature set $\mathcal{C}_{\text{EN}\to a}$ improves accuracy over the corresponding LRL baseline for Thai, Korean, and Vietnamese.

These gains are largest for Korean and Vietnamese, indicating that the selected directions can partially recover reasoning behavior that is more reliably elicited in the English reference language.

Importantly, these gains do not generally arise from switching the generated output to the reference language. Automatic language identification with GlotLID (Kargaran et al., 2023) shows that steering largely preserves target-language generation; detailed results across models, datasets, and reference–target pairs are reported in Appendix B (Table 4).

Functional necessity. For a reference language b, we test whether suppressing the same features degrades reference-language reasoning:

$$\nu_S^b(\lambda) = \mathbb{E}_i\left[Y_i^b\Big(\text{do}(h_{\ell,t} \leftarrow h_{\ell,t} + \Delta h_{S,t}^-(\lambda))\Big) - Y_i^b\right], \quad (14)$$

where $\Delta h^{-}_{S,t}(\lambda)$ indicates an anti-steering perturbation effect. If $\nu^{b}_{C}(\lambda) < 0$, while matched random-feature suppression has a smaller effect, then the selected features are functionally involved in producing the reference-language behavior. The HRL ablation row in Table 2 instantiates this test on the source side: for each target language a, we suppress the same pair-specific feature set $\mathcal{C}_{\mathrm{EN}\to a}$ during English inference. Although the ablation is always performed in English, the Thai, Korean, and Vietnamese columns denote the target language used to construct $\mathcal{C}_{\mathrm{EN}\to a}$; the resulting drops from the English baseline show that these features are functionally involved in source-language reasoning.

Specificity. We compare $\mathcal{C}$ against matched random features $\mathcal{R}$ and excluded features $\mathcal{E}$. These controls test whether the observed effects are due to the selected task-associated features rather than arbitrary residual perturbation, intervention magnitude, or generic SAE feature manipulation. Table 2 shows that top-k frequent features and random steering do not reproduce the same consistent improvement pattern as the contrastively selected feature set. Moreover, applying the negative steering vector reduces target-language accuracy, suggesting that the signed HRL–LRL activation difference is functionally meaningful rather than merely increasing residual-stream activation magnitude.

Together, the three tests provide interventional evidence that $\mathcal{C}$ is functionally implicated in the cross-lingual reasoning gap: activating $\mathcal{C}$ improves target-language performance, suppressing $\mathcal{C}$ degrades reference-language performance, and matched controls fail to reproduce either effect. Table 2 directly operationalizes these criteria by evaluating the same pair-specific feature sets under target-side activation, source-side ablation, and control interventions. We therefore interpret the selected sparse features as a mechanistic handle through which target-language inference can access computations more reliably elicited in the reference language.

5.2 Are the steered features language-agnostic reasoning features?

To understand why injecting HRL-derived feature directions helps LRL reasoning, we ask what the steered features actually encode. We take the $(\mathrm{en}, \mathrm{ko})$ contrastive pool used for steering and, for each feature, examine which tokens it activates on in English versus Korean.

Feat.	Concept	English (peak)	Korean (peak)
13089	requirement	needs, needed (63.6)	필요 (28.0)
12672	quantity	number, many (51.0)	수, 몇 (23.5)
6224	aggregation	both, three (68.5)	모두 (17.9)
629	knowledge	know, understand (58.2)	알 (27.0)
6763	rate	per, / (69.7)	/ (21.8)
4404	construction	make, makes (46.3)	만들 (12.2)
4301	equality	equals, equation (53.8)	= (11.4)

Table 3: Cross-lingual reasoning features in the $(\mathrm{en}, \mathrm{ko})$ steering pool (`gemma-2-9b-it`, layer 20). For each feature we give the concept it encodes and its top-activating tokens in English and Korean reasoning traces (peak activation in parentheses).

Setup. For 40 MGSM problems answered correctly in English, we perform a single forward pass on both the English and Korean reasoning traces, extract the layer-20 residual-stream activations, and encode them using the Gemma-Scope SAE. For each of the 28 selected features, we then identify the top-activating tokens and record the peak activation in each language.

The features are largely language-agnostic. Only 3 of the 28 features are English-token specific and do not activate on Korean text (*of*, *the*, *to*); the remaining 25 activate in both languages. Among these, a semantically interpretable subset captures reasoning concepts shared across languages while activating the same SAE latent (Table 3).

For example, feature 13089 activates on English *needs/needed* and Korean 필요 ("need"), feature 12672 on *number/how many* and 수/몇, and feature 4301 on *equals/equation* and the symbol =. Other shared features correspond to syntactic categories, such as copulas, case particles, and prepositions, rather than task-specific reasoning content.

Interpretation. These observations help explain the steering result. If the features primarily captured English surface patterns, activating them during Korean inference would be unlikely to improve reasoning. Instead, they encode language-agnostic reasoning concepts shared across languages, such as quantity, requirement, equality, and aggregation. Moreover, these features typically activate *more weakly* on Korean tokens than on their English counterparts (e.g., feature 13089 peaks at 28.0 in Korean vs. 63.6 in English). This pattern is consistent with our hypothesis that the model underelicits task-relevant reasoning features in the low-resource language, and that steering can partially compensate for this activation gap.

6 Related Work

Multilingual reasoning and cross-lingual transfer. Large language models are increasingly multilingual, yet they reason far better in HRLs than in low-resource ones (Bang et al., 2023; Zhu et al., 2024; Kang et al., 2026). This gap is commonly attributed to the imbalance of pretraining and instruction-tuning data across languages (Joshi et al., 2020; Penedo et al., 2024), and prior remedies often rely on additional multilingual training (Zhang et al., 2025). A complementary line of work studies multilingualism mechanistically: multilingual transformers appear to process non-English inputs through a latent English-centric representation (Wendler et al., 2024), reasoning frequently improves when inputs are routed through English (Etxaniz et al., 2024), and specific neurons govern language-specific behaviour (Tang et al., 2024). These findings suggest cross-lingual gaps reflect not only the input distribution but also which internal mechanisms a model engages, motivating an intervention-based approach that requires no additional training.

Mechanistic interpretability and SAEs. Mechanistic interpretability studies LLM internals by decomposing models into interpretable units (Olah et al., 2020), often validating hypotheses through interventions such as activation patching and causal mediation analysis that localize behaviour to specific components (Meng et al., 2022; Wang et al., 2023; Syed et al., 2023). Sparse autoencoders (Ng et al., 2011; Bricken et al., 2023; Cunningham et al., 2023) decompose the residual stream of transformer models into sparse and often human-understandable features, supporting the hypothesis that the latent space of LLMs is composed of linear and interpretable directions (Arora et al., 2018; Elhage et al., 2022). We use the open Gemma Scope SAEs (Lieberum et al., 2024) to obtain such features and, in this interventionist tradition, validate our selected features causally by activating, suppressing, and substituting matched random or excluded controls.

Activation and feature steering. Activation steering controls model outputs by adding a vector to the residual stream, typically derived from contrasting positive and negative prompts (Turner et al., 2024). Subsequent work has scaled the approach to larger models (Panickssery et al., 2024), broadened the range of behaviors it can target (Zou et al., 2023), learned steering vectors directly for desired outputs (Cao et al., 2024), and introduced input-conditional variants (Lee et al., 2024). A complementary line of work steers at the level of individual SAE features, taking advantage of their interpretability to clamp or edit specific concepts (Templeton et al., 2024; Marks et al., 2024; Durmus et al., 2024; Zhao et al., 2024). Our method belongs to this latter family, but differs in how features are selected: rather than relying on a single labeled concept or hand-chosen latents, we identify reasoning-relevant features contrastively from paired reference-correct, target-incorrect traces, and use the resulting set to bridge a *cross-lingual* performance gap that prior steering work has not addressed.

7 Conclusion

We have studied cross-lingual reasoning gaps in large language models from a mechanistic perspective, asking whether language modulates the elicitation of latent computations rather than only whether models perform better in some languages than others. Using sparse autoencoders, we identify task-relevant features through contrastive comparison between successful reference-language reasoning traces and matched target-language failures, and inject them at inference time as a residual-stream steering direction. Across two models, three benchmarks, and four low-resource target languages, this intervention closes 25–33% of the gap to the reference language with no parameter updates, no translation, and no change to the user-facing language. Three intervention tests—partial sufficiency, functional necessity, and specificity—provide interventional evidence that the selected features are functionally implicated in the observed gap. Together, these results suggest that part of the cross-lingual reasoning gap reflects *under-elicitation* of computations the model already possesses, rather than their absence, and point to feature-level intervention as a complementary route to translation- and training-based remedies.

Limitations

Models and benchmarks. We evaluate two instruction-tuned models (Gemma-2-9B-it, Qwen2.5-7B-Instruct) on three benchmarks (MATH500, MGSM, and the psychology subset of MMLU-ProX) covering four low-resource target languages. Generalization to larger or differently

architected models, to other reasoning domains (e.g., commonsense, code), and to truly extremely low-resource languages remains to be verified.

Reliance on pretrained SAEs. Our feature library is constructed in the basis of an externally trained SAE applied at a single layer (layer 20). Both the resolution of the identified features and the effectiveness of steering are therefore bounded by the quality of the underlying SAE; polysemantic or under-trained latents may dilute the signal isolated by our contrastive procedure.

Transfer across tasks. Our experiments primarily evaluate feature transfer across languages within the same task. Cross-task transfer may depend on differences in task distributions and the extent to which reasoning mechanisms are shared across datasets, which we do not explicitly model in this work. Establishing when task-specific features transfer across benchmarks therefore remains an important direction for future work.

Acknowledgments

This work was supported in part by the National Research Foundation of Korea [NRF-2023R1A2C3004176], the Ministry of Health & Welfare, Republic of Korea [HR20C002103], the Ministry of Science and ICT (MSIT) [RS-2023-00262002], the Institute of Information & Communications Technology Planning & Evaluation(IITP)-ICT Creative Consilience Program grant funded by the Korea government(MSIT)(IITP-2026-RS-2020-II201819), and the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT and MOE) (No. RS-2025-16652968). The work of Junhyun Lee was supported by the Hankuk University of Foreign Studies Research Fund (2026).

References

Sanjeev Arora, Yuanzhi Li, Yingyu Liang, Tengyu Ma, and Andrej Risteski. 2018. Linear algebraic structure of word senses, with applications to polysemy. *Transactions of the Association for Computational Linguistics*, 6:483–495.

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.

Joseph Bloom, Curt Tigges, Anthony Duong, and David Chanin. 2024. Saelens. https://github.com/decoderesearch/SAELens.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, and 6 others. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

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

Sviatoslav Chalnev, Matthew Siu, and Arthur Conmy. 2024. Improving steering vectors by targeting sparse autoencoder features. *Preprint*, arXiv:2411.02193.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, and 1 others. 2023. Sparse autoencoders find highly interpretable features in language models. *Preprint*, arXiv:2309.08600.

Esin Durmus, Alex Tamkin, Jack Clark, Jerry Wei, Jonathan Marcus, Joshua Batson, Kunal Handa, Liane Lovitt, Meg Tong, Miles McCain, Oliver Rausch, Saffron Huang, Sam Bowman, Stuart Ritchie, Tom Hennighan, and Deep Ganguli. 2024. Evaluating feature steering: A case study in mitigating social biases. https://anthropic.com/research/evaluating-feature-steering. Accessed 2024-10-25.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, and 1 others. 2022. Toy models of superposition. *arXiv preprint arXiv:2209.10652*.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Julen Etxaniz, Gorka Azkune, Aitor Soroa, Oier Lopez de Lacalle, and Mikel Artetxe. 2024. Do multilingual language models think better in English? In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 550–564, Mexico City, Mexico. Association for Computational Linguistics.

Gemma Team. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akash Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *NeurIPS*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Deokhyung Kang, Seonjeong Hwang, Daehui Kim, Hyounghun Kim, and Gary Geunbae Lee. 2026. Why do multilingual reasoning gaps emerge in reasoning language models? *Preprint*, arXiv:2510.27269.

Amir Hossein Kargaran, Ayyoob Imani, François Yvon, and Hinrich Schuetze. 2023. Glotlid: Language identification for low-resource languages. *arXiv preprint arXiv:2310.16248*.

Bruce W. Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. 2024. Programming refusal with conditional activation steering. *Preprint*, arXiv:2409.05907.

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*.

Zhaoyi Li, Gangwei Jiang, Hong Xie, Linqi Song, Defu Lian, and Ying Wei. 2024. Understanding and patching compositional reasoning in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9668–9688, Bangkok, Thailand. Association for Computational Linguistics.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. *Preprint*, arXiv:2408.05147.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Wenyang Luo, Wayne Xin Zhao, Jing Sha, Shijin Wang, and Ji-Rong Wen. 2025. Mmath: A multilingual benchmark for mathematical reasoning. *Preprint*, arXiv:2505.19126.

Samuel Marks, Can Rager, Eric J. Michaud, and 1 others. 2024. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *Computing Research Repository*, arXiv:2403.19647.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Andrew Ng and 1 others. 2011. Sparse autoencoder. *CS294A Lecture notes*, 72(2011):1–19.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. Culturax: A cleaned, enormous, and multilingual dataset for large language models in 167 languages. *Preprint*, arXiv:2309.09400.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*.

Nina Panickssery, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. 2024. Steering Llama-2 via Contrastive Activation Addition. *Preprint*, arXiv:2312.06681.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Judea Pearl. 2001. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, page 411–420, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

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.

Senthooran Rajamanoharan, Tom Lieberum, Nicolas Sonnerat, Arthur Conmy, Vikrant Varma, János Kramár, and Neel Nanda. 2024. Jumping ahead: Improving reconstruction fidelity with jumprelu sparse autoencoders. *Preprint*, arXiv:2407.14435.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15504–15522, Bangkok, Thailand. Association for Computational Linguistics.

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

Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. 2025. Layer by layer: Uncovering hidden representations in language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 55854–55875. PMLR.

Aaquib Syed, Can Rager, and Arthur Conmy. 2023. Attribution patching outperforms automated circuit discovery. *Preprint*, arXiv:2310.10348.

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.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, and 3 others. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2024. Activation addition: Steering language models without optimization. *Preprint*, arXiv:2308.10248.

Kevin Ro Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2023. Interpretability in the wild: a circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, and 375 others. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

Weihao Xuan, Rui Yang, Heli Qi, Qingcheng Zeng, Yunze Xiao, Aosong Feng, Dairui Liu, Yun Xing, Junjue Wang, Fan Gao, Jinghui Lu, Yuang Jiang, Huitao Li, Xin Li, Kunyu Yu, Ruihai Dong, Shangding Gu, Yuekang Li, Xiaofei Xie, and 13 others. 2025. Mmlu-prox: A multilingual benchmark for advanced large language model evaluation. *Preprint*, arXiv:2503.10497.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A massively multilingual pre-trained text-to-text transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Hongbin Zhang, Kehai Chen, Xuefeng Bai, Yang Xiang, and Min Zhang. 2025. Lingualift: An effective two-stage instruction tuning framework for low-resource language reasoning. *Preprint*, arXiv:2412.12499.

Yu Zhao, Alessio Devoto, Giwon Hong, Xiaotang Du, Aryo Pradipta Gema, Hongru Wang, Xuanli He, Kam-Fai Wong, and Pasquale Minervini. 2024. Steering knowledge selection behaviours in llms via sae-based representation engineering. *Preprint*, arXiv:2410.15999.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2765–2781, Mexico

City, Mexico. Association for Computational Linguistics.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2023. Representation engineering: A top-down approach to ai transparency. *Preprint*, arXiv:2310.01405.

A Implementation Details

Models. We use two open instruction-tuned models: `google/gemma-2-9b-it` (Gemma-2-9B-it) and `Qwen/Qwen2.5-7B-Instruct`. Both are loaded with HuggingFace transformers (Wolf et al., 2019). All interventions are applied to the layer-20 residual stream.

Sparse autoencoders. For Gemma-2-9B-it we use the publicly released Gemma Scope residual-stream SAEs (Lieberum et al., 2024) (`gemma-scope-9b-pt-res-canonical`, layer 20, width 16k, i.e. d_{sae}=16,384; JumpReLU), loaded through `sae_lens`. For Qwen2.5-7B-Instruct we use the layer-20 matryoshka SAE (`chanind/qwen2.5-7B-it-layer-20-saes`, `lmsys/matryoshka/k-100`; JumpReLU, d_{sae}=65,536), loaded with a manual configuration shim to bridge `sae_lens` (Bloom et al., 2024) config-schema versions. SAE activations are computed in `float32` on the captured residual stream.

Software. All experiments use Python 3.11, PyTorch 2.4.1 (Paszke et al., 2019) (CUDA 12.1), `transformers` 4.51.0, `sae_lens` 5.5.2. Mathematical answers are scored with the `math_verify` library; multiple-choice (MMLU-ProX) answers use a dedicated letter parser that extracts the leading letter from `\boxed{...}` with an "answer is X" fallback.

Hardware. Experiments run on 4 NVIDIA H100 GPUs (80 GB), using a single GPU per job.

Inference and decoding. We decode greedily (`do_sample=False`) with a maximum of 1,024 new tokens. A response is marked truncated if generation reaches the token limit without emitting an end-of-sequence token. Prompts follow a per-language native-language template instructing the model to reason step by step and to place its final answer in `\boxed{...}`.

B Output Language Consistency

To verify that steering does not cause the model to revert to the high-resource reference language, we perform automatic language identification on all generated responses using GlotLID (Kargaran et al., 2023). We measure the proportion of output identified as the intended target low-resource language at two granularities: *per-segment*, where each response is divided into sentence-level segments and classified individually, and *per-response*, where the entire generated response receives a single language label.

Table 4 reports the target-language proportion before and after steering for every model, dataset, reference language, and target language combination. Overall, steering leaves the output language largely unchanged from the baseline: at the segment level, the target-language proportion changes by less than 1.5 percentage points in 37 out of 40 settings (average absolute change: 0.8 points; median: 0.41 points). This suggests that the observed accuracy improvements arise primarily from improved reasoning within the target language, rather than from the model reverting to the higher-resource reference language.

C Additional Layer Interventions

We use layer 20 as the primary intervention point, motivated by prior findings that reasoning-relevant representations emerge prominently in intermediate layers (Li et al., 2024; Skean et al., 2025). To test whether our results are specific to this choice, we additionally apply steering at layers 4, 12, 34, and 38 of Gemma-2-9B on MGSM, using English as the reference language.

As shown in Table 5, steering improves performance over the baseline across a broad range of intervention layers. Layer 20 achieves the strongest overall performance across target languages, consistent with prior findings that intermediate layers contain strong task-relevant representations.

D Comparison with Related Steering Method

Contrastive Activation Addition (CAA) (Rimsky et al., 2024) and our SAE-based approach both intervene on the residual stream at inference time. CAA operates directly on dense hidden-state differences, whereas our method steers sparse features associated with reasoning-relevant concepts shared across languages.

SAE-based steering achieves higher accuracy than CAA in 11 of the 12 evaluation settings while providing feature-level interpretability through individually identifiable features (Table 6).

E Generalization to Unseen Target Languages

We further evaluate whether our method generalizes to target languages beyond those used in the main experiments. We consider Bengali (BN) and Telugu (TE) on MGSM with Gemma-2-9B, using English as the reference language. Pair-specific steering improves accuracy from 75.6% to 81.2% for Bengali and from 74.4% to 75.6% for Telugu, while preserving target-language generation (Table 7).

We also test whether features identified for one target language can be reused for unseen target languages. Specifically, we apply feature sets extracted from EN→KO and EN→TH directly to Bengali and Telugu inference. Both transferred feature sets improve over the baseline, although pair-specific feature extraction remains strongest (Table 8).

We interpret this as evidence that each target language may under-elicit a partially different subset of reasoning features. Consequently, features selected for one language may be less relevant or already sufficiently active in another, leading to only partial cross-language transfer. These results suggest that the feature sets are transferable across target languages, while pair-specific extraction better captures each language's own under-activated features.

Table 4: Target-language consistency before and after steering, measured using GlotLID. Values denote the percentage of generated output identified as the intended target language. Each cell reports **Baseline** → **Steered**.

Model	Dataset	Reference	Target	per-segment (%)	per-response (%)
Gemma-2-9B	MATH500	EN	KO	98.3 → 98.2	99.4 → 99.4
		EN	TH	97.6 → 97.9	99.4 → 99.4
		EN	VI	99.0 → 99.2	99.7 → 100.0
		ES	KO	98.3 → 98.8	99.4 → 99.4
		ES	TH	97.6 → 93.1	99.4 → 95.8
		ES	VI	99.0 → 89.7	99.7 → 96.8
	MGSM	EN	KO	100.0 → 100.0	100.0 → 100.0
		EN	TH	100.0 → 100.0	100.0 → 100.0
		EN	SW	97.6 → 97.4	99.2 → 99.6
		ES	KO	100.0 → 100.0	100.0 → 100.0
		ES	TH	100.0 → 100.0	100.0 → 100.0
		ES	SW	97.6 → 96.9	99.2 → 99.2
	MMLU-ProX (psychology)	EN	KO	99.9 → 99.9	100.0 → 99.9
		EN	TH	99.2 → 99.1	99.6 → 99.6
		EN	VI	98.9 → 99.8	99.5 → 99.9
		EN	SW	99.4 → 99.4	99.8 → 99.8
		ES	KO	99.9 → 99.9	100.0 → 100.0
		ES	TH	99.2 → 98.2	99.6 → 99.9
		ES	VI	98.9 → 99.5	99.5 → 99.8
		ES	SW	99.4 → 99.2	99.8 → 99.8
Qwen2.5-7B	MATH500	EN	KO	95.0 → 96.0	95.8 → 97.4
		EN	TH	95.2 → 95.8	95.8 → 95.5
		EN	VI	98.5 → 98.1	99.0 → 99.0
		ES	KO	95.0 → 93.9	95.8 → 94.5
		ES	TH	95.2 → 95.2	95.8 → 95.5
		ES	VI	98.5 → 99.0	99.0 → 99.4
	MGSM	EN	KO	99.9 → 98.5	100.0 → 98.4
		EN	TH	72.0 → 71.2	63.6 → 63.6
		EN	SW	97.3 → 98.3	99.6 → 99.6
		ES	KO	99.9 → 99.7	100.0 → 99.6
		ES	TH	72.0 → 71.9	63.6 → 63.6
		ES	SW	97.3 → 99.0	99.6 → 100.0
	MMLU-ProX (psychology)	EN	KO	96.7 → 95.7	97.0 → 95.5
		EN	TH	93.0 → 92.0	91.7 → 90.6
		EN	VI	99.7 → 99.4	99.8 → 99.6
		EN	SW	92.1 → 93.0	96.1 → 94.9
		ES	KO	96.7 → 95.8	97.0 → 95.7
		ES	TH	93.0 → 93.1	91.7 → 91.7
		ES	VI	99.7 → 99.7	99.8 → 99.9
		ES	SW	92.1 → 91.6	96.1 → 95.2

Table 5: Accuracy (%) on MGSM when steering Gemma-2-9B at different intervention layers, using English as the reference language.

Layer	KO	TH	SW
Baseline	76.4	78.4	75.6
Layer 4	78.0	81.6	76.8
Layer 12	**79.2**	80.8	76.0
Layer 20	78.0	**82.4**	**79.2**
Layer 34	78.0	81.6	77.2
Layer 38	75.2	81.2	76.8

Table 6: Comparison with CAA on Gemma-2-9B. Values report accuracy (%). Best steering results are shown in bold.

		MATH500			MGSM		
Ref.	**Method**	**TH**	**KO**	**VI**	**TH**	**KO**	**SW**
EN	Baseline	48.23	49.20	48.23	78.40	76.40	75.60
	CAA	46.95	47.91	51.45	80.40	77.20	**80.00**
	Ours	**49.20**	**52.41**	**51.77**	**82.40**	**78.40**	79.20
ES	Baseline	48.23	49.20	48.23	78.40	76.40	75.60
	CAA	44.40	48.60	48.90	80.80	77.20	76.00
	Ours	**48.87**	**50.16**	**50.48**	**82.40**	**78.00**	**77.20**

Table 7: Generalization to additional low-resource target languages on MGSM with Gemma-2-9B. Recovery denotes the fraction of the performance gap to the English reference recovered by steering.

Metric	**Method**	**BN**	**TE**
Accuracy (%)	Baseline	75.6	74.4
	+Steering	**81.2**	**75.6**
	Recovery	+40%	+8%
Target language proportion (%)	Baseline	99.76	99.88
	+Steering	100.00	99.63

Table 8: Cross-language transfer to unseen target languages on MGSM with Gemma-2-9B. Values report accuracy (%). Target-specific features are extracted separately for each target language, while EN→KO and EN→TH features are reused directly for BN and TE.

Feature Set	**BN**	**TE**
Baseline	75.6	74.4
Target-specific	**81.2**	**75.6**
EN→KO features	78.8	74.8
EN→TH features	78.8	75.2