Multilingual Steering by Design: Multilingual Sparse Autoencoders and Principled Layer Selection

Anonymous ACL submission

Abstract

Sparse autoencoders (SAEs) have emerged as a tool for interpreting and steering large language models (LLMs), with prior work showing that they can be used to force models to produce outputs in a desired target language. However, controlling the output language of LLMs with SAEs remains unreliable when SAEs are trained on English-only data and when steering layers are chosen heuristically. We address both issues with two contributions. First, we show that training SAEs on multilingual data, rather than English-only data, consistently improves language steering across layers and models, yielding higher BLEU score, stronger language identification accuracy, and more reliable generation in the target language. Second, we introduce a principled, *a priori* layer-selection rule based on the intersection of multilingual alignment and language separability, which identifies effective steering depths without exhaustive layerwise search while substantially reducing time and compute. Evaluated on LLaMA-3.1-8B and Gemma-2-9B, our approach turns SAEs into a reliable and efficient mechanism for multilingual control. We release multilingual SAE checkpoints and code to facilitate reproducibility and further research.

1 Introduction

Large language models (LLMs) can produce text in many languages, yet *controlling* the output language remains challenging: generations are shaped by biases from the training data, and interventions to reliably enforce a target language are still poorly understood (Tang et al., 2024; Deng et al., 2025). Sparse autoencoders (SAEs) have emerged as a promising tool to interpret internal activations and construct steering vectors for controlling the language of the output (Cunningham et al., 2023; Templeton, 2024). However, two practical obstacles limit their effectiveness in multilingual settings. First, most SAEs are trained on English-only data (He et al., 2024a), which biases their features toward monolingual structure and weakens cross-lingual control. Second, steering layers are typically chosen heuristically (e.g., "mid-to-late" depths), requiring expensive layer sweeps and yielding inconsistent results across models (Bayat et al., 2025; Chou et al., 2025).

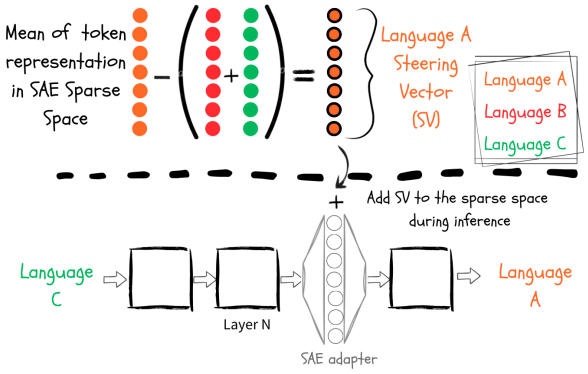

Figure 1: Illustration of language control with an SAE. **Top:** A per-language steering vector is formed in the SAE's sparse space by contrasting the target language with all other languages. **Bottom:** Inserted at a chosen layer, an SAE adapter applies this vector to the model's activations, steering generation into the target language.

We address both obstacles by **training SAEs on multilingual data** for LLaMA-3.1-8B (Grattafiori et al., 2024) and Gemma-2-9B (Team et al., 2024) and **selecting steering layers using a *principled* metric**. We show that multilingual training of SAEs enables robust and quality-preserving steering into target languages across model families (Figure 1 illustrates our steering method). We further compare the trained SAEs to their open-source equivalents–Llama-Scope (He et al., 2024b) and Gemma-Scope (Lieberum et al., 2024). To avoid heuristic depth choices, we introduce a principled layer selection metric based on the *intersection* of multilingual alignment and language separability, which predicts effective steering layers without exhaustive search.

Our contributions are threefold:

- **Multilingual SAE framework.** We show that compared to English-only SAEs, multilingual SAEs consistently yield stronger language control, improving BLEU score and language identification accuracy (LangID) while maintaining controllability.
- **Principled steering layer selection.** We introduce an intersection-based metric to identify effective steering depths, eliminating layer sweeps and stabilizing performance across models, saving time and compute.
- **Open-source multilingual SAE suite.** We release multilingual SAE checkpoints for LLaMA-3.1-8B and Gemma-2-9B, with code to facilitate reproducibility and further research.

2 Related Work

Steering LLM Activations via Sparse Features. Recent work explores sparse autoencoders (SAEs) for interpreting and steering LLM activations (Templeton, 2024; Zhao et al., 2024; O'Brien et al., 2024; Wang et al., 2025; Zhao et al., 2025). Bayat et al. (2025) introduce *Sparse Activation Steering (SAS)* using JumpReLU SAEs trained on *Gemma-2* to steer behaviors such as refusals and hallucinations; steering is applied in *intermediate layers* (e.g., 12 for availability, 14 for stability) without a principled selection rule. Other SAE-based approaches, including Feature-Guided Activation Addition (FGAA) (Soo et al., 2025) and SAE-Targeted Steering (SAE-TS) (Chalnev et al., 2024), likewise demonstrate that manipulating small sets of sparse features can causally influence outputs, but they typically rely on fixed or heuristic layer choices.

Output Language Control via SAEs. Several studies apply SAEs to control output language. Chou et al. (2025) show that editing a single SAE feature at an appropriate layer of *Gemma-2-9B* can flip outputs between English and target languages (Chinese, Japanese, Spanish, French), with strongest effects in *mid-to-late* layers (e.g., 23); layer choice, importantly however, is determined by manual exploration. Deng et al. (2025) extend this to *Gemma-2* and *LLaMA-3.1-8B*, proposing a *monolinguality metric* to quantify language-specificity of SAE features; ablating such features selectively impairs the corresponding language, and gating steering vectors improves control. Nonetheless, SAE and layer selection remain heuristic (e.g., choosing the SAE with the second-smallest L_0 from Gemma-Scope at each layer, where L_0 denotes the baseline cross-entropy loss of the unsteered model for a target language). Consistent with prior reports, we also find that the most effective steering depths tend to occur in mid-to-late layers; moreover, our intersection-based selection provides a principled alternative to manual exploration by identifying layers that jointly optimize output quality and language identification, averaged across languages.

Evaluating SAEs. Standardized SAE evaluation is emerging through complementary benchmarks and metrics. Karvonen et al. (2025) introduce *SAE-Bench*, covering fidelity, interpretability, and intervention quality across architectures and motivating metrics beyond reconstruction error. Wu et al. (2025) propose *AxBench*, a benchmark for activation-steering methods that reports strong performance for simple residual-stream baselines and mixed results for SAE-based steering. Orthogonally, Härle et al. (2025) define the *Feature Monosemanticity Score (FMS)* to quantify how consistently features map to single concepts. Other work has emphasized fidelity: *Gemma-Scope* (Lieberum et al., 2024) defines fidelity via mean squared error and causal loss when splicing SAEs into the model, showing that high reconstruction error in residual-stream SAEs harms causal fidelity; *LLaMA-Scope* (He et al., 2024b) extends Top-K SAEs across residual, attention, and MLP sites in *LLaMA-3.1-8B*, but reports relatively high reconstruction error, which may contribute to weak multilingual signals in residual-space features; and *JumpReLU* (Rajamanoharan et al., 2024) improves the fidelity–sparsity frontier via straight-through training of a discontinuous activation, boosting reconstruction fidelity without sacrificing interpretability. We build on these insights by showing that high-fidelity JumpReLU SAEs can preserve rare multilingual features which otherwise tend to collapse in lower-fidelity residual SAEs.

Language Features inside Models. Beyond SAEs, analyses point to layer-dependent language signals. Tang et al. (2024) identify language-specific neurons in BLOOM and LLaMA-2 and show that toggling them can switch the output language. Chang et al. (2022) study multilingual geometry in XLM-R, finding that languages occupy parallel subspaces separated by linear "language vectors," especially in middle layers; shifting hid-

den states along these directions flips predictions. Our findings echo these trends in the distribution of multilinguality across models and support treating language as a steering direction in representation space, which also reveals correlations among language families (Gurgurov et al., 2025).

3 SAE-based Language Vectors and Layer Selection

We first define **SAE-based language vectors**, which both provide steering directions and specify how to apply them at inference. We then introduce a **principled layer-selection criterion** to identify where steering is most effective.

3.1 SAE-based Language Vectors

To measure presence and steerability of individual languages, we extract language vectors from the residual stream and the sparse representation of the SAE, as presented in Figure 1. We describe the mathematical details in the following subsections.

3.1.1 Representation Extraction

At each layer ℓ of the model, we begin with the hidden (dense) representation from the residual stream:

$$h_\ell(x) \in \mathbb{R}^D,$$

for input x.

To obtain a sparse representation, we apply an encoder–decoder sparse autoencoder (SAE) on these hidden states. The encoder maps the dense activation to a high-dimensional sparse code:

$$z_\ell(x) = \mathrm{Encoder}_\ell(h_\ell(x)),\ z_\ell(x) \in \mathbb{R}^K,\ K \gg D,$$

and the decoder reconstructs it back into the dense space:

$$\hat{h}_\ell(x) = \mathrm{Decoder}_\ell(z_\ell(x)).$$

In practice, $z_\ell(x)$ is highly sparse, enforced by the SAE's sparsity objective, allowing us to isolate interpretable features for steering.

3.1.2 Steering

In the sparse space, steering vectors are constructed by contrasting activations using *DiffMean* (Wu et al., 2025)[1] from positive (target language) and negative (all other languages) examples. Let $\mathcal{Z}^+$ and $\mathcal{Z}^-$ denote the sets of sparse codes obtained from the SAE encoder at layer ℓ. We first compute the mean codes

$$\bar{z}_\ell^+ = \frac{1}{|\mathcal{Z}^+|} \sum_{z \in \mathcal{Z}^+} z, \qquad \bar{z}_\ell^- = \frac{1}{|\mathcal{Z}^-|} \sum_{z \in \mathcal{Z}^-} z.$$

In the sparse SAE residual space, the resulting steering vector is then defined as

$$w_{\mathrm{DiffMean}}(\ell) = \bar{z}_\ell^+ - \bar{z}_\ell^-$$

3.1.3 Inference

At inference time, given a hidden activation $h_\ell(x)$ we apply the following procedure for steering:

1. Encode into sparse space:

$$z_\ell(x) = \mathrm{Encoder}_\ell(h_\ell(x)).$$

2. Apply the steering vector:

$$z'_\ell(x) = z_\ell(x) + \alpha\, w_{\mathrm{DiffMean}}(\ell).$$

3. Decode back to dense space:

$$\hat{h}'_\ell(x) = \mathrm{Decoder}_\ell(z'_\ell(x)).$$

4. Correct reconstruction error with the residual:

$$\tilde{h}_\ell(x) = \hat{h}'_\ell(x) + \big(h_\ell(x) - \mathrm{Decoder}_\ell(z_\ell(x))\big).$$

This procedure shifts activations toward the target language by amplifying discriminative sparse features while suppressing overlapping ones.

3.2 Principled Layer Selection

Localizing where to steer is difficult because representations evolve with depth: early-mid layers emphasize shared, language-agnostic structure, while later layers sharpen language-specific form (Wendler et al., 2024). Heuristic mid–late choices are brittle, and exhaustive layer sweeps are expensive. Our approach targets depths where *alignment* (shared structure) and *separability* (distinct language signals) are balanced to support cross-lingual tasks like translation and contain enough language-specific separation to make a small push to reliably change the output language.

[1] Wu et al. (2025) apply the DiffMean method directly in the *dense* representation (without SAE) and use $h'_\ell(x) = h_\ell(x) + \alpha\, w_{\mathrm{DenseDiffMean}}(\ell)$. for inference. We refer to this method as dense steering.

3.2.1 Language Correlation Detection

Per-language representation. For each language i and layer ℓ, we construct a contrast (language) vector $\mathbf{v}_i$ using *DiffMean* (Wu et al., 2025): samples from language i are positives and all other languages are negatives. Let $\mathcal{Z}_i^+$ denote sparse representation (Section 3.1.1) from language i and $\mathcal{Z}_i^-$ from all others. We then define vector v_i as:

$$\mathbf{v}_i \;=\; \frac{1}{|\mathcal{Z}_i^+|}\sum_{z\in\mathcal{Z}_i^+} z \;-\; \frac{1}{|\mathcal{Z}_i^-|}\sum_{z\in\mathcal{Z}_i^-} z\,.$$

This represents a language by amplifying features specific to language i while suppressing shared features.

Correlation across languages. With $\{\mathbf{v}_i\}_{i=1}^L$, we compute a pairwise *Pearson correlation* matrix

$$C_\ell \in \mathbb{R}^{L\times L}, \qquad C_{ij} \;=\; \mathrm{corr}(\mathbf{v}_i, \mathbf{v}_j),$$

capturing how similar languages appear (at layer ℓ) when represented by these contrast vectors.

3.2.2 Layer Selection

Multilinguality vs. separability. From C_ℓ we derive two complementary metrics:

$$\text{Multilinguality (FirstPC)}: \quad f_\ell \;=\; \frac{\max_i \lambda_i}{\sum_{j=1}^L \lambda_j}\,,$$

$$\text{Separability}: \quad s_\ell \;=\; 1 - f_\ell\,,$$

where $\{\lambda_j\}_{j=1}^L$ are the eigenvalues of C_ℓ and *FirstPC* is the explained-variance ratio of the first principal component of C_ℓ. This values measures the shared alignment across languages (higher $\Rightarrow$ stronger alignment), whereas s_ℓ captures how distinct the languages remain. Intuitively, the *Multilinguality* measure language alignment based on dominant shared directions in space of these languages, while the complementary *Separability* quantifies language-specific dispersion.

Intersection criterion. We select steering layers at intersection points of f_ℓ and s_ℓ, i.e., points where shared and language-specific signals are balanced. These intersections serve as *a priori* candidates for effective steering depths. Concretely, since $s_\ell = 1 - f_\ell$, a intersection occurs where $f_\ell = s_\ell \approx 0.5$, equivalently where $2f_\ell - 1$ changes sign between adjacent layers. In practice, we detect these sign changes with a small tolerance (treating near-equality as a intersection) and linearly interpolate between layer indices.

4 Experimental Protocol

We summarize our setup for evaluating multilingual SAE steering.

4.1 Models

We evaluate multilingual steering with SAEs on two base models: *LLaMA-3.1-8B* (Grattafiori et al., 2024) and *Gemma-2-9B* (Team et al., 2024). We report results for open-source baselines *LLaMA-Scope* (He et al., 2024b) and *Gemma-Scope* (Lieberum et al., 2024). In addition, for each base model we train two suites of JumpReLU SAEs–one on English-only data and one on multilingual data–following a unified setup described in Section 4.4.

4.2 Languages

Our experiments cover 21 target languages from FLORES–200 (Costa-Jussà et al., 2022): English, Tibetan, Maltese, Italian, Spanish, German, Japanese, Arabic, Chinese (Simplified), Afrikaans, Dutch, French, Portuguese, Russian, Korean, Hindi, Turkish, Polish, Swedish, Danish, and Norwegian Bokmål (see Appendix C).

4.3 SAE Training Data

We train two SAE suites per base model using the Wikipedia data (Wikimedia Foundation, 2023). The multilingual suite (*Our-multi-SAEs*) is *balanced* to target $\sim$ 1,000,000 row per language across the 21 languages. For the English-only suite (*Our-en-SAEs*), we use all the available data $\sim$6,500,000 rows.

4.4 SAE Training Procedure

For each layer of each base model, we train JumpReLU SAEs (Rajamanoharan et al., 2024) on the residual stream with an 8$\times$ expansion factor, using identical hyperparameters for *Our-en-SAEs* and *Our-multi-SAEs*. This controlled setup isolates the effect of multilingual vs. monolingual training data from architectural or optimization differences. Additional training details are provided in Appendix A.

4.5 Evaluation

We use the FLORES–200 `devtest` split for both constructing steering vectors and evaluating. For each non-English target language, we build the per-language steering vector from the *last 100* `devtest` sentences across 20 targets. We then evaluate machine translation on the *first 50* `devtest` sentences

per target (total $20 \times 50 = 1000$ sentences), with the source always English (eng_Latn). We report average scores across the 20 targets. Prompts are written *in the target language*: for each target language i, we retrieve equivalents of "Translate this sentence:" and "Translation:" (e.g., German: "*Übersetze diesen Satz:* … *Übersetzung:*") and format inputs as:

$$\underbrace{\texttt{Translate this sentence:}}_{\text{instruction in target language}}\ \underbrace{\text{“}\texttt{<source text>}\text{”}}_{\text{Always English}}.$$

$$\underbrace{\texttt{Translation:}}_{\text{answer cue in target language}}$$

We report two metrics: (1) **LangID**–language identification accuracy computed by applying a fastText classifier (Joulin et al., 2016) to generated outputs; and (2) **BLEU** (Post, 2018) against the reference in the intended *steer language*.[2]

For each target language i in our non-English set ($|i|$=20), we define an English→tgt_i task, construct the corresponding per-language steering vector, and apply it to steer generations into steer_i. For improved readability, the average across all 20 tasks is reported in the following, while per-language scores are provided in Appendix A.

As a baseline, we use the same translation prompt (English→steer_i) without any steering (i.e., no intervention), keeping all other settings identical.

5 Empirical Findings on Multilingual SAE Steering

We assess steering performance and layer sensitivity, focusing on: (i) benefits from multilingual SAE training, (ii) optimal intervention layers, and (iii) comparisons with open-source SAEs.

5.1 Benefits of Multilingual Training for SAEs

We study the effect of training data on SAE steering, comparing monolingual (English-only) SAEs with multilingual SAEs across transformer layers. We evaluate two downstream tasks: (i) language identification (LangID) and (ii) machine translation quality (BLEU score).

[2]We use tgt_i for the FLORES target language code and steer_i for the intended output language after steering. In our setup, steer_i is the language for which we construct the steering vector. We use the term "steer language" to emphasize that the output language is controlled via a steering intervention.

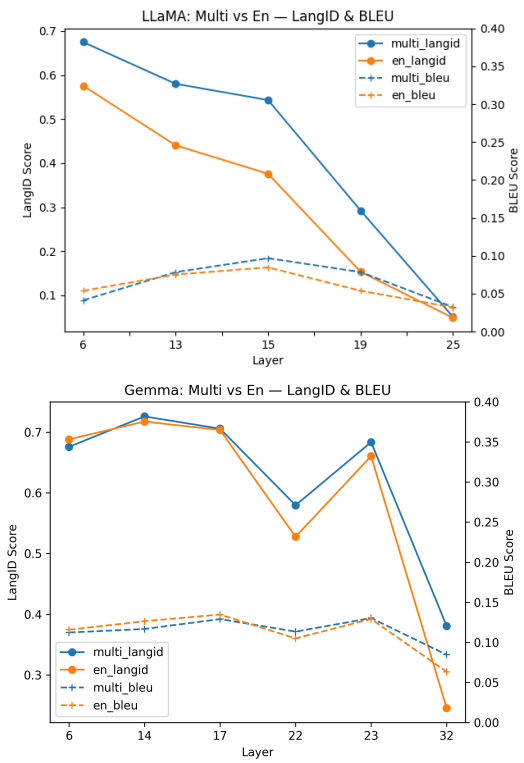

Figure 2: Multilingual vs. English-only SAEs across layers for **LLaMA-3.1-8B** (top) and **Gemma-2-9B** (bottom). Multilingual training yields strong, widespread gains on LLaMA and smaller but consistent improvements on Gemma for both LangID and BLEU. The left y-axis corresponds to LangID accuracy, while the right y-axis corresponds to BLEU.

Multilingual training improves steering. As shown in Figure 2, LLaMA-3.1-8B exhibits clear, sizeable gains from multilingual training across most layers for both LangID and BLEU. For Gemma-2-9B, the improvements are *smaller in magnitude* but remain *directionally consistent* across layers–often matching or slightly exceeding the English-only baseline. Overall, compared to English-only, multilingual SAEs yield better aggregate performance and indicate richer cross-lingual structure, with effect size varying by model family.

Sparse steering outperforms dense steering. Figure 3 compares steering in the SAE's *sparse* space with steering in the *dense* residual space. On LLaMA-3.1-8B, sparse steering yields noticeably stronger LangID scores across most layers and comparable or slightly higher BLEU. On Gemma-2-9B, the effect is smaller but consistent: LangID is generally on par with dense steering (with minor drops at a few layers), while BLEU shows modest, steady gains. Overall, sparse-space interventions

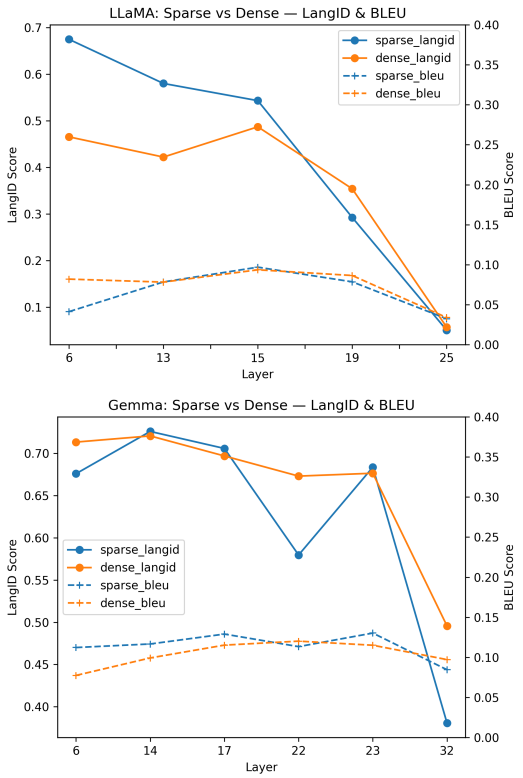

Figure 3: Sparse vs. dense steering in (top) LLaMA-3.1-8B and (bottom) Gemma-2-9B. Sparse steering strengthens LangID for Llama and yields modest BLEU improvements for Gemma. The left y-axis corresponds to LangID accuracy, while the right y-axis corresponds to BLEU.

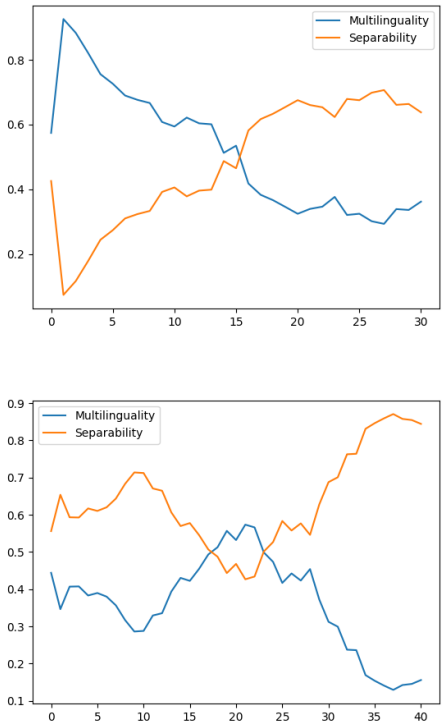

Figure 4: Intersection curves (multilinguality vs. separability) and detected balance points for our multilingual SAEs. Selected depths: LLaMA-3.1-8B (L15) and Gemma-2-9B (L17, L23).

improve language control without sacrificing translation quality.

5.2 Optimal Layers

For LLaMA-3.1-8B, we detect a clear intersection at **L15** (Figure 4), and for Gemma-2-9B two intersections at **L17** and **L23**. We define these intersections as balance points between *multilinguality* (FirstPC ratio f_ℓ; shared cross-lingual structure that supports tasks like translation) and *separability* ($s_\ell = 1 - f_\ell$; language-specific distinctiveness that helps to identify where to find language specific features). Mid layers typically raise f_ℓ (improving machine translation, but blurring language separability), while deeper layers raise s_ℓ (better separability between the languages, while eroding shared semantics). Intersection points therefore predict depths where multilinguality and separability are jointly strong (Figure 5). This pattern aligns with prior analyses showing that cross-lingual alignment peaks around middle depths while language-specific signals strengthen later: e.g., parallel subspaces and language directions in XLM-R (Chang et al., 2022), mid-layer alignment measured by Mexa (Kargaran et al., 2024), and increasing language selectivity in later layers/neuron groups (Tang et al., 2024; Tan et al., 2024). Intervention studies likewise report strongest language-control effects at mid–late depths (Chou et al., 2025; Deng et al., 2025).

In LLaMA-3.1-8B, a pronounced "jump" in multilinguality near **L14** is followed by a rise in separability, yielding a single intersection point at **L15** that delivers the best (BLEU, LangID) trade-off. In Gemma-2-9B, the curves exhibit a *two-hump* shape that produces intersections at **L17** (mid-layer balance) and **L23** (late-layer balance); both maintain stable LangID while delivering strong BLEU, matching reports that effective multilingual steering spans mid to late layers in Gemma-2 (Chou et al., 2025; Deng et al., 2025).

5.3 Additional Experiments: Steering with Open-Source SAEs

We compare our multilingual SAEs against the open-source suites *LLaMA-Scope* and *Gemma-*

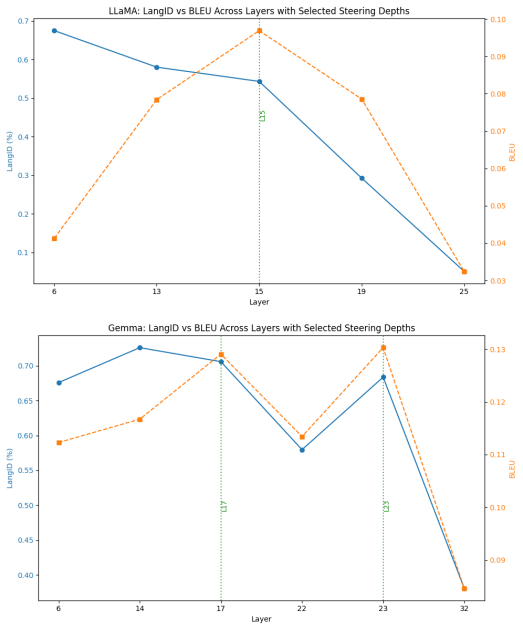

Figure 5: LangID and BLEU across layers for LLaMA-3.1-8B (top) and Gemma-2-9B (bottom). Green markers/lines indicate layers selected by the intersection rule, which deliver the best joint trade-off. The left y-axis corresponds to LangID accuracy, while the right y-axis corresponds to BLEU.

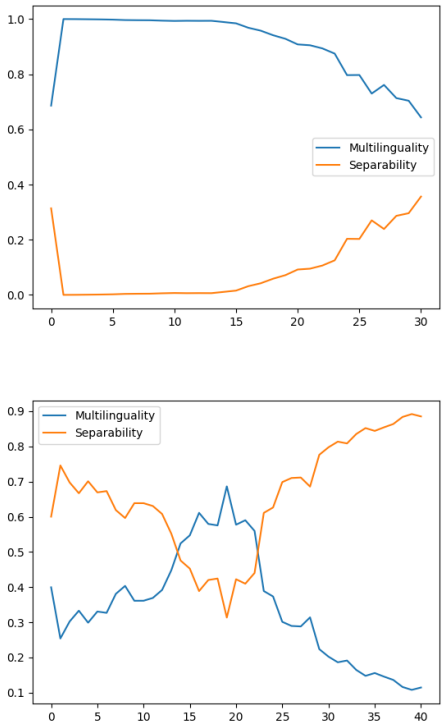

Figure 6: Layer-selection outputs for open-source suites: LLaMA-Scope (top) and Gemma-Scope (bottom). LLaMA-Scope does not exhibit a clear intersection layer; Gemma-Scope selects L14 and L22.

Scope. We find no intersection layers based on LLamaScope (Figure 6), which aligns with its negligible downstream gains. On Gemma-Scope, **L14** and **L22** are found as intersection points, where steering performs reasonably but still below our multilingual SAEs. Table 1 summarizes the LLaMA-3.1-8B comparison: our selection at **L15** achieves a strong BLEU–LangID balance while LLaMA-Scope provides little improvement across layers. Table 2 reports the Gemma-2-9B results: our selections at **L17** and **L23** outperform Gemma-Scope's **L14** and **L22**, with higher BLEU and higher LangID.

We further analyze language vectors and separability scores across layers and find that *LLaMA-Scope* exhibits weak multilingual structure (Figure 7); its sparse representations struggle to separate languages, often performing worse than the dense residual stream, likely due to English skewed activation data and training choices that collapse rare multilingual features. Figure 7 also reproduces the early–late pattern of multilinguality reported in prior work (Gurgurov et al., 2025; Tan et al., 2024); moreover, language *separability* increases from middle to late layers, and in the SAE sparse space this separation emerges earlier and more strongly than in the dense residual stream, helping to explain why steering is most effective at mid-to-late depths.

Layer	LLaMA-Scope	Our mult. SAEs
L6	01.44 / 00.89	67.49 / 04.13
L13	00.63 / 00.55	58.02 / 07.84
L15	00.17 / 02.12	**54.34 / 10.00 †**
L19	00.05 / 00.40	29.27 / 07.85
L25	00.22 / 00.52	05.08 / 03.24

Table 1: LLaMA-3.1-8B: LLaMA-Scope vs. our multilingual SAEs (LangID/BLEU). A † indicates a layer selected by our intersection-based layer selection method.

6 Take-Aways and Discussion

Our results show that multilingual training and principled layer selection jointly enable reliable SAE-based steering.

(I) Multilingual training strengthens representations. Across LLaMA-3.1-8B and Gemma-2-9B, multilingual SAEs outperform English-only SAEs on both LangID and BLEU. This suggests that multilingual training does more than widen language coverage: it induces richer cross-lingual structure

Layer	Gemma-Scope	Our mult. SAEs
L6	50.52 / 02.03	67.59 / 11.22
L14	48.46 / 02.72 †	72.59 / 11.67
L17	**56.66 / 06.08**	**70.58 / 12.90 †**
L22	**65.17 / 11.72 †**	57.96 / 11.34
L23	65.80 / 10.56	**68.36 / 13.03 †**
L32	46.63 / 09.41	38.06 / 08.45

Table 2: Gemma-2-9B: Gemma-Scope vs. our multilingual SAEs (LangID/BLEU). A † indicates a layer selected by our intersection-based layer selection method.

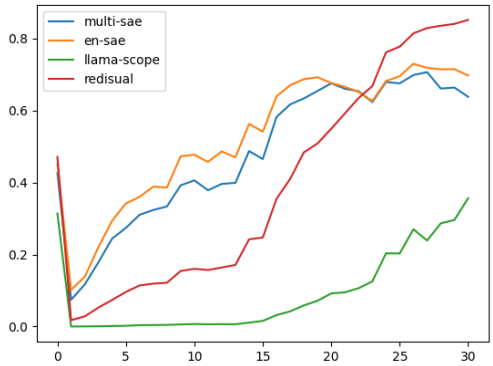

Figure 7: Separability across layers for LLaMA-3.1-8B, comparing our multilingual SAEs, our English-only SAEs, and LLaMA-Scope. Higher values indicate more distinct language-specific representations.

and cleaner language-specific signals in the sparse feature space.

(II) Sparse steering outperforms dense steering. Sparse-space interventions yield more disentangled language control maintaining translation quality comparable to dense steering. The effect is model-dependent and smaller than the gains from multilingual training, but directionally consistent.

(III) Intersection points predict optimal steering depths. Balancing multilingual alignment and language separability identifies layers where BLEU and LangID are jointly strong. These intersection points provide an interpretable, *a priori* rule that avoids exhaustive layer sweeps and explains why heuristic "mid–late" choices are best for steering vectors. The finding generalizes to open-source SAEs. Although the exact depths differ (e.g., L15 in LLaMA-scope; L17/L23 in Gemma-scope), the same intersection-based principle applies, indicating the method is not tied to a particular architecture.

(IV) Performance of open-source SAEs varies. Open-source suites provide useful baselines but illustrate the limits of heuristic layer choices. *LLaMA-Scope* does not reveal clear intersection points (Figure 6) and yields negligible gains across layers (Table 1). Consistent with this, its sparse representations show weak multilingual separability—often worse than the dense residual stream—suggesting that English-skewed data and training choices correlate with collapsed multilingual features (Figure 7). *Gemma-Scope* selects mid–late layers (L14, L22; Figure 6) and performs reasonably, yet our multilingual SAEs with intersection-based selections (L17, L23) achieve a stronger BLEU–LangID trade-off (Table 2). Together, these observations reinforce two points: multilingual SAE training improves the quality of language-specific structure, and intersection-based depth selection reliably identifies effective steering layers while avoiding compute-intensive layer sweeps.

7 Conclusion

We showed that reliable multilingual control in LLMs emerges from two simple ingredients: *multilingual SAE training* and *a principled rule for where to steer*. Training SAEs on multilingual data induces richer cross-lingual structure in the sparse feature space and yields consistent improvements in both BLEU and LangID—large on LLaMA-3.1-8B and smaller but steady on Gemma-2-9B. To replace heuristic mid–late choices and costly layer sweeps, we introduced a *principled* layer-selection rule based on the intersection of multilingual alignment and language separability; the selected depths (e.g., L15 in LLaMA, L17/L23 in Gemma) deliver the best joint trade-off between translation quality and language identification [3].

Practically, steering in the sparse space with simple DiffMean vectors is sufficient to realize these gains, improving language control without sacrificing translation quality while reducing compute by avoiding exhaustive searches. Because the procedure depends only on contrast vectors and eigenstructure at a layer, it is model-agnostic and transfers across architectures. We release multilingual SAE checkpoints and code to facilitate reproduction and further work on controllable, interpretable multilingual generation.

[3] In addition, we observe language-family correlations in our language vectors; details are provided in Appendix B.

Limitations

While our work establishes multilingual SAEs and intersection-based layer selection as effective tools for steering, several limitations remain. Our analysis was validated on a relatively small set of 21 languages, and the intersection points may shift with different language selections, particularly if more low-resource or single-family subsets are included. We evaluated primarily on LLaMA-3.1-8B and Gemma-2-9B, but larger or instruction-tuned models may exhibit different cross-lingual dynamics. The evaluation scope was restricted to BLEU and LangID, which do not capture stylistic fidelity, code-switching, robustness to noisy prompts, or human preferences. Moreover, our findings are based on JumpReLU SAEs trained on the residual stream, and their generalization to other sparse architectures or intervention sites (e.g., attention or MLP activations) remains to be tested. Finally, multilingual gains likely depend on corpus balance and coverage, and we did not exhaustively analyze the effects of language frequency distribution.

References

Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. 2025. Steering large language model activations in sparse spaces. *Preprint*, arXiv:2503.00177.

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

Tyler A Chang, Zhuowen Tu, and Benjamin K Bergen. 2022. The geometry of multilingual language model representations. *arXiv preprint arXiv:2205.10964*.

Cheng-Ting Chou, George Liu, Jessica Sun, Cole Blondin, Kevin Zhu, Vasu Sharma, and Sean O'Brien. 2025. Causal language control in multilingual transformers via sparse feature steering. In *ACL 2025 Student Research Workshop*.

Marta R Costa-Jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, and 1 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Hoagy Cunningham, Aidan Ewart, Logan Riggs, Robert Huben, and Lee Sharkey. 2023. Sparse autoencoders find highly interpretable features in language models. *arXiv preprint arXiv:2309.08600*.

Boyi Deng, Yu Wan, Yidan Zhang, Baosong Yang, and Fuli Feng. 2025. Unveiling language-specific features in large language models via sparse autoencoders. *Preprint*, arXiv:2505.05111.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daniil Gurgurov, Katharina Trinley, Yusser Al Ghussin, Tanja Baeumel, Josef van Genabith, and Simon Ostermann. 2025. Language arithmetics: Towards systematic language neuron identification and manipulation. *Preprint*, arXiv:2507.22608.

Ruben Härle, Felix Friedrich, Manuel Brack, Stephan Wäldchen, Björn Deiseroth, Patrick Schramowski, and Kristian Kersting. 2025. Measuring and guiding monosemanticity. *arXiv preprint arXiv:2506.19382*.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, Yu-Gang Jiang, and Xipeng Qiu. 2024a. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *Preprint*, arXiv:2410.20526.

Zhengfu He, Wentao Shu, Xuyang Ge, Lingjie Chen, Junxuan Wang, Yunhua Zhou, Frances Liu, Qipeng Guo, Xuanjing Huang, Zuxuan Wu, and 1 others. 2024b. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *arXiv preprint arXiv:2410.20526*.

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

Amir Hossein Kargaran, Ali Modarressi, Nafiseh Nikeghbal, Jana Diesner, François Yvon, and Hinrich Schütze. 2024. Mexa: Multilingual evaluation of english-centric llms via cross-lingual alignment. *arXiv preprint arXiv:2410.05873*.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, and 1 others. 2025. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability. *arXiv preprint arXiv:2503.09532*.

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. *arXiv preprint arXiv:2408.05147*.

Kyle O'Brien, David Majercak, Xavier Fernandes, Richard Edgar, Blake Bullwinkel, Jingya Chen, Harsha Nori, Dean Carignan, Eric Horvitz, and Forough Poursabzi-Sangdeh. 2024. Steering language model refusal with sparse autoencoders. *arXiv preprint arXiv:2411.11296*.

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

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, 2024b. *URL https://arxiv.org/abs/2407.14435*.

Samuel Soo, Chen Guang, Wesley Teng, Chandrasekaran Balaganesh, Tan Guoxian, and Yan Ming. 2025. Interpretable steering of large language models with feature guided activation additions. *arXiv preprint arXiv:2501.09929*.

Shaomu Tan, Di Wu, and Christof Monz. 2024. Neuron specialization: Leveraging intrinsic task modularity for multilingual machine translation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6506–6527, Miami, Florida, USA. Association for Computational Linguistics.

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. *arXiv preprint arXiv:2402.16438*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Adly Templeton. 2024. *Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet*. Anthropic.

Anyi Wang, Xuansheng Wu, Dong Shu, Yunpu Ma, and Ninghao Liu. 2025. Enhancing llm steering through sparse autoencoder-based vector refinement. *arXiv preprint arXiv:2509.23799*.

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.

Wikimedia Foundation. 2023. Wikipedia dump, november 1, 2023. https://dumps.wikimedia.org/. Accessed: 2025-10-06.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D Manning, and Christopher Potts. 2025. Axbench: Steering llms? even simple baselines outperform sparse autoencoders. *arXiv preprint arXiv:2501.17148*.

Haiyan Zhao, Xuansheng Wu, Fan Yang, Bo Shen, Ninghao Liu, and Mengnan Du. 2025. Denoising concept vectors with sparse autoencoders for improved language model steering. *arXiv preprint arXiv:2505.15038*.

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. *arXiv preprint arXiv:2410.15999*.

Appendix

A SAE Training

A.1 Hyperparameters

We train SAEs with `SAELens`[4] using a JumpReLU architecture on the residual stream at multiple layers. The base model is loaded in `float16`, while SAE training uses `float32`. Hook sites follow `blocks.{layer}.hook_resid_post`

Key hyperparameters (from code).

- **Architecture:** `jumprelu` (expansion factor $= 8$), L_1 coefficient $= 5.0$, JumpReLU bandwidth $= 10^{-3}$, init threshold $= 10^{-3}$, decoder init = zeros, transpose encoder init, decoder heuristic init enabled, sparsity penalty scaled by decoder norm.
- **Training length (#steps):** $15{,}000$.
- **Batch size (tokens/step):** $4{,}096$.
- **Total training tokens:** $15{,}000 \times 4{,}096 = 61{,}440{,}000$ tokens.
- **Context size:** 512.
- **Optimizer / schedule:** Adam ($\beta_1 = 0.9$, $\beta_2 = 0.999$), LR $= 5{\times}10^{-5}$, LR warmup $= 1{,}500$ steps, LR decay steps $= 3{,}000$, L_1 warmup $= 1{,}500$ steps.
- **Dead/feature refresh:** feature sampling window $= 2000$, dead-feature window $= 1000$, dead threshold $= 10^{-4}$.
- **Data loader:** streaming enabled, `prepend_bos=True`.

A.2 Computational Budget

For all experiments we used 1×Nvidia A100 80GB. Each SAE model was trained for approximately 10 million tokens, corresponding to about 3 GPU hours per run.

A.3 License and Availability

All trained SAE checkpoints produced in this work, including both English-only and multilingual variants for *LLaMA-3.1-8B* and *Gemma-2-9B*, are released under the **Apache License 2.0**. This license permits use, modification, and distribution of the models for both research and commercial purposes, provided that proper attribution is maintained.

We emphasize that the underlying base models (*LLaMA-3.1-8B*, *Gemma-2-9B*) remain subject to their original licenses as released by Meta and Google, respectively. Users of our checkpoints must therefore comply with both the Apache 2.0 license governing our SAEs and the terms of the corresponding base model licenses.

Code for training and evaluation, together with pretrained multilingual SAE checkpoints, is available at: `<anonymous>`.

[4] `https://github.com/jbloomAus/SAELens`

B Language Pairwise Correlations

Language vectors show linguistic proximity. Figure 21 in Appendix B shows pairwise correlations between per-language DiffMean vectors at the layers selected by our intersection rule–L15 in LLaMA-3.1-8B and L23 in Gemma-2-9B. Both matrices exhibit a clear block structure: languages within the same family (e.g., Romance, Germanic) have high within-block correlations, while cross-family pairs remain less correlated. The residual off-diagonal mass reflects a shared multilingual component, but the pronounced blocks indicate preserved language-specific directions. This alignment-with-separability profile is exactly what our selection criterion targets and helps explain why these depths deliver the best joint BLEU–LangID trade-off: shared structure supports transfer, whereas family-specific structure enables precise steering into the target language.

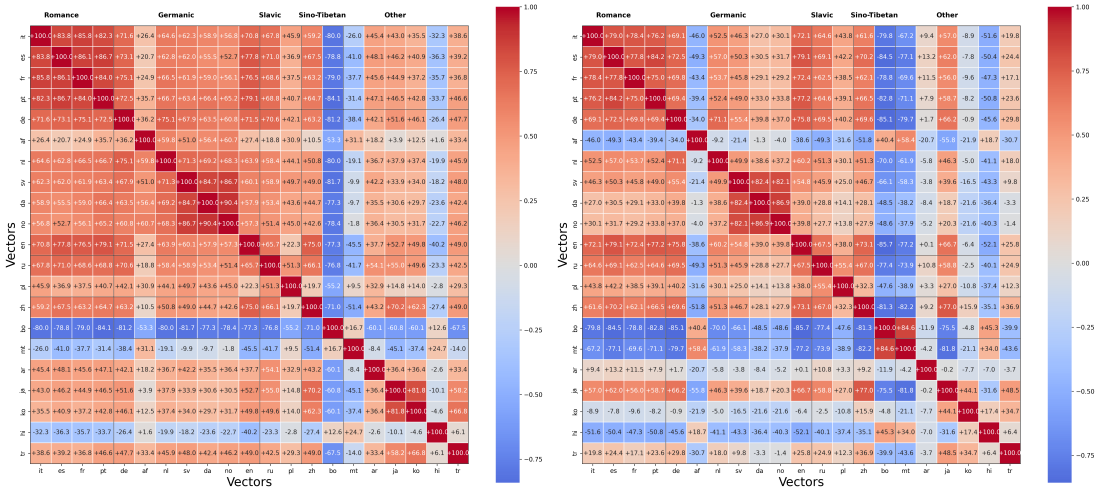

Figure 8: Correlation matrices of per-language contrast (DiffMean) vectors at the selected layers. **Left:** LLaMA-3.1-8B (L15). **Right:** Gemma-2-9B (L23). Block structure reveals language-family groupings (e.g., Romance, Germanic), indicating layers that preserve both cross-lingual alignment and language-specific distinctions.

C Language Labels

Language	Code
English	eng_Latn
Tibetan	bod_Tibt
Maltese	mlt_Latn
Italian	ita_Latn
Spanish	spa_Latn
German	deu_Latn
Japanese	jpn_Jpan
Arabic	arb_Arab
Chinese (Simplified)	zho_Hans
Afrikaans	afr_Latn
Dutch	nld_Latn
French	fra_Latn
Portuguese	por_Latn
Russian	rus_Cyrl
Korean	kor_Hang
Hindi	hin_Deva
Turkish	tur_Latn
Polish	pol_Latn
Swedish	swe_Latn
Danish	dan_Latn
Norwegian Bokmål	nob_Latn

Table 3: List of 21 target languages from FLORES–200 and their language codes.

D Benchmarking SAE Variants with SAEbench

We evaluate multiple SAE variants with SAEbench. Figure 9 summarizes two core axes. The top plot relates sparsity (measured by L_0) to the mean *absorption score*, quantifying how well learned features align with directions present in the underlying model. While reconstruction quality is generally high across variants, absorption remains relatively low, indicating limited alignment with interpretable directions. The bottom plot contrasts sparsity with *loss recovered*, highlighting the fidelity–interpretability trade-off: denser SAEs (higher L_0) approach perfect loss recovery, whereas sparser SAEs recover slightly less but offer greater potential for interpretable, disentangled features.

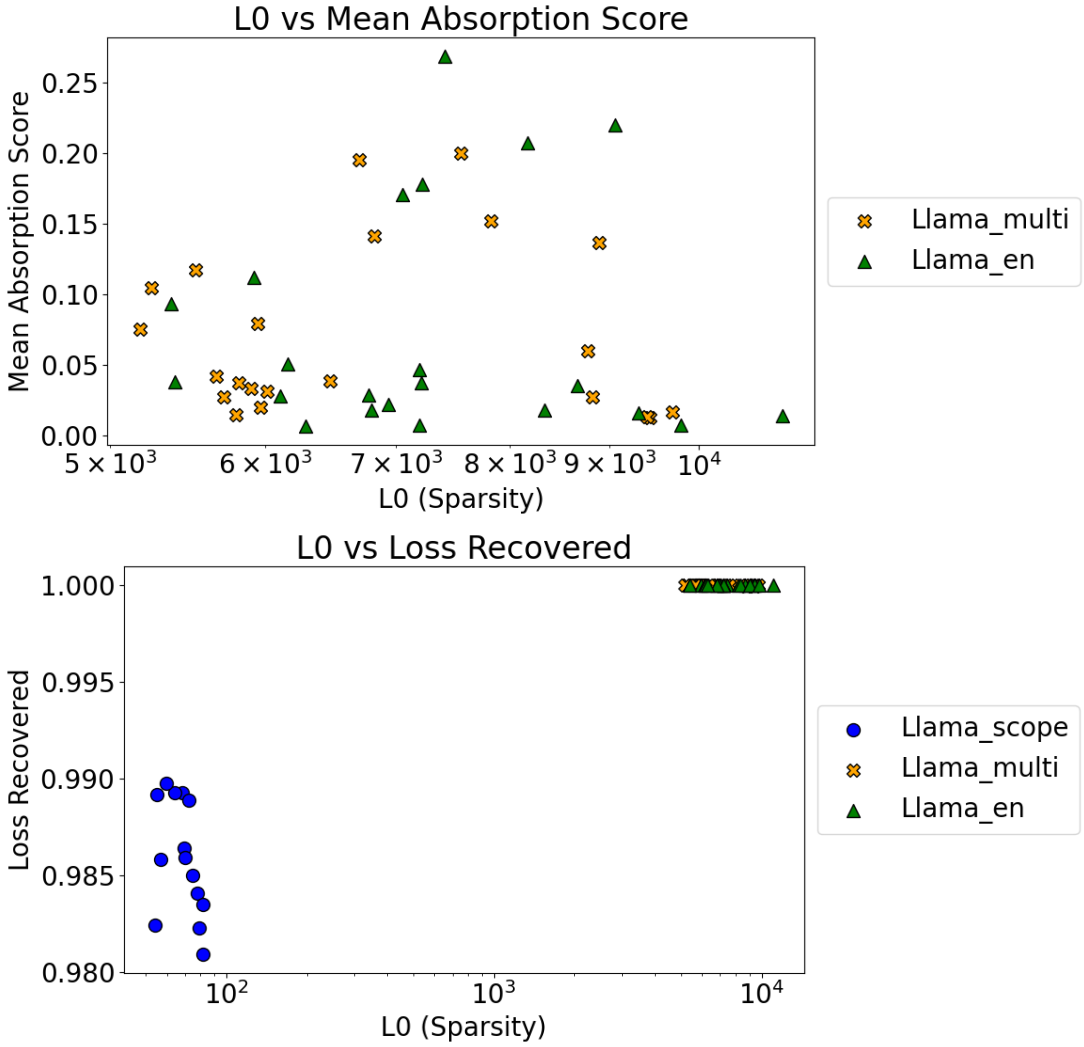

Figure 9: SAEbench comparison of SAE variants. **Top:** L_0 sparsity vs. mean absorption score (alignment with model directions). **Bottom:** L_0 sparsity vs. loss recovered, illustrating the fidelity–sparsity trade-off.

H Llama: Benchmarking SAE Variants with BLEU and LangID

Method	Layer	Resid	Llama-Scope	Our-multi-SAEs	Our-en-SAEs
Diffmean	L6	46.56 / 08.19	01.44 / 00.89	67.49 / 04.13	57.59 / 05.38
	L13	42.23 / 07.82	00.63 / 00.55	58.02 / 07.84	44.07 / 07.53
	L15	48.72 / 09.36	00.17 / 02.12	**54.34 / 10.00**	37.60 / 08.48
	L19	35.42 / 08.65	00.05 / 00.40	29.27 / 07.85	15.36 / 05.39
	L25	05.73 / 03.39	00.22 / 00.52	05.08 / 03.24	04.88 / 03.19
Prompt	-	**93.09 / 18.40**	**93.09 / 18.40**	**93.09 / 18.40**	**93.09 / 18.40**

Table 4: Comparison of results across Llama-Scope, and Our-Llama-SAEs (LangID/Bleu).

E Language vectors correlations and sparsity score

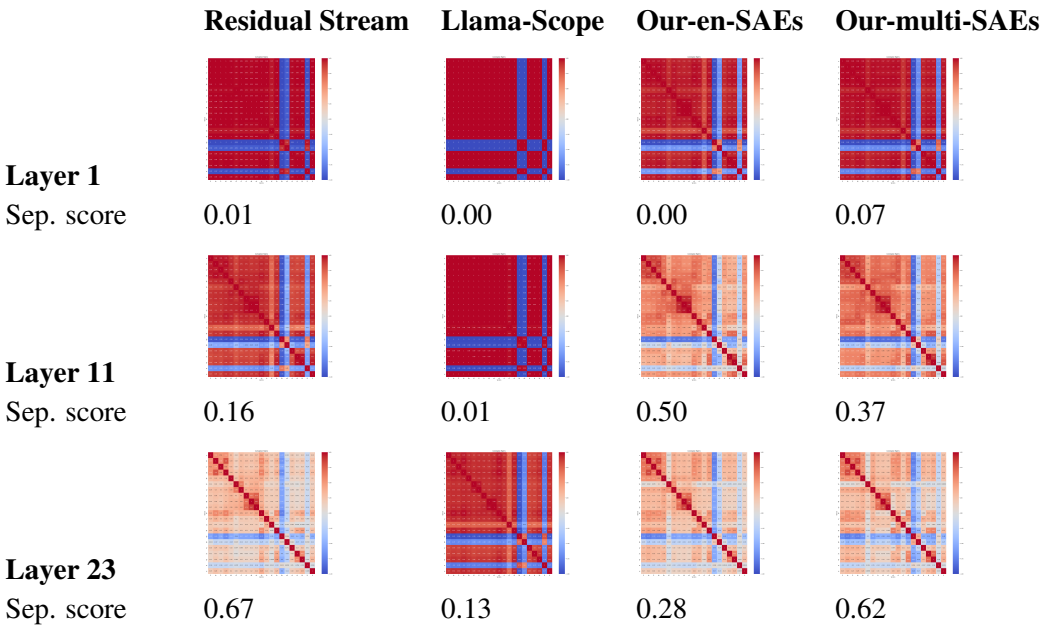

Figure 10: Comparison of LLama-3.1-8B model representation space using residual stream vectors, LLama-Scope sparse space vectors. and our trained SAEs sparse space vectors.

I Gemma: Benchmarking SAE Variants with BLEU and LangID

Method	Layer	Resid	Gemma-Scope	Our-multi-SAEs	Our-en-SAEs
Diffmean	L6	71.33 / 07.74	50.52 / 02.03	67.59 / 11.22	68.81 / 11.57
SA	L14	72.06 / 09.93	48.46 / 02.72	72.59 / 11.67	71.76 / 12.64
	L17	69.69 / 11.52	56.66 / 06.08	**70.58 / 12.90**	70.36 / 13.47
	L22	**67.31 / 12.02**	65.17 / 11.72	57.96 / 11.34	52.76 / 10.49
	L23	67.65 / 11.51	65.80 / 10.56	**68.36 / 13.03**	66.05 / 12.91
	32	49.57 / 09.72	46.63 / 09.41	38.06 / 08.45	24.53 / 06.33
Prompt	-	**79.59 / 17.35**	**79.59 / 17.35**	**79.59 / 17.35**	**79.59 / 17.35**

Table 5: Comparison of results across Gemma-Scope, and Our-Gemma-SAEs (LangID/Bleu).

F Gemma: BLEU reconstruction loss per-language

Figure 11: BLEU scores after adding one SAE one layer at a time and prompting for translation using the same prompt, showcasing the high translation errors caused by Gemma-Scope reconstruction.

G Llama: BLEU reconstruction loss per-language

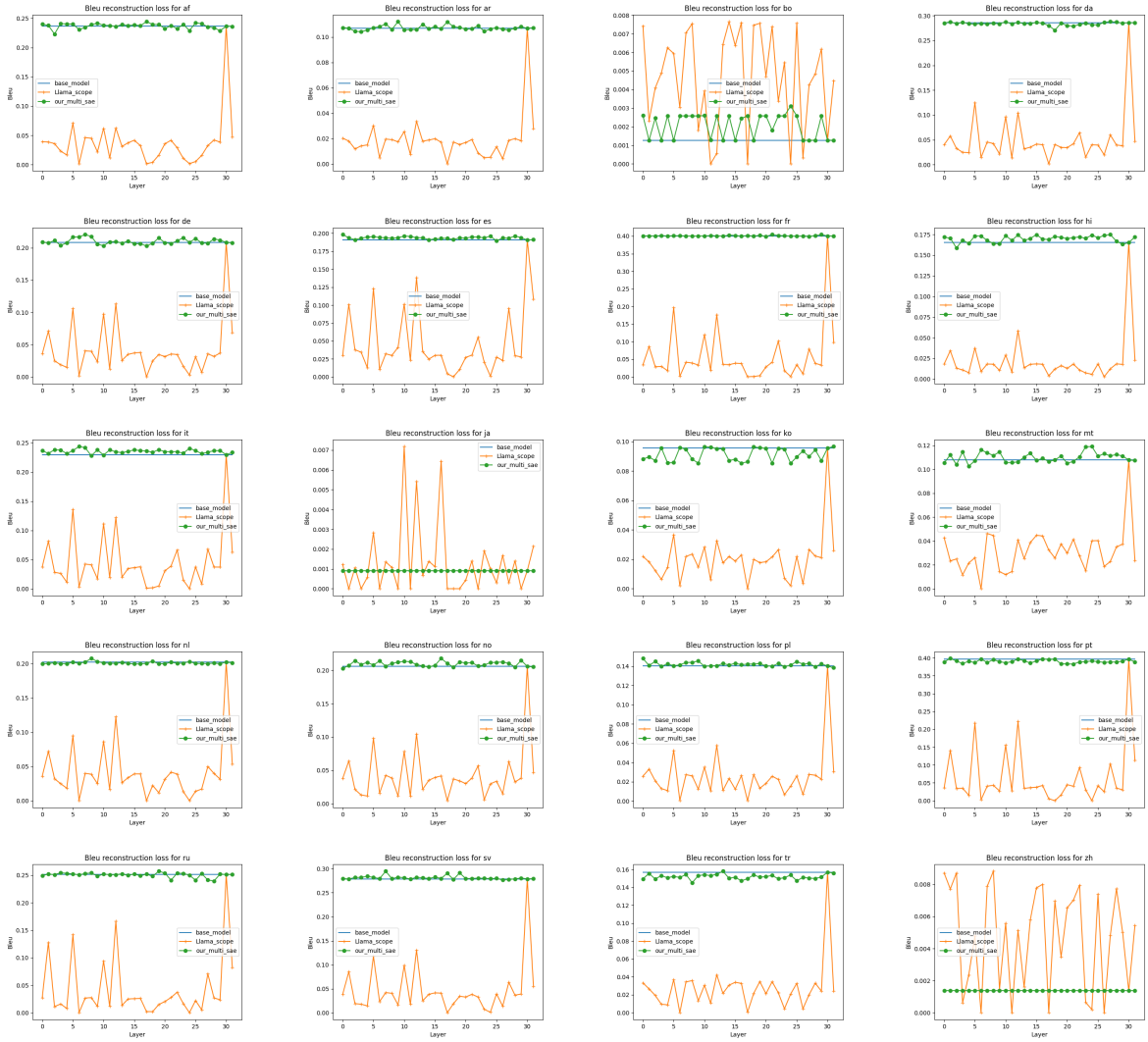

Figure 12: BLEU scores after adding one SAE one layer at a time and prompting for translation using the same prompt, showcasing the high translation errors caused by Llama-Scope reconstruction.

J Per-Language Results

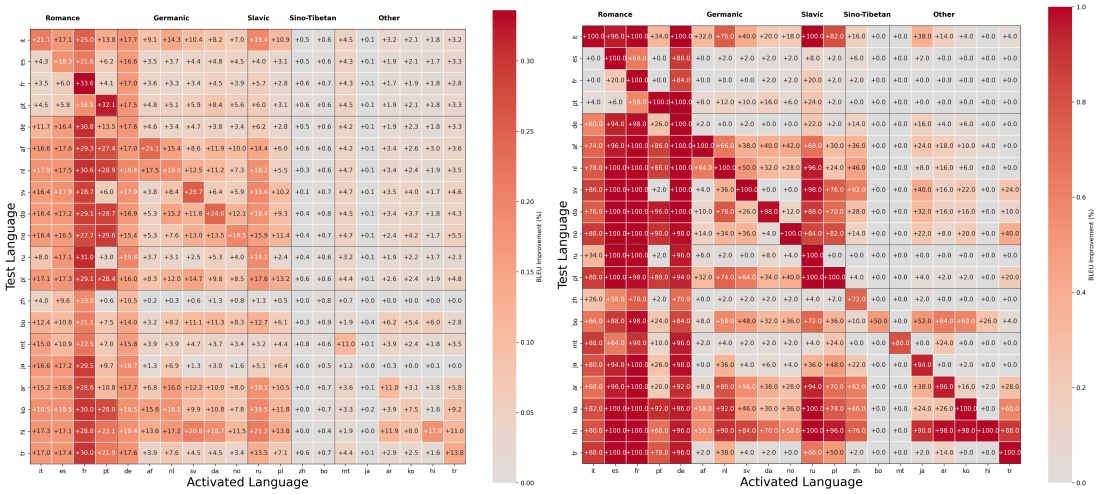

Figure 13: Llama (En-SAEs): Layer 15 per language results. Left is BLEU score and right is LangID

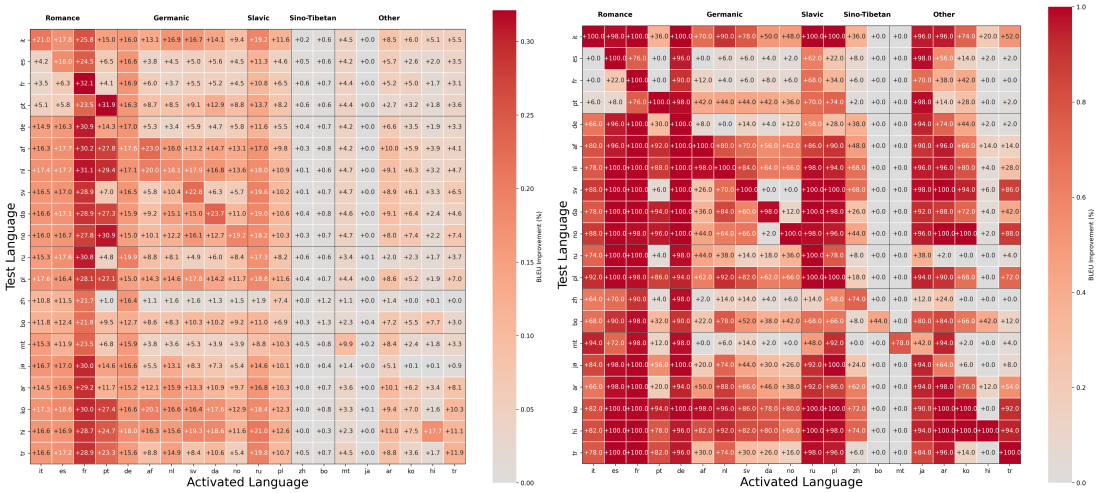

Figure 14: Llama (Multi-SAEs): Layer 15 per language results. Left is BLEU score and right is LangID

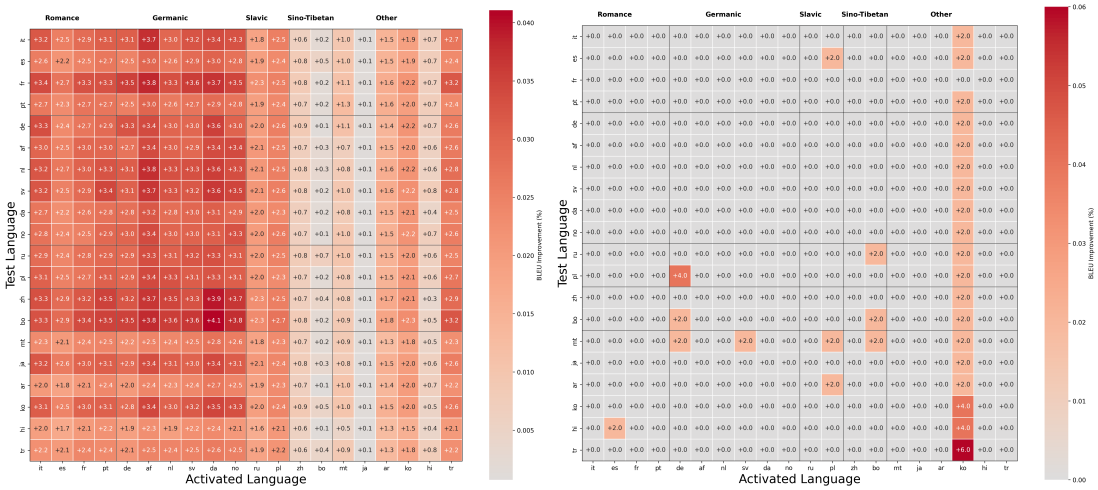

Figure 15: Llama (Llama-Scope): Layer 15 per language results. Left is BLEU score and right is LangID

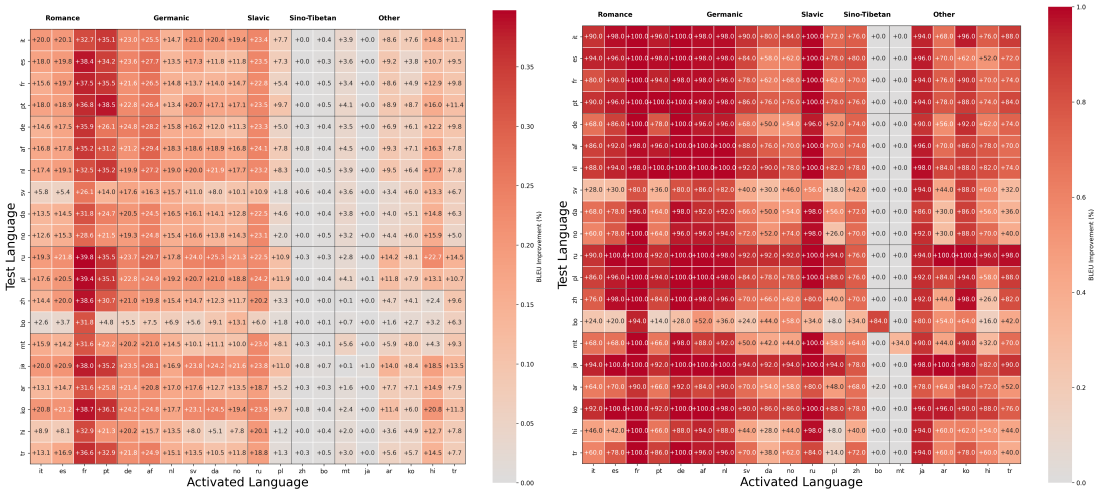

Figure 16: Gemma (En-SAEs): Layer 17 per language results. Left is BLEU score and right is LangID

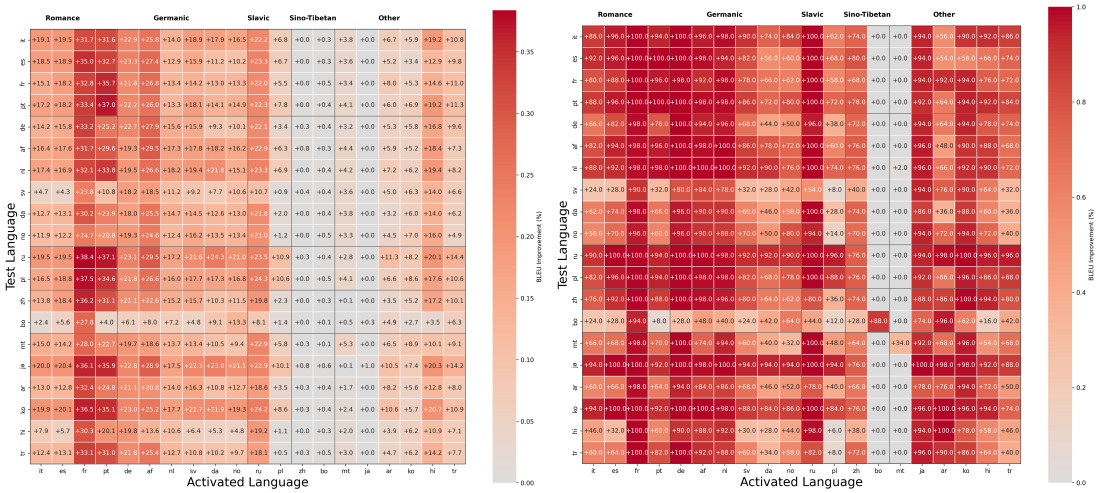

Figure 17: Gemma (Multi-SAEs): Layer 17 per language results. Left is BLEU score and right is LangID

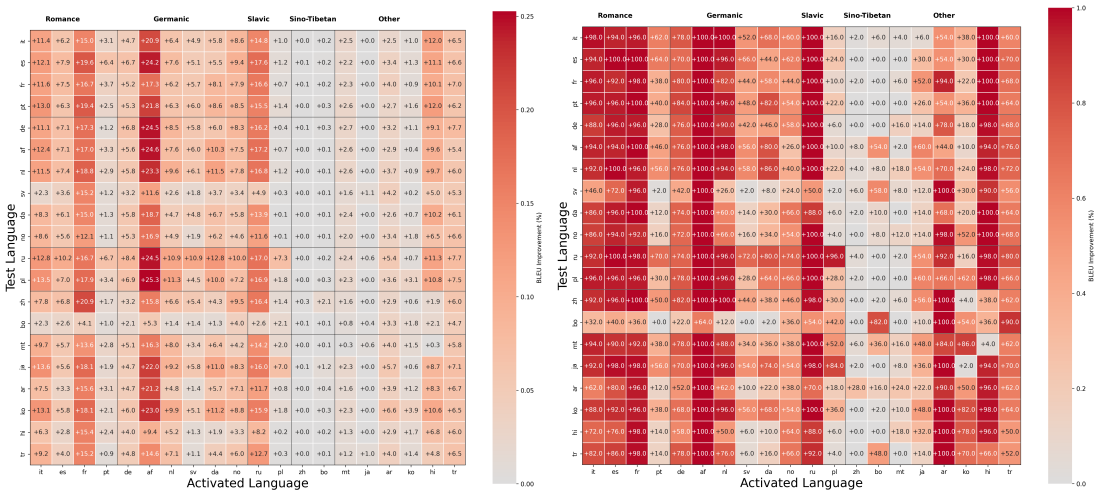

Figure 18: Gemma (Gemma-Scope): Layer 17 per language results. Left is BLEU score and right is LangID

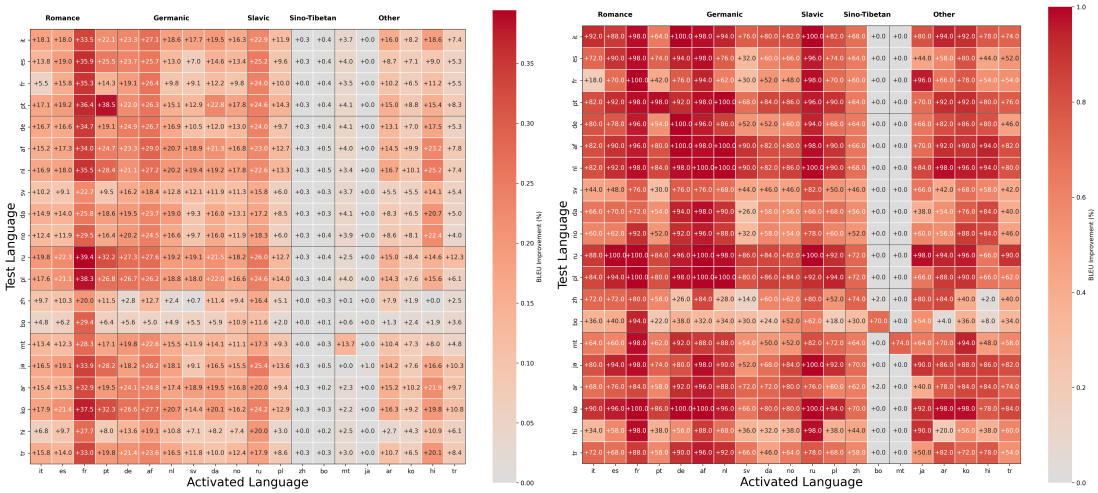

Figure 19: Gemma (En-SAEs): Layer 23 per language results. Left is BLEU score and right is LangID

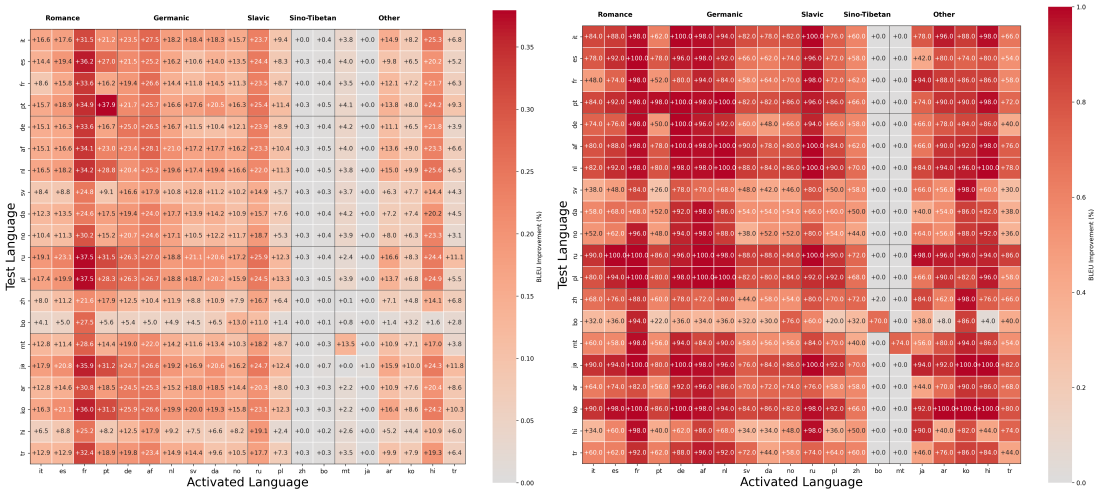

Figure 20: Gemma (Multi-SAEs): Layer 23 per language results. Left is BLEU score and right is LangID

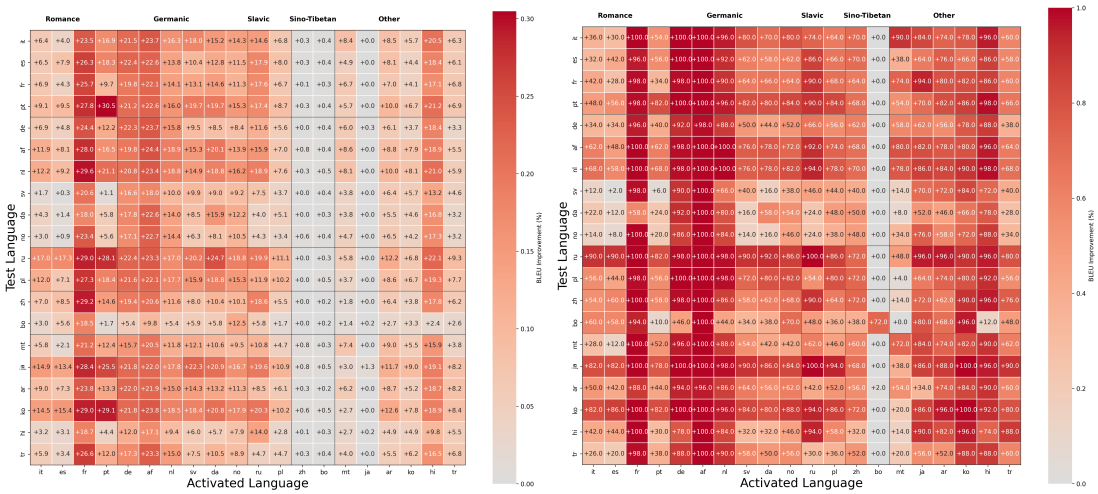

Figure 21: Gemma (Gemma-Scope): Layer 23 per language results. Left is BLEU score and right is LangID