Feature-Aware Training with Sparse Autoencoder Features for Multilingual Language Control

Anonymous ACL submission

Abstract

Controlling the output language of multilingual language models via activation-level interventions has shown promising results, but often comes at the cost of generation instability. We investigate whether sparse autoencoder (SAE) features associated with specific languages can be incorporated into training-time objectives to achieve more stable control, and propose *feature-aware supervised fine-tuning*, which integrates feature activation guidance with standard language modeling objectives and distributional regularization. Across several model families and languages, we find that feature-aware training yields limited but consistent improvements in language controllability, while reducing collapse and preserving fluency compared to inference-time steering. Our results reveal a clear trade-off between controllability and stability, and suggest that training-time feature alignment can help regularize this behavior in multilingual language models.

1 Introduction

Activation-level steering has emerged as a promising approach for controlling language model behavior by manipulating internal representations (Chalnev et al., 2024a; Chou et al., 2025; Turner et al., 2024; Rimsky et al., 2024). A recent line of work leverages *Sparse Autoencoders* (SAEs) to decompose model activations into a set of interpretable and sparsely activated internal features, enabling targeted interventions at the representation level.

However, current steering methods exhibit severe instability: interventions that successfully amplify target features often cause output distribution collapse, producing repeated punctuation, character loops, or complete fluency breakdown (Zwirner et al., 2025). This brittleness is especially severe in multilingual models, where language-specific features are highly entangled in shared representations (Chou et al., 2025). A key source of this instability is a mismatch between intervention space and training objective (Arad et al., 2025). SAE features operate in the model's residual stream as intermediate activation patterns, whereas language models are trained via next-token prediction objectives that optimize token-level likelihoods. As a result, naively amplifying language-related features may improve activation while producing nonsensical or degenerate text.

In this paper, we propose *feature-aware supervised fine-tuning*, which jointly optimizes SAE feature activation and constraints on the output distribution, ensuring that increases in target feature activation do not come at the cost of fluent and diverse generation.

2 Related work

Our work sits at the intersection of mechanistic interpretability, activation steering, and multilinguality. We review relevant literature in these areas to motivate our feature-aware training objective.

2.1 Sparse Autoencoders for Interpretability

SAEs have emerged as a powerful technique to disentangle these representations by decomposing dense activations into a sparse linear combination of interpretable features (Bricken et al., 2023; Huben et al., 2024). Recent scaling efforts have successfully applied SAEs to large language models (LLMs), creating a set of sparse and interpretable features that facilitate fine-grained analysis of internal model representations (Stolfo et al., 2025).

2.2 Activation Steering and Language Control

Activation steering (or representation engineering) aims to control model output by intervening on internal hidden states during inference (Turner et al., 2024; Chalnev et al., 2024b). By adding a specific *steering vector* to the residual stream, prior work

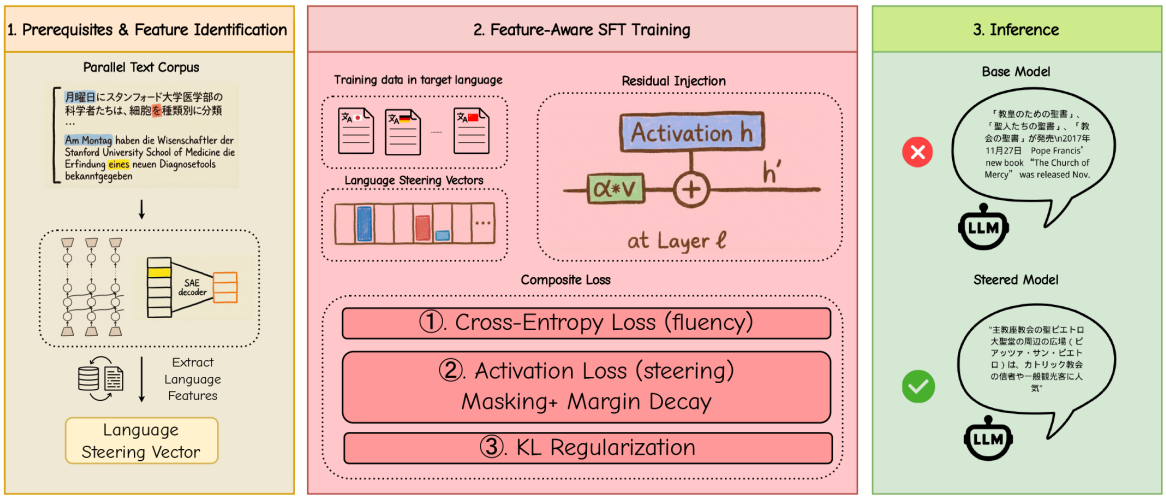

Figure 1: Overview of our framework. (**Left**) We identify language-specific features by applying SAEs to model activations on a parallel multilingual corpus, and aggregate them into language steering vectors. (**Middle**) During feature-aware supervised fine-tuning, steering vectors are injected into the residual stream at a selected layer, while a composite loss jointly optimizes fluency (cross-entropy), language-specific feature activation, and output distribution regularization. (**Right**) At inference time, the trained model produces target-language outputs without requiring high-strength activation-level interventions.

has successfully elicited behaviors ranging from sentiment control to truthfulness (Rimsky et al., 2024; Studdiford et al., 2025).

In the context of multilingualism, recent studies have identified subspaces or directions that encode language identity (Kojima et al., 2024). Zwirner et al. (2025) and Chou et al. (2025) extended this to SAE-based steering, demonstrating that amplifying language features can force models to switch output languages. However, inference-time steering is notoriously brittle; excessive intervention strength often pushes activations off-manifold, leading to distribution collapse or repetition loops (Zwirner et al., 2025).

2.3 Internalizing Steering into Weights

While standard Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL) optimize token-level output probabilities, they do not explicitly constrain the internal processes that generate these outputs. Many studies explore *white-box* adaptation methods that utilize internal representations during training. For instance, latent space regularization has been used to minimize catastrophic forgetting or enforce semantic constraints (Li et al., 2022). Closer to our approach, recent investigations suggest that successful steering vectors can be baked into model weights via fine-tuning on steered data or orthogonalizing weight matrices (Wu et al., 2024; Qiu et al., 2025; Ngugi, 2025).

3 Methodology

We propose a feature-based steering framework that biases multilingual language models toward a target language by intervening in their hidden states.

Our method proceeds in three stages: (1) identifying language-specific latent features with SAEs, (2) constructing steering directions from these features, and (3) performing feature-aware supervised fine-tuning to achieve stable language steering.

3.1 Language Feature Identification

Our first goal is to identify internal features that reliably indicate the presence of a specific language. We leverage pretrained SAEs (He et al., 2024; Lieberum et al., 2024; Kissane et al., 2024), which decompose model activations into a set of sparse and interpretable latent features, many of which correspond to semantically meaningful concepts (Bricken et al., 2023; Huben et al., 2024). We treat each SAE feature as a candidate indicator for language identity.

To quantify the association between a feature f and a target language L, we use Pointwise Mutual Information (PMI) (Church and Hanks, 1990). Feature activations are first binarized using a fixed threshold, and PMI is computed as

$$\mathrm{PMI}(f, L) = \log \frac{P(f \mid L)}{P(f)}, \qquad (1)$$

where $P(f \mid L)$ denotes the fraction of samples in language L whose binarized activation of feature f exceeds the threshold (see Appendix B), and $P(f)$ is the corresponding trigger probability estimated on a baseline corpus. Here, the baseline corpus refers to a reference corpus used to estimate the marginal activation frequency of feature f. We consider two choices for the baseline corpus: (i) English-only data, and (ii) the union of all languages used in our experiments. We found that within the selected feature set, PMI rankings are nearly identical when computed using English or the set of languages considered in our experiments as the baseline. Based on this preliminary comparison, we adopt English as the baseline corpus in all subsequent experiments.

However, frequency-based metrics (including PMI) alone do not guarantee language specificity, especially among closely related languages (Zwirner et al., 2025). To address this, we apply a uniqueness constraint that removes any feature that also appears among the top candidates of other languages, ensuring that the remaining set is language-exclusive.

3.2 Steering Vector Construction

Given the final set of language-specific feature indices $\mathcal{F}$ and a preselected intervention layer (Table 1), we construct a steering vector that intervenes in the model's residual stream. Let the SAE decoder be $W_{\text{dec}} \in \mathbb{R}^{H \times K}$, where H is the model hidden dimension and K is the number of SAE features. For a feature index $f \in \{1, \ldots, K\}$, we denote its decoder direction as the corresponding decoder column $\mathbf{d}_f = W_{\text{dec}}[:, f] \in \mathbb{R}^H$. We then aggregate the selected directions and normalize to obtain a unit steering direction:

$$\tilde{\mathbf{v}} = \sum_{f \in \mathcal{F}} \mathbf{d}_f, \qquad \mathbf{v} = \frac{\tilde{\mathbf{v}}}{\|\tilde{\mathbf{v}}\|_2}. \tag{2}$$

During training, we inject the steering direction into the residual stream at a preselected transformer layer ℓ via an additive intervention:

$$\mathbf{h}_\ell \leftarrow \mathbf{h}_\ell + \alpha \mathbf{v}, \tag{3}$$

where $\mathbf{h}_\ell \in \mathbb{R}^H$ is the residual stream activation at layer ℓ, and $\alpha \in \mathbb{R}$ controls the steering strength. Rather than using a fixed value for α, we randomly sample α for each batch, including a non-zero probability of setting $\alpha = 0$. This regularizes the intervention and helps preserve the model's original behavior when no steering is applied.

3.3 Feature-Aware Supervised Fine-Tuning

Overview. A central difficulty in language steering is that naively maximizing feature activation often leads to degenerate generations (Zwirner et al., 2025).

To address this issue, we introduce *feature-aware supervised fine-tuning*, a supervised training paradigm that augments standard next-token prediction with feature-level constraints.

Training is performed on multilingual text-only data covering the target languages, without requiring task-specific annotations, as supervision is fully specified by the loss functions.

Feature-level activation objective. At a designated layer, we extract token-level residual stream activations h_t and encode them using a pretrained SAE, yielding feature activations $z_t = \text{SAE}(h_t)$, where $z_t[f]$ denotes the activation of feature f at token position t. The SAE is used solely for feature readout.

For a given steering strength α, we define a hinge-style activation loss that enforces a minimum average activation over the target feature set $\mathcal{F}$:

$$\mathcal{L}_{\text{act}} = \max\left(0, m_0\left(1 - \frac{\alpha}{\alpha_{\max}}\right) - \frac{1}{T}\sum_{t=1}^{T}\sum_{f \in \mathcal{F}} z_t[f]\right), \tag{4}$$

where T denotes the sequence length, m_0 is the initial margin specifying the minimum required average activation at zero steering strength, and $\alpha_{\max}$ is the maximum steering strength. The margin decay reflects the fact that stronger steering naturally induces higher feature activations.

Motivated by the instability of excessive feature amplification, the hinge loss enforces sufficient activation without encouraging over-activation.

Overall training objective. To preserve generation quality and distributional stability, we combine the activation loss with standard cross-entropy and KL regularization. The final objective is a weighted sum of the three losses, with uncertainty-based weighting (Kendall et al., 2018):

$$\mathcal{L} = \sum_i \frac{1}{2\sigma_i^2}\mathcal{L}_i + \frac{1}{2}\log \sigma_i^2, \tag{5}$$

where i indexes the three objectives and σ_i^2 are learnable noise variances that adaptively balance their relative contributions during optimization.

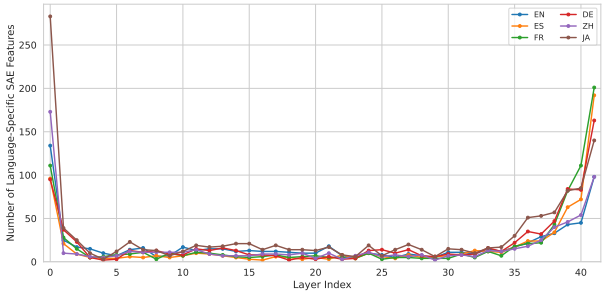

Figure 2: Layer-wise distribution of language-specific SAE features in Gemma-2-9B.

Model	Target language	Selected layer
GPT-2	de	3
	es	3
	fr	2
	ja	1
	zh	1
Llama-3.1-8B	de	0
	es	30
	fr	31
	ja	0
	zh	31
Gemma-2-9B	de	41
	es	41
	fr	41
	ja	41
	zh	41

Table 1: Selected transformer layer for feature intervention across models and target languages.

4 Experiments

We evaluate the potential of SAE-based features for controllable multilingual generation. Our study spans multiple model families (GPT-2 (Radford et al., 2019), Gemma-2 (Riviere et al., 2024), and Llama (Grattafiori et al., 2024)) and five languages (German, French, Spanish, Chinese, and Japanese). We organize our experiments as a three-stage investigation. First, we identify and validate language-specific features on a multilingual corpus (FLORES+ (Costa-jussà et al., 2022)) (Section 4.1). Second, we demonstrate that while these features enable steering, direct inference-time intervention leads to instability (Section 4.2). This motivates our feature-aware training objective, which we then evaluate against baselines and analyze through ablations and side-effect studies. Throughout our experiments, we compare feature-aware training against pretrained base models and inference-time activation steering, and evaluate performance using language accuracy, perplexity, and fluency metrics, as reported in the subsequent sections.

4.1 Feature Analysis

Following the procedure in Section 3.1, we analyze the architectural distribution and activation specificity of the extracted language-specific features.

Layer-wise Distribution. Analysis in Figure 2 reveals that language-related features are predominantly concentrated in the extreme (early and late) layers across all model families. This pattern aligns with prior observations that language identification and formatting tend to emerge at the periphery of the residual stream (Wendler et al., 2024). Based on this distribution, we select the layers with the highest feature density for subsequent intervention experiments, as summarized in Table 1.

Activation Specificity. At the individual feature level, we observe a hierarchy of specificity. Top-ranked features act as highly specific language switches, activating almost exclusively on the target language. In contrast, lower-ranked features exhibit broader semantic overlap; for instance, certain Japanese-associated features show residual activations on Chinese inputs, likely reflecting shared character sets or structural similarities.

Out-of-Domain Validation We further validate the language specificity of the identified features using an out-of-domain (OOD) contrastive test on non-parallel Wikipedia text[1]. For each target language L, we compare the mean activation of its language-specific feature set $\mathcal{F}_L$ on OOD text written in L against the average activation of feature sets associated with other languages.

Formally, if the features capture genuine language-related structure rather than dataset-specific artifacts, we expect the following inequality to hold:

$$A(\mathcal{F}_L \mid L\text{-text}) \gg A(\mathcal{F}_k \mid L\text{-text}), \quad k \neq L.$$

As shown in Figure 3, this inequality consistently holds across all evaluated languages, with Own/Other activation ratios ranging from $1.22\times$ to $2.50\times$. The effect is strongest for languages such as Chinese and Japanese, while English exhibits a smaller but still positive margin, consistent with prior findings that English often occupies a more central or shared representational space in multilingual language models. These validated features

[1]Wikipedia articles from the Wikimedia dumps (November 2023 snapshot).

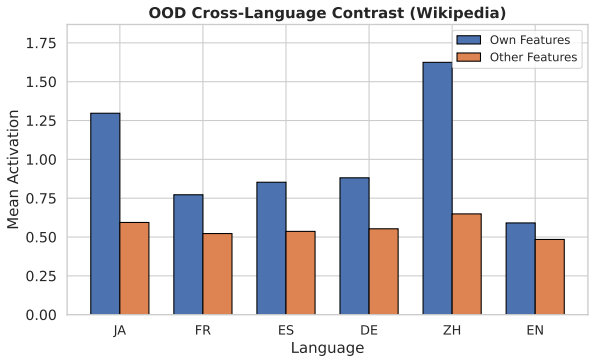

Figure 3: Out-of-domain validation on Wikipedia text. For each language, we report the mean activation of its language-specific SAE features on Wikipedia text written in the same language (Own Act.) and on text from other languages (Other Act.)

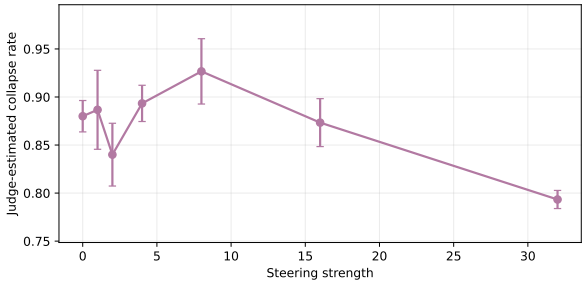

Figure 4: Effect of inference-time steering strength on generation stability. We report the proportion of collapsed outputs, as judged by GPT-4o.

provide the necessary 'control knobs' for multilingual steering, the efficacy of which we examine in the next section.

4.2 Language Steering via Feature Injection (Baseline)

To establish a baseline and verify the functional relevance of the features identified in Section 4.1, we first examine inference-time steering using these identified language-specific features.

Following prior work, we perform unprompted generation by injecting a language steering vector into the residual stream at a fixed layer, using only the beginning-of-sequence (BOS) token as input (Kojima et al., 2024). For each target language, we measure the proportion of outputs classified as the target language using an automatic language identification model.

Consistent with previous observations, inference-time feature steering reliably biases generation toward the desired target language, confirming that the identified features retain their steering effect under our experimental setup (Zwirner et al., 2025).

However, Figure 4 reveals a critical limitation: as steering strength increases to achieve higher accuracy, the model's generation stability collapses. This suggests that naive injection disrupts the residual stream, necessitating a more principled approach to integrate these features into the model's weights.

4.3 Proposed Method: Feature-Aware Training

To resolve the stability issues identified in Section 4.2, we propose a training objective that internalizes language-specific features while preserving the original generation ability of the base model.

Training Objective. Our training objective aims to increase the activation of specific language-related hidden-state patterns through controlled interventions, while avoiding undesirable shifts in the model's original knowledge and general language modeling ability. To achieve this, we optimize a combination of three loss terms.

- **Cross-Entropy Loss**: To preserve general language modeling ability and output fluency.
- **Feature Activation Loss**: Encourages the activation of language-specific SAE features.
- **KL divergence**: Regularizes the output distribution toward that of the pretrained model to prevent degeneration.

Because the relative importance of these objectives is unclear, we combine the losses using uncertainty-based weighting, allowing the model to automatically adjust their contributions during training (Kendall et al., 2018). We observe that the weight assigned to the CE term gradually decreases during training, while the activation and regularization terms gain importance, although the overall margin between the three remains small. This behavior reflects a gradual shift from preserving the original distribution toward selectively exploiting language-specific representations.

Because the objective guides the model to better utilize existing language representations rather than acquire new linguistic knowledge, only a small amount of training data is required for convergence.

Intervention and Sampling Strategy. Feature interventions are applied at the transformer layer where language-specific features exhibit the highest

quality. Layer selection is determined by quantitative criteria, including feature coverage and the number of language-specific features, providing a reproducible measure of feature reliability.

During training, a language steering vector is additively injected into the residual stream. The steering strength is randomly sampled for each batch and includes a non-zero probability of zero steering. This stochastic strategy prevents overfitting to a single intervention pattern and promotes robustness to both steered and unsteered inputs. Individual features are weighted according to their quality, so that more reliable language features contribute more strongly to the steering signal.

Embedding Freezing Although updating input embeddings can increase representational flexibility, our goal is not lexical adaptation but to isolate the effects of feature-level interventions. Allowing embedding updates introduces additional degrees of freedom that make controlled feature activation harder to analyze and often leads to unstable training or reward hacking (Shihab et al., 2025; Gu et al., 2025; Ziarko et al., 2024).We therefore freeze the input embedding layer in all experiments unless otherwise specified.

With this training framework established, we now turn to a systematic evaluation of its performance.

4.4 Results

We evaluate performance along two dimensions: language controllability and generation quality.

4.4.1 Language Controllability

We first evaluate language controllability, asking whether different methods can reliably bias generation toward a desired target language under weak prompting conditions, without degrading base performance.

Language controllability is measured as language accuracy using the fastText (Joulin et al., 2017) language identification classifier. While language identification on short continuations can be noisy, fastText provides a consistent proxy that enables large-scale comparison across models and methods. Accordingly, we interpret language accuracy trends comparatively rather than as absolute performance measures.

To provide minimal contextual guidance while leaving room for deviation, we construct prompts by truncating parallel corpus sentences to their initial 20% tokens to maximize the reliance on the model's internal steering signals rather than external context. This truncated continuation setting allowing us to isolate language control effects under weak prompting rather than to model realistic user prompts.

Model	Lang	Base	Steer	Ours
GPT-2	ja	63%	77%	**84%**
	zh	55%	**61%**	52%
	fr	75%	76%	**79%**
	de	68%	71%	**85%**
	es	73%	67%	**88%**
Llama-3.1	de	87%	**95%**	87%
	ja	82%	**99%**	82%
	es	88%	88%	88%
	fr	89%	**92%**	89%
	zh	91%	**93%**	91%
Gemma-2	de	100%	100%	100%
	es	100%	99%	100%
	fr	100%	99%	100%
	ja	100%	99%	100%
	zh	99%	99%	99%

Table 2: Language accuracy (%) on the FLORES+ continuation task. Base: pretrained model; Steer: inference-time steering; Ours: feature-aware SFT.

For large multilingual models, language accuracy on this task is often already near saturation, making further improvements difficult to observe in absolute terms. We therefore focus on whether feature-aware training preserves this behavior while enabling explicit control mechanisms.

As shown in Table 2, inference-time steering often achieves the highest language accuracy, particularly for large multilingual models, but this comes with significant stability issues (see Table 3). In contrast, feature-aware training internalizes language control into the model parameters, preserving base-level language accuracy on strong models while substantially improving controllability for weaker models such as GPT-2.

4.4.2 Generation Quality and Stability

Table 3 summarizes the generation quality. Most notably, our method resolves the stability issues of inference-time steering, reducing collapse rates to negligible levels (0.0% for Llama and Gemma) while maintaining high lexical diversity.

Collapse is defined as degenerate generation patterns such as repetitive symbols, punctuation loops, or non-linguistic sequences. Fluency is rated on a 1–5 Likert scale by the same judge, with higher scores indicating more natural text. Judgments are obtained using a fixed evaluation prompt with GPT-4o, applied uniformly across all methods.

Model	Method	PPL ↓	Distinct-2 ↑	Collapse ↓	Fluency ↑
GPT-2	Base	**65.45**	0.918	1.2%	**1.04**
	Inference Steer	-	0.916	**0.4%**	1.03
	Ours	65.48	**0.936**	0.8%	1.02
Llama-3.1	Base	13.77	0.909	0.4%	3.17
	Inference Steer	-	0.826	3.6%	2.54
	Ours	**11.73**	**0.911**	**0.0%**	**3.29**
Gemma-2	Base	22.14	0.906	0.2%	3.77
	Inference Steer	-	0.907	0.1%	3.76
	Ours	**18.12**	**0.911**	**0.0%**	**3.79**

Table 3: Generation quality averaged across 5 languages (de, es, fr, ja, zh). PPL: geometric mean of perplexity; other metrics: arithmetic mean. Inference steering does not affect perplexity.

In terms of fluency and perplexity, we observe improvements across most multilingual models. For GPT-2, the gains are comparatively limited. This inconsistency can be partly attributed to its vocabulary and tokenizer, which provide only sparse coverage of non-English languages. We note that the magnitude of the aggregated gains is somewhat dampened by outlier performance in specific language directions, where feature separability remains a bottleneck.

Despite these localized variances, the overall trend suggests that our objective internalizes control without degrading the model's general capabilities.

4.5 Ablation Study

We analyze the contribution of each training objective by removing them in isolation. Table 4 summarizes the results across models, reporting language accuracy, fluency, and perplexity.

Training with cross-entropy alone preserves fluent generation but provides limited incentives for inducing language-specific control. Optimizing feature activation alone directly targets internal representations, but without additional constraints, it is underconstrained with respect to the output distribution and can lead to degraded performance, particularly in smaller models. Removing KL regularization further amplifies this effect by allowing overly aggressive feature optimization.

The full objective balances cross-entropy supervision, feature activation, and KL regularization, resulting in more stable improvements across language accuracy, fluency, and perplexity.

4.6 Cross-Lingual Side Effects

While feature-aware training improves controllability and stability for the target language, we observe non-trivial side effects on non-target languages.

Model	Method	Acc ↑	Flu. ↑	PPL ↓
GPT-2	CE only	0.56	1.08	**25.99**
	Feature only	0.46	1.08	26.41
	w/o KL	**0.58**	**1.12**	26.12
	Ours (Full)	0.54	1.10	26.02
Llama-3.1	CE only	0.67	2.56	11.56
	Feature only	0.88	3.06	9.46
	w/o KL	0.88	3.04	9.46
	Ours (Full)	**0.94**	**3.16**	**8.57**
Gemma-2	CE only	0.92	3.06	32.67
	Feature only	**1.00**	3.52	20.31
	w/o KL	**1.00**	3.64	20.12
	Ours (Full)	**1.00**	**3.66**	**20.06**

Table 4: Ablation study of training objectives. Acc: Language Accuracy; Flu.: Fluency (GPT-4o judged 1-5).

Specifically, although perplexity on the target language often improves, the language modeling performance on other languages can degrade substantially. This pattern is consistently observed across our evaluated models and language pairs.

This behavior indicate cross-lingual interference and is consistent with catastrophic forgetting induced by targeted fine-tuning (Li et al., 2024).

The severity of degradation is not uniform across languages. Closely related languages tend to be less affected. For instance, Chinese exhibits significantly smaller performance drops when Japanese is targeted, likely due to shared CJK representations and overlapping language features. In contrast, more distant languages suffer more severe degradation.

5 Discussion

Feature-aware training as representation reweighting Our results suggest that feature-aware supervised fine-tuning does not induce new linguistic knowledge, but primarily re-weights exist-

ing language representations already present in the pretrained model. For strong multilingual models such as Gemma and Llama, controllability improvements are therefore limited, while weaker models like GPT-2 benefit more from selectively amplifying latent language-specific features. This distinguishes feature-aware training from both prompt-based control and inference-time activation steering, which rely on explicit or high-strength interventions at generation time.

Why feature activation alone is an insufficient training signal Maximizing activation alone does not sufficiently constrain the desired output behavior, allowing the model to satisfy the objective through degenerate strategies such as repetition or low-entropy outputs that strongly activate features without producing coherent language.

This issue arises because SAE features operate at the representation level and are not directly tied to token-level likelihoods.

As a result, activation-only optimization can distort the output distribution or lead to collapse.

Outlier behaviors In both our study and prior work that adopts similar setups, it is common to observe that certain target languages exhibit substantially worse performance under intervention (Zwirner et al., 2025).

A plausible explanation is the incompleteness of the language set used during feature identification. Language-specific features are selected only from a limited set of target languages, without explicit negative constraints from unseen languages. As a result, features that are strongly associated with a target language may also respond to unmodeled languages, leading to representation overlap.

When such features are amplified during steering or training, these latent cross-language activations can become visible in generation, resulting in unexpected behaviors.

We view this as a limitation of the current feature selection pipeline, which relies on a fixed and relatively small language set. Expanding the language coverage during feature identification may help mitigate such outlier behaviors.

We leave a systematic investigation of these strategies to future work.

Why no reinforcement learning? We also experimented with reinforcement learning to further encourage internalization of feature-level objectives.

While RL sometimes improved surface-level language fluency, it failed to consistently increase activation of the target SAE features. We hypothesize that this is due to the difficulty of credit assignment when the optimization target is defined over internal representations rather than observable outputs.

Since general improvements in language fluency that works under all settings alone do not directly align with our goal, we decided not to incorporate RL into the final training pipeline.

6 Conclusion

We study whether language-specific internal features identified by SAEs can be used as training-time control signals for multilingual language models.

We show that guiding feature activation during fine-tuning improves target-language controllability while preserving generation quality, and avoids the instability observed in strong inference-time steering. Our results suggest that internal feature-level supervision provides a practical bridge between interpretability and controllable model adaptation.

Limitations

Our study has several limitations. First, our method assumes access to pretrained SAEs, which we view not as a limitation of the approach itself, but as a dependency on the current interpretability tooling ecosystem.

Second, feature selection is performed over a fixed and relatively small set of languages. Languages outside this set are not explicitly modeled as negatives, which may cause some features to respond to unseen or typologically related languages, as observed in certain outlier cases.

Third, our approach relies on the quality and granularity of SAE features. Although SAEs provide a useful abstraction, the learned features are not guaranteed to be perfectly disentangled, and some may encode mixed or non-linguistic patterns. We do not attempt systematic feature denoising or semantic decomposition in this work.

Finally, our experiments focus on SFT with relatively small models and datasets. While the results suggest data-efficient alignment, we do not evaluate scaling behavior or long-horizon training dynamics.

References

Dana Arad, Aaron Mueller, and Yonatan Belinkov. 2025. Saes are good for steering – if you select the right features. *Preprint*, arXiv:2505.20063.

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*.

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

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

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. *Preprint*, arXiv:2507.13410.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, and 19 others. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Jian Gu, Aldeida Aleti, Chunyang Chen, and Hongyu Zhang. 2025. A semantic-aware layer-freezing approach to computation-efficient fine-tuning of language models. *Preprint*, arXiv:2406.11753.

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. 2024. Llama scope: Extracting millions of features from llama-3.1-8b with sparse autoencoders. *Preprint*, arXiv:2410.20526.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *Preprint*, arXiv:1705.07115.

Connor Kissane, Robert Krzyzanowski, Joseph Isaac Bloom, Arthur Conmy, and Neel Nanda. 2024. Interpreting attention layer outputs with sparse autoencoders. *Preprint*, arXiv:2406.17759.

Takeshi Kojima, Itsuki Okimura, Yusuke Iwasawa, Hitomi Yanaka, and Yutaka Matsuo. 2024. On the multilingual ability of decoder-based pre-trained language models: Finding and controlling language-specific neurons. pages 6919–6971, Mexico City, Mexico.

Dingcheng Li, Zheng Chen, Eunah Cho, Jie Hao, Xiaohu Liu, Fan Xing, Chenlei Guo, and Yang Liu. 2022. Overcoming catastrophic forgetting during domain adaptation of seq2seq language generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5441–5454, Seattle, United States. Association for Computational Linguistics.

Hongyu Li, Liang Ding, Meng Fang, and Dacheng Tao. 2024. Revisiting catastrophic forgetting in large language model tuning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4297–4308, Miami, Florida, USA. Association for Computational Linguistics.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, Janos Kramar, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. pages 278–300, Miami, Florida, US.

Stanley Ngugi. 2025. Targeted lexical injection: Unlocking latent cross-lingual alignment in lugha-llama via early-layer lora fine-tuning. *Preprint*, arXiv:2506.15415.

Zeju Qiu, Simon Buchholz, Tim Z. Xiao, Maximilian Dax, Bernhard Schölkopf, and Weiyang Liu. 2025. Reparameterized llm training via orthogonal equivalence transformation. *Preprint*, arXiv:2506.08001.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019.

Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. 2024. Steering llama 2 via contrastive activation addition. pages 15504–15522, Bangkok, Thailand.

Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, and 178 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Ibne Farabi Shihab, Sanjeda Akter, and Anuj Sharma. 2025. Detecting and mitigating reward hacking in reinforcement learning systems: A comprehensive empirical study. *Preprint*, arXiv:2507.05619.

Alessandro Stolfo, Vidhisha Balachandran, Safoora Yousefi, Eric Horvitz, and Besmira Nushi. 2025. Improving instruction-following in language models through activation steering. *Preprint*, arXiv:2410.12877.

Zach Studdiford, Timothy T. Rogers, Siddharth Suresh, and Kushin Mukherjee. 2025. Evaluating steering techniques using human similarity judgments. *Preprint*, arXiv:2505.19333.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J. Vazquez, Ulisse Mini, and Monte MacDiarmid. 2024. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in English? on the latent language of multilingual transformers. pages 15366–15394, Bangkok, Thailand.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, Atticus Geiger, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. 2024. Reft: Representation finetuning for language models. *Preprint*, arXiv:2404.03592.

Alicja Ziarko, Albert Q. Jiang, Bartosz Piotrowski, Wenda Li, Mateja Jamnik, and Piotr Miłoś. 2024. Repurposing language models into embedding models: Finding the compute-optimal recipe. *Preprint*, arXiv:2406.04165.

Sebastian Zwirner, Wentao Hu, Koshiro Aoki, and Daisuke Kawahara. 2025. Sparse autoencoders as a tool for steering the output language of large language models. In *Proceedings of the Thirty-first Annual Meeting of the Association for Natural Language Processing*.

A Training Objective Details

Our feature-aware training objective augments standard supervised fine-tuning with explicit constraints on internal feature activations. The full objective consists of three components: cross-entropy loss for language modeling, feature activation loss for representation alignment, and KL regularization to preserve distributional stability. We provide the detailed formulations below for completeness and reproducibility.

Cross-Entropy Loss. The standard next-token prediction loss is included to preserve general language modeling ability:

$$\mathcal{L}_{\text{CE}} = -\frac{1}{T}\sum_{t=1}^{T} \log p_\theta(x_t \mid x_{<t}), \tag{6}$$

where p_θ denotes the model's predicted token distribution and x_t is the ground-truth token at position t. This term anchors training to the original language modeling objective and helps maintain fluency and coherence during feature-aware optimization.

KL Regularization. To prevent the fine-tuned model from drifting excessively from its pretrained behavior, we include a KL divergence term:

$$\mathcal{L}_{\text{KL}} = \frac{1}{T}\sum_{t=1}^{T} D_{\text{KL}}\big(p_{\text{ref}}(\cdot \mid x_{<t}) \,\|\, p_\theta(\cdot \mid x_{<t})\big), \tag{7}$$

where p_{ref} is the frozen pretrained model. This regularizer constrains the output distribution during training and mitigates known failure modes of activation-level optimization, such as repetition loops and distribution collapse.

Feature Activation Loss. The feature activation loss (defined in Section 3.3) encourages the model to activate a selected set of language-specific SAE features above a dynamic margin threshold. Rather than directly maximizing activation magnitude, this hinge-style formulation enforces a lower bound on average activation, which empirically provides a more stable trade-off between controllability and generation quality.

B Hyperparameters for Feature-Aware Training

Table 5 summarizes the key hyperparameters used across all experiments. Unless otherwise specified, the same configuration is applied to all models and languages.

The intervention layer is selected based on the layer-wise distribution of language-specific SAE features (Figure 2), choosing the layer with the highest feature concentration. Features are selected using PMI with an additional uniqueness filter to remove features shared across multiple languages (see Section 3.1 for details). We select a single intervention layer for each model–language pair, as summarized in Table 1. During training, the steering strength α is sampled independently for each batch, with a non-zero probability of applying no steering to preserve robustness to unsteered inputs.

Activation Threshold. For PMI computation, SAE feature activations are binarized using a fixed threshold of $\tau = 0.5$, i.e., a feature is considered active if its activation exceeds τ. We use the same threshold across all models and languages.

C Evaluation Details

We use a combination of heuristic rules and large language model judgments to evaluate generation quality and stability. Heuristic criteria are used for automatic pre-filtering and for illustrating typical collapse patterns, while all reported fluency and collapse metrics in the main results are obtained using GPT-4o under a fixed evaluation prompt.

C.1 GPT-4o Evaluation Prompt

The GPT-4o evaluation prompt is structured as follows:

```
You are evaluating the quality of machine-generated text continuations.

Given the following text, rate its fluency on a scale of 1-5:

  • 1: Incomprehensible or degenerate (repetition, symbols)
  • 2: Partially readable but with major errors
  • 3: Understandable but unnatural phrasing
  • 4: Fluent with minor issues
  • 5: Native-like fluency

Also indicate if the text exhibits collapse (Yes/No), defined as:

  • Repetitive punctuation or symbols
  • Character or word loops
  • Non-linguistic sequences

Text: [GENERATED_TEXT]

Respond in JSON format: {"fluency": <int>, "collapse": <bool>}
```

Hyperparameter	Value / Strategy
Initial margin (m_0)	0.5
Maximum steering strength ($\alpha_{\max}$)	100
Steering strength sampling	Uniform $[0, \alpha_{\max}]$
Probability of zero steering	0.1
Learning rate	1×10^{-5}
Batch size	8
Training steps	5000–10000
Embedding freezing	Enabled (all experiments)

Table 5: Hyperparameter settings for feature-aware supervised fine-tuning.

Language	Own Act.	Other Act.	Ratio
Japanese (ja)	1.30	0.59	2.18×
French (fr)	0.77	0.52	1.48×
Spanish (es)	0.85	0.54	1.59×
German (de)	0.88	0.55	1.59×
Chinese (zh)	1.62	0.65	2.50×
English (en)	0.59	0.48	1.22×

Table 6: Detailed OOD activation statistics for Llama-3.1 Layer 31 features.

D Out-of-Domain Validation Data

We compute mean SAE feature activations on non-parallel Wikipedia text to assess language specificity under distribution shift. For each target language L, we report the average activation of its associated feature set on Wikipedia text written in L (Own Act.) and on text written in other languages (Other Act.), as summarized in Table 6.

E PMI-Based Language Feature Selection

This appendix provides a precise algorithmic description of the PMI-based language-specific feature selection procedure used in Section 3.1. The algorithm is included for reproducibility and does not introduce additional modeling assumptions beyond those described in the main text.

We use the same activation threshold τ and baseline corpus choice as described in Appendix B.

Algorithm 1 Language-Specific Feature Selection via PMI and Uniqueness

Require: Pretrained SAE features $\{f_k\}_{k=1}^{K}$; language-specific corpus $\mathcal{D}_L$; baseline corpus $\mathcal{D}_{\text{base}}$; activation threshold τ

Ensure: Language-exclusive feature set $\mathcal{F}_L$

1: **for** each feature f_k **do**

2: Compute trigger rate

$$P(f_k \mid L) \leftarrow \frac{1}{|\mathcal{D}_L|} \sum_{x \in \mathcal{D}_L} \mathbb{I}\big[z(x, f_k) > \tau\big]$$

3: Compute marginal trigger rate

$$P(f_k) \leftarrow \frac{1}{|\mathcal{D}_{\text{base}}|} \sum_{x \in \mathcal{D}_{\text{base}}} \mathbb{I}\big[z(x, f_k) > \tau\big]$$

4: Compute association score

$$\text{PMI}(f_k, L) \leftarrow \log \frac{P(f_k \mid L)}{P(f_k)}$$

5: **end for**

6: Rank features by $\text{PMI}(f_k, L)$

7: Remove features that appear among top-ranked candidates of other languages

8: **return** remaining language-exclusive feature set $\mathcal{F}_L$

F Compute and Model Scale

F.1 Computational Resources

All experiments were conducted on 8 NVIDIA H200 GPUs. The training was performed using standard deep learning frameworks with distributed computing support where applicable.

F.2 Total Compute Budget

Our experimental setup involved training multiple model variants across different SAE configurations and target languages:

- **Model configurations**: 15 variants covering 3 model families, 5 languages
- **Total GPU hours**: Several hundred GPU hours across all experiments
- **Training regime**: Multi-epoch training with early stopping mechanisms, with training steps aligned across compared conditions for fair comparison.

G Qualitative Language-Steered Examples

We present representative generation examples to illustrate the qualitative effects of language steering under different settings.

G.1 GPT-2 Generation Examples

We provide representative generation examples from GPT-2 to qualitatively illustrate the effects of language steering under different settings. All examples are generated from an empty prompt (i.e., BOS continuation) to isolate unconditional language bias.

G.1.1 French Steering

Prompt: <BOS>

Base model (no steering):
Katherine O'Connell. This is a rush transcript. Copy may not be in its final form and must at least appear with the appropriate level of detail...

Inference-time activation steering:
ingre-cv plus d, dujmvre déléche une su-verfment réquunéd de vrémin c fuvert à jpe quautle'h·dulement rphuse...

Feature-aware training (ours):
Décor's French brand was named the first of three luxury footwear that will be released by Guilhem after a year in production. The line consists, d'art un pour élément par l'âtrez en découver du monde de la femme...

G.1.2 Japanese Steering

Prompt: <BOS>

Base model (no steering):
Katherine O'Connell. This is a rush transcript. Copy may not be in its final form and must at least appear with the appropriate level of detail...

Inference-time activation steering:
K you no It-, and NO) 剛ご修表（先、道・ブレスー物ゝノ）など。NY! & Y M 大ロウルンの（だいまたらとのさ）。の氾

Feature-aware training (ours):
Seed: No matter what, if a single Pokémon is an enemy or has one in its island. その虚、彼くにいときて激めを行現では思かなまする事がバルステロウマイブラストレーマング。