Sparse Autoencoder Features for Classifications and Transferability

Jack Gallifant[1,2†], **Shan Chen**[1,2,3†], **Kuleen Sasse**[4], **Hugo Aerts**[1,2,5], **Thomas Hartvigsen**[6], **Danielle S. Bitterman**[1,2,3§]

[†]Co-first authors, [§]Corresponding author: dbitterman@bwh.harvard.edu

[1]Harvard University, [2]Mass General Brigham, [3]Boston Children's Hospital, [4]Johns Hopkins University, [5]Maastricht University, [6]University of Virginia

Abstract

Sparse Autoencoders (SAEs) provide potentials for uncovering structured, human-interpretable representations in Large Language Models (LLMs), making them a crucial tool for transparent and controllable AI systems. We systematically analyze SAE for interpretable feature extraction from LLMs in safety-critical classification tasks[1]. Our framework evaluates (1) model-layer selection and scaling properties, (2) SAE architectural configurations, including width and pooling strategies, and (3) the effect of binarizing continuous SAE activations. SAE-derived features achieve macro F1 > 0.8, outperforming hidden-state and BoW baselines while demonstrating cross-model transfer from Gemma 2 2B to 9B-IT models. These features generalize in a zero-shot manner to cross-lingual toxicity detection and visual classification tasks. Our analysis highlights the significant impact of pooling strategies and binarization thresholds, showing that binarization offers an efficient alternative to traditional feature selection while maintaining or improving performance. These findings establish new best practices for SAE-based interpretability and enable scalable, transparent deployment of LLMs in real-world applications.

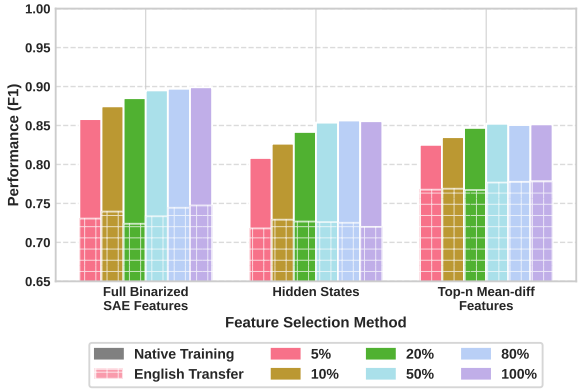


Figure 1: Multilingual performance comparison across three feature selection methods under varying training data sampling rates. Solid bars represent models trained on native language data, while hatched bars show performance with English transfer learning. Binarized SAE features demonstrate robustness across different training data constraints.

1 Introduction

Large language models (LLMs) have transformed natural language processing (NLP), demonstrating impressive performance on diverse tasks and languages, even in knowledge-intensive and safety-sensitive scenarios (Hendrycks et al., 2023; Ngo et al., 2025; Cammarata et al., 2021). However, the internal decision-making processes of LLMs remain largely opaque (Cammarata et al., 2021), raising concerns about trustworthiness and oversight, especially given the potential for deceptive or unintended behaviors. Mechanistic interpretability (MI), the study of the internal processes and representations that drive a model's outputs, offers a promising approach to address this challenge (Elhage et al., 2022; Wang et al., 2022). However, despite its potential, applying MI to real-world tasks presents significant challenges.

Sparse Autoencoders (SAEs) have recently emerged as a promising technique within MI for understanding LLMs. SAEs generally work by learning a compressed, sparse representation of the LLM's internal activations. This is achieved by up-projecting the dense hidden state of the LLM to a sparser, ideally monosemantic, representation (Bricken et al., 2023; Gao et al., 2024b). Identifying semantically meaningful features within LLMs using SAEs allows for deploying these features into explainable classification pipelines. This has the potential to boost performance and detect harmful biases or spurious correlations before they manifest in downstream tasks (Bricken et al., 2024). The ability to employ SAE features for classification across diverse settings, ranging from toxicity detection to user intent, offers a scalable form of "model

[1]Full repo: https://github.com/shan23chen/MOSAIC

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 29939–29963
November 4–9, 2025.

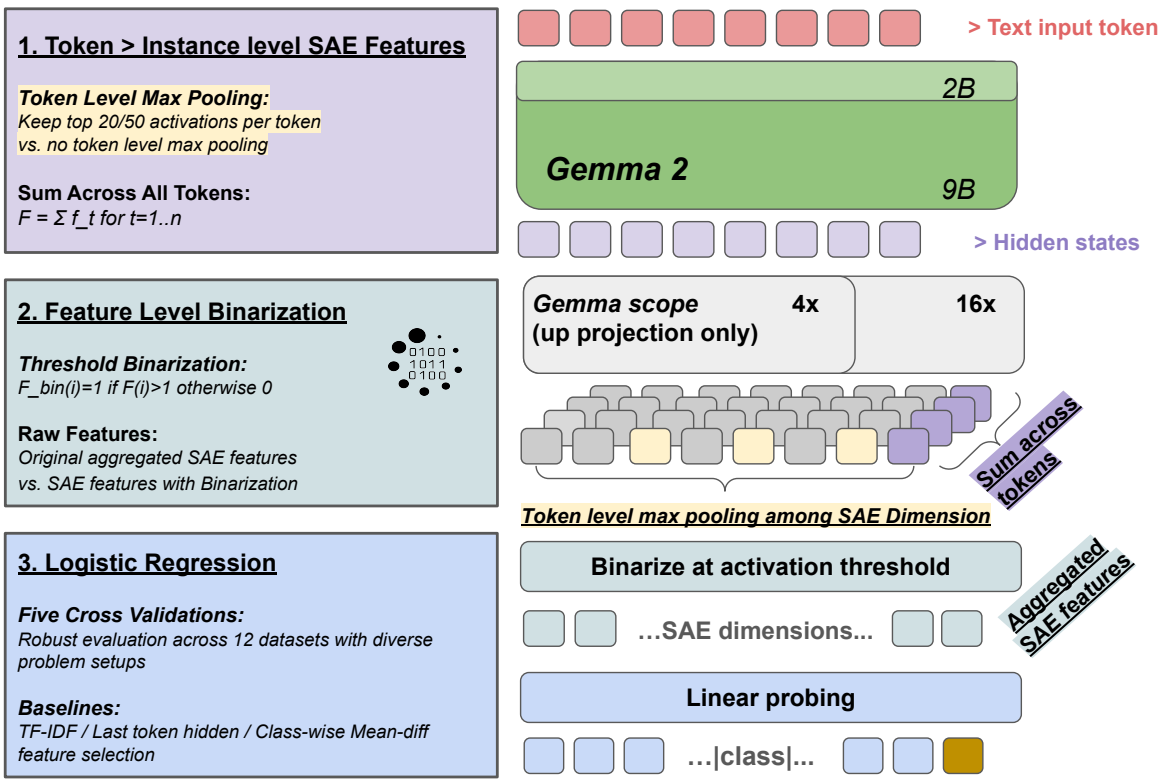


Figure 2: Diagram explaining our approaches to evaluating token-level pooling and aggregation of SAE features.

insight" (Bowman et al., 2022), which is crucial for building trust, safety, and accountability in high-stakes domains like medicine and law (Abdulaal et al., 2024).

Despite the promise of SAEs for MI, surprisingly few systematic studies have provided practical guidance on their use for classification. While promising results have been reported across various tasks (Bricken et al., 2024; Kantamneni et al., 2024; Chen et al., 2024), inconsistent experimental protocols, a lack of standardized benchmarks, and limited exploration of key architectural decisions hinder comparability and the development of best practices. Although tools like Transformer Lens (Nanda and Bloom, 2022) and SAE Lens (Joseph Bloom and Chanin, 2024) have improved standardization in sampling activations, critical questions about optimal configurations for diverse tasks, particularly in multilingual and multimodal settings, remain unanswered. This makes it challenging to establish the robustness and generalizability of SAE-based classification approaches.

This work directly addresses these limitations by providing a comprehensive and systematic investigation of SAE-based classification for LLMs. We introduce a reproducible pipeline for large-scale activation extraction and classification, enabling robust and generalizable conclusions. Specifically, we explore critical methodological choices, evaluate performance across diverse datasets and tasks, and investigate the potential for SAEs to facilitate model introspection and oversight (Figure 2).

Summary of Contributions

1. *Systematic Classification Benchmarks (Section 4, Part 1):* We introduce a robust methodology to evaluate and select SAE-based features in safety-critical classification tasks and show superior performance overall.
2. *Multilingual Transfer Analysis (Section 5, Part 2):* We analyze the cross-lingual transferability of SAE features in multilingual toxicity detection and show SAE features outperform everything in-domain and demonstrate potential on cross-lingual feature generalization.
3. *Behavioral Analysis and Model Oversight (Section 6, Part 3):* We extend SAE-based features to model introspection tasks, investigating whether LLMs can predict their own correctness and that of larger models, showing the potential of scalable model oversight.

2 Related Work

2.1 Interpretable Feature Extraction

MI has evolved from neuron-level analysis to sophisticated feature extraction frameworks (Olah et al., 2020; Rajamanoharan et al., 2024). Early approaches targeting individual neurons encountered fundamental limitations due to polysemanticity, where activation patterns span multiple, often unrelated concepts (Bolukbasi et al., 2021; Elhage et al., 2022). While techniques like activation patching (Meng et al., 2022) and attribution patching (Syed et al., 2023) offered insights into component-level contributions, they highlighted the need for more comprehensive representational frameworks.

SAEs address these limitations by providing more interpretable feature sets (Bricken et al., 2023; Cunningham et al., 2023). Recent scaling efforts have demonstrated SAE viability across LLMs from Claude 3 Sonnet (Templeton et al., 2024) to GPT-4 (Gao et al., 2024a) with extensions to multimodal architectures like CLIP (Bhalla et al., 2024). Although these studies have revealed interpretable feature dimensions and computational circuits (Marks et al., 2024; Zhao et al., 2024), they focus mainly on descriptive feature discovery rather than systematic evaluation of their downstream applications. Our work bridges this gap by providing standardized evaluation frameworks for SAE-based classification and cross-modal transfer, establishing quantitative metrics and methods for feature utility across diverse tasks.

2.2 SAE-Based Classification and its Limitations

Reports have demonstrated that SAE-derived features can outperform traditional hidden-state probing for classification, particularly in scenarios with noisy or limited data with closed datasets (Bricken et al., 2024) or simplified tasks (Kantamneni et al., 2024). However, more recent studies, such as Wu et al. (2025), suggest that SAEs may not be superior, particularly for model steering (instead of classification). These seemingly conflicting results highlight a critical gap in the current understanding of SAE-based classification: a lack of systematic exploration of how hyperparameters, feature aggregation strategies, and other methodological choices impact performance.

Existing evaluations often focus on narrow settings, making it unclear whether discrepancies arise from task differences, dataset choices, or specific configurations. This work addresses this gap by systematically evaluating SAE-based classification. We examine key hyperparameters and methodological choices like feature pooling, layer selection, and SAE width across diverse datasets and tasks, ensuring a fair comparison with established baselines.

3 Preliminaries

Experimental Setup Rationale: Our primary goal is to evaluate pre-trained SAE features for interpretable, zero-shot classification tasks. Accordingly, we selected the Gemma 2 SAE suite as it was the only publicly available family offering matched model backbones (2B, 9B, 9B-IT) with identical training settings and systematic layer and width pairings. We compare against two standard interpretable baselines: linear probes on hidden-state activations and TF-IDF on a bag-of-words representation. We deliberately exclude fine-tuned models, as they operate under a different, less-interpretable paradigm and fall outside our zero-shot evaluation scope. The TF-IDF baseline serves as a strong, classic non-neural benchmark for interpretability and performance.

Notation and Setup: Let M be a pretrained LLM with hidden dimension d. When M processes an input sequence of tokens of length n, it produces hidden representations $\{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\}$ for each layer, where each $\mathbf{h}_t \in \mathbb{R}^d$. We consider three versions of `Gemma 2` models (Team et al., 2024) in this work, the **2B**, **9B** and instruction-tuned variant, **9B-IT**.

SAE-Based Activation Extraction: We use **pretrained** SAEs provided by `Gemma Scope` (Lieberum et al., 2024), choosing the SAE with L_0 loss closest to 100. We extract each token's residual stream activations from layers that have been instrumented with the `SAELens` (Joseph Bloom and Chanin, 2024) tool. Specifically for the 2B model, we extract SAE features from layers 5, 12, 19 (early, middle, late) where 9B & 9B-IT models with layers 9, 20, and 31 from the residual stream.

Each SAE has a designated *width* (i.e., number of feature directions). We evaluate **16K** and **65K** widths for the 2B model, and **16K** and **131K** for 9B and 9B-IT [2], following the pretrained SAEs made available in `Gemma Scope` (Lieberum et al., 2024).

[2] we choose 131k for 9B and 65k for 2B models due to their same expansion ratio to original model hidden states

Note: we do *not* train any SAEs ourselves; our workflow involves only extracting the hidden states and the corresponding *pretrained* SAE activations.

Pooling and Binarization Since SAEs generate token-level feature activations, an essential step in classification is aggregating these activations into a fixed-size sequence representation. Without pooling, the model lacks a structured way to combine token-level representations. Previous NLP works have explored various pooling strategies for feature aggregation in neural representations (Shen et al., 2018). However, it remains unclear which pooling method is most effective for LLMs' SAE features. We systematically evaluate different pooling approaches (displayed in 2, considering (1) *Top-N feature selection per token* [3] and (2) *summation-based aggregation*[4] which collapses token-level activations into a single sequence vector:

$$\mathbf{F} = \sum_{t=1}^{n} \mathbf{f}_t, \tag{1}$$

where $\mathbf{f}_t \in \mathbb{R}^m$ is the SAE feature vector of dimension m for token t. The summation method aggregates all token activations, while top-n selects the strongest activations per token. Further details are provided in A.1.

Beyond pooling, we investigate *binarization* to enhance interpretability and efficiency. This transformation converts $\mathbf{F}$ into a binary vector $\mathbf{F}_{\text{bin}}$, activating only the dimensions that exceed a threshold:

$$\mathbf{F}_{\text{bin}}[i] = \begin{cases} 1, & \text{if } \mathbf{F}[i] > 1, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Binarization provides multiple advantages: (1) it produces compact, memory-efficient representations, (2) it acts as a non-linear activation akin to ReLU (Agarap, 2019), and (3) it serves as an implicit feature selection mechanism, highlighting only the most salient SAE activations. By thresholding weaker activations, this approach enhances the robustness and interpretability of extracted features in downstream classification tasks.

Classification with Logistic Regression: To measure how informative these SAE-derived features are for various tasks, we train a *logistic regression* (LR) classifier. In all experiments, LR models are evaluated using **5-fold cross-validation**. This is the only learned component of our pipeline;

Baselines: We compare against:

- **TF-IDF**: Classic bag-of-words variation without neural representations (Spärck Jones, 1972).
- **Hidden State**: Like prior studies (**?**), we did compare to *last-token* hidden state probing as well.

Code and Reproducibility: All code for data loading, activation extraction, pooling, detailed hyper-parameters and classification results is provided in a public repository. A simple YAML configuration file controls model scale, layer indices, SAE width, and `huggingface` dataset paths, enabling reproducible workflows with Apache 2 license. All our experiments are conducted on three Nvidia A6000 GPUs with CUDA version 12.4.

4 Classification Tasks, Multimodal Transfer, and Hyperparameter Analysis

Here, we investigate best practices for using *GemmaScope* SAE features in classification tasks across model scale, SAE width, layer depth, pooling strategies, and binarization. We also briefly touch upon the cross-modal applicability of text-trained SAE features to a *PaliGemma 2* vision-language model.

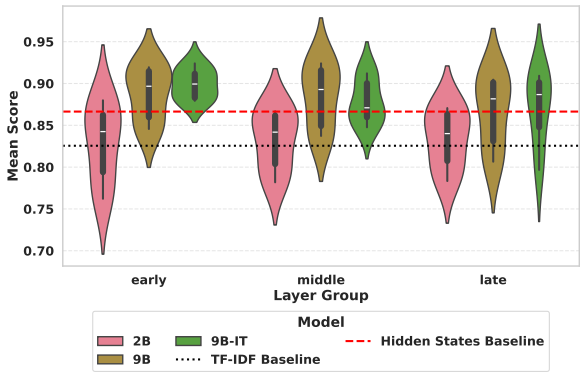
Datasets: We targeted scalable, safety-relevant *binary* classification tasks—jailbreak detection, harmful-prompt screening, and multilingual toxicity—to stress-test generality while keeping evaluation simple and comparable. Concretely, we selected publicly available datasets drawn from MTEB and other widely used classification corpora to ensure reproducibility and sufficient scale (Muennighoff et al., 2023). We prioritized (i) clear binary labels, (ii) coverage across multiple languages, and (iii) permissive licensing. At the time of experimentation, the pool of multilingual binary datasets was limited, so we focused on these three tasks; broadening the task set is an important direction for future work. Detailed dataset characteristics are in Appendix A.2.

4.1 Impact of Layer Depth and Model Scale

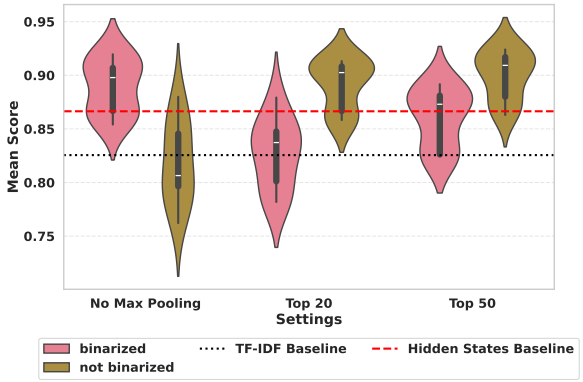
We evaluate `gemma-2-2b`, `9b`, and `9b-it`, using their early, middle, and late layers, with SAE widths of 16K/65K for `gemma-2-2b` and 16K/131K

[3] Token-level top-N where n=0 indicates the absence of max pooling. (Karvonen et al., 2025)

[4] this approach is also adopted by parallel research (Brinkmann et al., 2025).



(a) Layer-wise classification performance for each model scale. The dotted black line indicates a TF-IDF baseline, while the red dashed line indicates a last token hidden-state probe baseline. SAE-based methods (colored violin plots) often surpass these baselines, with middle-layer SAE features typically achieving the highest scores.



(b) Token level top-N vs. full binarized features. Token level top-N improves with larger values of N, and binarization can worsen this performance. However, binarization of all tokenwise activations reached the best performance of Token level top-N whilst removing the need to compute top-N values, which would be important as N scales, offering a more efficient alternative.

Figure 3: Analysis of model performance across different layers and pooling strategies. A strong baseline is established by averaging the optimal performance per task across the hidden states across three models.

for `gemma-2-9b` and `9b-it`, using different pooling strategies.

We extract token-level SAE features and train LR classifiers, comparing the results to TF-IDF and final-layer hidden-state baselines [5]. Figure 3(a) depicts the layer-wise performance for the three model scales across our text-based classification tasks. We observe:

- **Layer Influence:** Middle-layer activations typically produce slightly higher F1 scores than early- or late-layer features, indicating that mid-level representations strike a useful balance between semantic and syntactic information for classification tasks.
- **Model Scale:** Larger models (9B, 9B-IT) achieve consistently higher mean performance (above 0.85 F1) compared to the 2B model. This aligns with larger hidden dimension in these models having richer representations.
- **SAE Outperforms Baselines:** SAE based features often exceed the performance of the TF-IDF baseline (dotted black line) and final-hidden-state probe (red dashed line)

4.2 Pooling Strategies and Binarization

We next examine pooling and binarization strategies. Token level max activation pooling methods included no max pooling (top-0), top-20, and top-50 features per token. Binarization is applied after token aggregation. Figure 3(b) compares two feature selection strategies: (1) no max pooling with summation of *all* SAE features, and (2) selecting the top-N token level activations (here, 20 and 50), with and without binarization. LR classifiers are trained on the resulting features with L2 regularization.

- **Binarization:** Binarized and no max pooling of SAE features outperform both hidden-state probes and bag-of-words (dotted lines in Figure 3(b)). This indicates the effectiveness of SAE features, particularly when combined with binarization, for capturing relevant information.
- **Token level top-N Selection:** Can outperform the binarized and no max pooling approach in certain settings, especially when N increases, and not binarized. However, the margin is typically small, and top-N selection demands additional computation to identify discriminative features.

These observations motivate our decision to adopt binarized and no max pooling as a default due to theoretical reduced computational overhead whilst maintaining performance, while acknowledging that token-level top-N might excel for certain tasks.

Interpretability and Layer-Wise Insights: We find that *middle-layer* SAE features often produce the highest accuracy across tasks. This trend echoes prior work suggesting that intermediate layers encode richer, more compositional representations than either early or late layers. Crucially, we find that binarizing the full set of SAE features offers a

[5] we did not benchmark against mean-diff here because that required task to be binary classification

robust one-size-fits-all approach, whereas selecting a top-N subset can yield slightly higher performance but requires additional computational steps. From an interpretability perspective, the binarization strategy also grants a straightforward notion of "feature activation": whether or not a feature dimension was triggered above zero. Such a thresholding approach can facilitate more useful and usable feature-level analyses and potential explanations for model decisions.

4.3 Cross-Modal Transfer of Text-Trained SAE Features

Finally, we conduct a preliminary investigation into the cross-modal applicability of SAE features trained on text. Specifically, we tested whether features useful for text classification could also be beneficial in a vision-language setting.

Experimental Setup: Instead of using text-based Gemma models directly, we use a Gemma-based LLaVa model (*PaliGemma 2*) (Liu et al., 2023), which processes both image and text inputs. Activations from image-text pairs were fed into a Gemma-based SAE of equivalent size to assess whether a text-trained SAE could extract meaningful features from multimodal representations. We then classified images from CIFAR-100 (Krizhevsky and Hinton, 2009), Indian food (Rajistics, 2023), and Oxford Flowers (Nilsback and Zisserman, 2008) using SAE-derived features.

SAE Features Transfer Modalities Effectively: The results of these cross-modal experiments are detailed in Appendix A.4. We found that the binarization and no max pooling strategy, effective for text-only tasks, remained effective with SAE features derived from *PaliGemma 2* processing partial textual inputs in a vision-language environment. While these initial findings are promising, a more comprehensive study tailored for multimodal analysis is needed to fully explore the benefits and limitations of transferring text-trained SAE features to vision-language tasks.

5 Multilingual Classification and Transferability

This section evaluates the cross-lingual robustness of SAE features. We investigate whether features extracted from multilingual datasets are consistent with those found in monolingual contexts and explore the correlation between SAE feature transferability and cross-lingual prediction performance. We conduct three primary experiments: (1) comparing native and cross-lingual transfer, (2) evaluating different feature selection methods, and (3) assessing the impact of training data sampling.

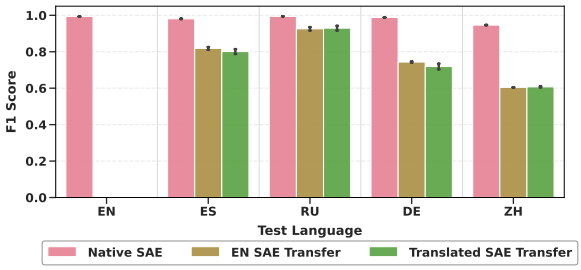


Figure 4: Multilingual toxicity detection results (middle-layer features): **Native SAE Training** (pink) consistently achieves the best F1 scores. Transferring from English (gold) or using translated inputs (green) leads to moderate performance declines. 9B-IT models show a similar trend, with slightly improved cross-lingual generalization in some language pairs.

Dataset: We use the multilingual toxicity detection dataset (Dementieva et al., 2024), which contains text in five languages labeled with a binary toxicity label: English (EN), Chinese (ZH), French (FR), Spanish (ES), and Russian (RU).

5.1 Native vs. Cross-Lingual Transfer

We first investigate the performance of SAE features when training and testing on the same language (native) versus training on one language and testing on another (cross-lingual).

Experimental Setup: Following the best configurations from previous Section, we extract SAE features from `gemma-2-9b` and `9b-it` (widths of 16K or 131K). We train linear classifiers on one language's data and test on the same or a different language. We also compare against a simpler SAE feature selection approach, the *top-n mean-difference* baseline (Mean-Diff) (Kantamneni et al., 2024), to determine if the entire feature set is necessary.

Results and Discussion: Figure 4 presents the F1 scores. Pink bars show *native SAE training*, gold bars show English-trained models tested on other languages, and green bars show English-translated models tested on translated inputs:

- **Native Training Superiority:** Native training consistently yields the highest F1 scores (e.g., EN $\rightarrow$ EN can reach over 0.99 F1).

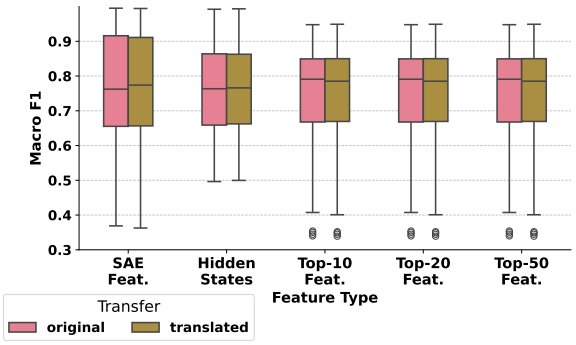


Figure 5: Comparison of average F1 scores by different feature selection methods on the Multilingual Classification and Transfer task. The boxes represent the mean ± standard deviation, and the whiskers indicate the interquartile range (IQR).

- **English Transfer Effectiveness:** Transferring SAE features trained on English (gold bars) achieves reasonable performance on ES, RU, and DE, but with a 15-20% F1 score decrease compared to native training. This indicates some cross-lingual features generalization internally inside of the models.
- **Direct Transfer Outperforms Translation:** Translating foreign language inputs into English before classification **does not** outperform direct training on the original language data. Native language signals can be effectively transferred into a shared SAE feature space, proving valuable even without explicit translation.

These results suggest that SAE-based representations have cross-lingual potential, but *native* training remains superior. Instruction tuning (`9B-IT`) yields modest gains, implying distributional shifts from instruction tuning may improve adaptability. Notably, an English-trained SAE performs well in related languages, even better than translations.

5.2 Feature Selection Methods: Full SAE vs. Hidden States vs. Mean-Diff

Experimental Setup: We compare feature selection methods on `gemma-2-9b` and `9b-it`, analyzing performance across different layers using: all SAE features (with binarization), last token hidden-state probing (baseline), and the top-N mean-difference (Mean-Diff) approach.

Results and Discussion: Figure 5 shows the average F1 scores across layers.[6] **SAE features achieve the highest macro F1 scores** but exhibit **greater variance**, particularly due to DE → ZH transfer. Despite this, they remain the **most preferable choice** due to their superior peak performance. **Hidden-state probing** performs competitively with **lower variance** but does not reach the highest scores, making it a more stable alternative. Meanwhile, **Mean-Diff top-N selection** (Top-10, Top-20, Top-50) consistently lags behind SAE features and hidden states, offering **similar variance but lower effectiveness**, reinforcing the benefit of using the full SAE feature set.[7]

However, when considering average rather than peak performance, Mean-Diff top-N selection actually outperforms SAE features, providing a higher mean F1 score and lower variance. This suggests it may be preferable in scenarios where stability across tasks is prioritized over peak performance. We then examine the robustness of SAE feature extraction with varying amounts of training data.

Experimental Setup: We assess performance across training set sampling rates (0.1–1.0), comparing native language training and English transfer. For each, we evaluate SAE binarized features, hidden states, and Mean-Diff top-N selection.

Results and Discussion: Figure 1 shows the performance across sampling rates. Key findings:

- **Native Outperforms Transfer:** Native language training consistently outperforms English transfer across **all sampling rates**.
- **SAE Features are Superior:** Our full binarized SAE features achieve superior F1 scores (0.85-0.90) compared to both hidden states (0.80-0.85) and top-N selection (0.75-0.80).
- **Stable Performance Gap:** The performance difference between native and transfer settings remains relatively stable even with limited data. This shows that our feature extraction method is robust even when data is scarce.

Clarifying differences from (Kantamneni et al., 2024) The use of L1 sparsity methods to perform feature selection, mean-difference approach

[6] Large variance of the box plot here are caused by transfer across 5 languages and 3 layer settings within 2 models.

[7] These different methods also utilize different important features to do classification which results in performance differences as shown in Appendix A.7.

of (Kantamneni et al., 2024), demonstrates strong performance and that a small number of features can contain most of the task-relevant information. However, for the specific task of our multilingual toxicity detection, the aggregated binarisation method from all features appears to preserve a stronger signal and greater transferability across languages in native and translated settings. This is in contrast to Kantamneni et al. (2025), and therefore, future work is needed to clarify the task sensitivity of the divergent findings. Major differences on feature selection methods may also drive differences and future work will focus on understanding the impact of different methods on varied interpretability approaches.

6 Behavioral (Action) Prediction

This section examines whether smaller models can predict the output correctness ("action") of larger, instruction-tuned models in knowledge-intensive QA tasks. This relates to *scalable oversight*, where a smaller, interpretable model monitors a more capable system. We focus on predicting the `9B-IT` model's behavior using features from smaller models and assess the impact of context fidelity.

Goal and Motivation: We aim to determine whether smaller and/or base models (`Gemma 2-2B, 9B`) can predict their own behavior or that of a larger and/or fine-tuned model (`9B-IT`) on knowledge-based QA tasks, based on correct or incorrect factual information. This aligns with a *scalable oversight* scenario, where a smaller model monitors a more capable system when they share the same corpus and architecture.

Datasets: We use the entity-based knowledge conflicts in question answering dataset (Longpre et al., 2022), which provides binary correctness labels for model responses. Open-ended generation is performed with *vllm* (Kwon et al., 2023), and answers are scored using *inspect ai* (AI Safety Institute) with GPT-4o-mini as the grader.

Experimental Design and Results: We focus on predicting `9B-IT`'s output correctness. For a given model M (`2B, 9B, 9B-IT`): **1)** We generate open-ended answers to prompts using the model. **2)** We use `GPT-4o-mini-0718` to label each answer as correct or incorrect. **3)** We extract pretrained SAE activations from the input question, with and without provided contexts. **4)** We train a logistic regression model to predict the binary correctness label from these extracted features.

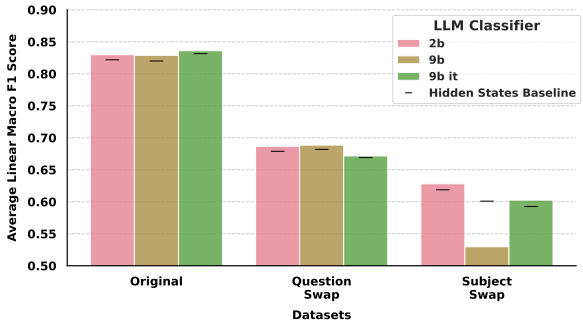


Figure 6: Action prediction performance for `9B-IT` across different context manipulations (Original, Question Swap, Subject Swap). Each bar represents a different LLM extracted features trained into classifiers (`2B, 9B, 9B-IT`) using SAE features; the black horizontal lines indicate the hidden-states baseline. High predictive power is observed with the correct context, dropping significantly with context manipulations. 2B-based features are competitive in predicting 9B-IT's behaviors.

We also perform cross-model prediction (e.g., `2B` predicting `9B`'s correctness), similar to (Binder et al., 2024). We fix the SAE width to 16K and compare the quality of predictions using full SAE binarized approach to those using the Top-N mean difference feature method, and analyze auto-interpretable descriptions of features to understand if similar explanations are shared in the top features across models. Figure 6 summarizes the macro F1 scores across several conditions from the *NQ-Swap* and *inspect_evals* datasets: Original context, Question Swap, and Subject Swap. Key findings:

- **Context Fidelity is Crucial:** Providing the correct context ("Original" setting) yields the highest F1 scores (above 80%). Removing or swapping the context causes a significant drop (20%), underscoring the importance of reliable response prediction across contexts.
- **Inter-Model Prediction is Effective:** Surprisingly, 2B-based SAE features can predict `9B-IT`'s correctness nearly as well as, and sometimes *better than*, `9B-IT`'s *own* features. This is a key result for scalable oversight.
- **SAE Features Outperform Hidden States:** Hidden-state baselines (black lines) generally perform worse than the binarized SAE feature sets, reinforcing the utility of the SAE-based approach for this "behavior prediction" task.

Implications for Scalable Oversight: These findings highlight the promise of using smaller SAEs to interpret or predict the actions of more powerful LMs. Although context consistency is critical, the ability to forecast a larger model's decisions has significant implications for AI safety and governance, especially in risk-sensitive domains. In summary, our results demonstrate that:

1. SAE-based features consistently outperform hidden-state and TF-IDF baselines across classification tasks, especially when using summation + binarization.
2. For multilingual toxicity detection, native training outperforms cross-lingual transfer, though instruction-tuned models (e.g., `9B-IT`) may exhibit modestly better transfer as you can see in Appendix A.8 and A.9.
3. Smaller LMs can leverage SAE features to accurately predict the behavior of larger instruction-tuned models, suggesting a scalable mechanism for oversight and auditing.

7 Conclusion

We present a comprehensive study of SAE features across multiple model scales, tasks, languages, and modalities, highlighting both their practical strengths and interpretive advantages. Specifically, summation-then-binarization of SAE features surpassed hidden-state probes and bag-of-words baselines in most tasks, while demonstrating cross-lingual transferability. Moreover, we showed that smaller LLMs equipped with SAE features can effectively predict the actions of larger models, pointing to a potential mechanism for *scalable oversight* and auditing. Taken together, these results reinforce the idea that learning (or adopting) a sparse, disentangled representation of internal activations can yield significant performance benefits and support interpretability objectives.

We hope this work will serve as a foundation for future studies that exploit SAEs in broader multimodal, diverse languages, and complex real-world workflows where trust and accountability are paramount. By marrying strong classification performance with clearer feature-level insights, SAE-based methods represent a promising path toward safer and more transparent LLM applications.

8 Limitations

While our study demonstrates the effectiveness of SAE features for classification and transferability, several limitations remain.

Dependence on Gemma 2 Pretrained-SAEs Our primary analysis is restricted to SAEs trained with Jump ReLU activation on Gemma 2 models as they were the only open-source models available that provided SAE's across varying layers, widths, and model sizes. This could potentially limit generalizability to other model architectures and training paradigms. Future work should explore diverse SAE training strategies and model sources.

Limited Multimodal and Cross-Lingual Evaluation Our cross-modal experiments are preliminary, and further research is needed to validate SAE generalization across different modalities and low-resource languages.

Sensitivity to Task and Data Distribution SAE performance varies across datasets, and its robustness under adversarial conditions or domain shifts needs further study.

Interpretability Challenges Despite improved feature transparency, the semantic alignment of SAE features with human-interpretable concepts remains an open question.

Potential Risks The toxicity or other safety-related open-sourced data we use contained offensive language, which we have not shown in the manuscript. And the auto-interp features are fully AI generated by neuronpedia.org.

Future Work: Robustness under Domain Shift A crucial next step is to investigate how SAE-derived features behave when the input distribution changes. This includes examining covariate, subpopulation, and temporal shifts by training probes on one domain and evaluating on held-out domains (e.g., news→social media; formal→informal), measuring activation drift and the stability of feature–label associations. This evaluation will clarify whether the observed transferability reflects domain-agnostic structure or domain-specific correlations.

Beyond robustness, there is a need to expand the task set beyond safety-oriented binary classification to include multilabel and non-safety tasks and additional multilingual benchmarks.

Acknowledgments

The authors acknowledge financial support from the Google PhD Fellowship (SC), the Woods Foun-

dation (DB, SC, JG), the NIH (NIH R01CA294033 (SC, JG, DB), NIH U54CA274516-01A1 (SC, DB) and the American Cancer Society and American Society for Radiation Oncology, ASTRO-CSDG-24-1244514-01-CTPS Grant DOI: ACS.ASTRO-CSDG-24-1244514-01-CTPS.pc.gr.222210 (DB).

The authors extend their gratitude to John Osborne from UAB for his support and to Zidi Xiong from Harvard for proofreading the preprint. Author SC also appreciates the advice on this project from Fred Zhang and Asma Ghandeharioun from Google through the mentorships program.

References

Ahmed Abdulaal, Hugo Fry, Nina Montaña-Brown, Ayodeji Ijishakin, Jack Gao, Stephanie Hyland, Daniel C. Alexander, and Daniel C. Castro. 2024. An x-ray is worth 15 features: Sparse autoencoders for interpretable radiology report generation.

Abien Fred Agarap. 2019. Deep learning using rectified linear units (relu).

UK AI Safety Institute. Inspect AI: Framework for Large Language Model Evaluations.

Anthropic. 2023. Election questions dataset.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*.

Usha Bhalla, Alex Oesterling, Suraj Srinivas, Flavio P. Calmon, and Himabindu Lakkaraju. 2024. Interpreting clip with sparse linear concept embeddings (splice).

Felix J Binder, James Chua, Tomek Korbak, Henry Sleight, John Hughes, Robert Long, Ethan Perez, Miles Turpin, and Owain Evans. 2024. Looking inward: Language models can learn about themselves by introspection.

Tolga Bolukbasi, Adam Pearce, Ann Yuan, Andy Coenen, Emily Reif, Fernanda Viégas, and Martin Wattenberg. 2021. An interpretability illusion for bert. *arXiv preprint arXiv:2104.07143*.

Samuel R. Bowman, Jeeyoon Hyun, Ethan Perez, Edwin Chen, Craig Pettit, Scott Heiner, Kamilė Lukošiūtė, Amanda Askell, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Christopher Olah, Daniela Amodei, Dario Amodei, Dawn Drain, Dustin Li, Eli Tran-Johnson, Jackson Kernion, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Liane Lovitt, Nelson Elhage, Nicholas Schiefer, Nicholas Joseph, Noemí Mercado, Nova DasSarma, Robin Larson, Sam McCandlish, Sandipan Kundu, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Timothy Telleen-Lawton, Tom Brown, Tom Henighan, Tristan Hume, Yuntao Bai, Zac Hatfield-Dodds, Ben Mann, and Jared Kaplan. 2022. Measuring progress on scalable oversight for large language models.

Trenton Bricken, Jonathan Marcus, Siddharth Mishra-Sharma, Meg Tong, Ethan Perez, Mrinank Sharma, Kelley Rivoire, Thomas Henighan, and Adam Jermyn. 2024. Using dictionary learning features as classifiers.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Chris Olah. 2023. Towards monosemanticity: Decomposing language models with dictionary learning. *transformer-circuits.pub*, monosemantic-features.

Jannik Brinkmann, Chris Wendler, Christian Bartelt, and Aaron Mueller. 2025. Large language models share representations of latent grammatical concepts across typologically diverse languages. *arXiv preprint arXiv:2501.06346*.

Nick Cammarata, Gabriel Goh, Shan Carter, Chelsea Voss, Ludwig Schubert, and Chris Olah. 2021. Curve circuits. *Distill*, 6(1):e00024–006.

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020*. Data available at https://github.com/PolyAI-LDN/task-specific-datasets.

Shan Chen, Jack Gallifant, Kuleen Sasse, and Danielle Bitterman. 2024. Are sae features from the base model still meaningful to llava? *LessWrong*.

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*.

Daryna Dementieva, Daniil Moskovskiy, Nikolay Babakov, Abinew Ali Ayele, Naquee Rizwan, Frolian Schneider, Xintog Wang, Seid Muhie Yimam, Dmitry Ustalov, Elisei Stakovskii, Alisa Smirnova, Ashraf Elnagar, Animesh Mukherjee, and Alexander Panchenko. 2024. Overview of the multilingual text detoxification task at pan 2024. In *Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum*. CEUR-WS.org.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared

Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition.

Jack FitzGerald, Christopher Hench, Charith Peris, Scott Mackie, Kay Rottmann, Ana Sanchez, Aaron Nash, Liam Urbach, Vishesh Kakarala, Richa Singh, Swetha Ranganath, Laurie Crist, Misha Britan, Wouter Leeuwis, Gokhan Tur, and Prem Natarajan. 2023. MASSIVE: A 1m-example multilingual natural language understanding dataset with 51 typologically-diverse languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024a. Scaling and evaluating sparse autoencoders. *arXiv preprint arXiv:2406.04093*.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2024b. Scaling and evaluating sparse autoencoders.

Jack Hao. 2023. Jailbreak classification dataset.

Dan Hendrycks, Mantas Mazeika, and Thomas Woodside. 2023. An overview of catastrophic ai risks. *arXiv preprint arXiv:2306.12001*.

Musashi Hinck, Matthew L. Olson, David Cobbley, Shao-Yen Tseng, and Vasudev Lal. 2024. Llava-gemma: Accelerating multimodal foundation models with a compact language model.

Curt Tigges Joseph Bloom and David Chanin. 2024. Saelens. `https://github.com/jbloomAus/SAELens`.

Subhash Kantamneni, Josh Engels, Senthooran Rajamanoharan, and Neel Nanda. 2024. Sae probing: What is it good for? absolutely something! *LessWrong*.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum McDougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. 2025. Saebench: A comprehensive benchmark for sparse autoencoders in language model interpretability.

Alex Krizhevsky and Geoffrey Hinton. 2009. Learning multiple layers of features from tiny images.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

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.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023. Improved baselines with visual instruction tuning.

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. 2022. Entity-based knowledge conflicts in question answering.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2024. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. *arXiv preprint arXiv:2403.19647*.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in Neural Information Processing Systems*, 35:17359–17372.

Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. 2023. Mteb: Massive text embedding benchmark.

Neel Nanda and Joseph Bloom. 2022. Transformerlens. `https://github.com/TransformerLensOrg/TransformerLens`.

Nelorth. 2023. Oxford flowers dataset.

Richard Ngo, Lawrence Chan, and Sören Mindermann. 2025. The alignment problem from a deep learning perspective.

Maria-Elena Nilsback and Andrew Zisserman. 2008. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024–001.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. 2024. Improving dictionary learning with gated sparse autoencoders.

Rajistics. 2023. Indian food images dataset.

SetFit. 2023. Tweeteval stance abortion dataset.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding-based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 440–450.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21.

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

Gemma Team, 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, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozińska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucińska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjoesund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, Lilly McNealus, Livio Baldini Soares, Logan Kilpatrick, Lucas Dixon, Luciano Martins, Machel Reid, Manvinder Singh, Mark Iverson, Martin Görner, Mat Velloso, Mateo Wirth, Matt Davidow, Matt Miller, Matthew Rahtz, Matthew Watson, Meg Risdal, Mehran Kazemi, Michael Moynihan, Ming Zhang, Minsuk Kahng, Minwoo Park, Mofi Rahman, Mohit Khatwani, Natalie Dao, Nenshad Bardoliwalla, Nesh Devanathan, Neta Dumai, Nilay Chauhan, Oscar Wahltinez, Pankil Botarda, Parker Barnes, Paul Barham, Paul Michel, Pengchong Jin, Petko Georgiev, Phil Culliton, Pradeep Kuppala, Ramona Comanescu, Ramona Merhej, Reena Jana, Reza Ardeshir Rokni, Rishabh Agarwal, Ryan Mullins, Samaneh Saadat, Sara Mc Carthy, Sarah Cogan, Sarah Perrin, Sébastien M. R. Arnold, Sebastian Krause, Shengyang Dai, Shruti Garg, Shruti Sheth, Sue Ronstrom, Susan Chan, Timothy Jordan, Ting Yu, Tom Eccles, Tom Hennigan, Tomas Kocisky, Tulsee Doshi, Vihan Jain, Vikas Yadav, Vilobh Meshram, Vishal Dharmadhikari, Warren Barkley, Wei Wei, Wenming Ye, Woohyun Han, Woosuk Kwon, Xiang Xu, Zhe Shen, Zhitao Gong, Zichuan Wei, Victor Cotruta, Phoebe Kirk, Anand Rao, Minh Giang, Ludovic Peran, Tris Warkentin, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, D. Sculley, Jeanine Banks, Anca Dragan, Slav Petrov, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Sebastian Borgeaud, Noah Fiedel, Armand Joulin, Kathleen Kenealy, Robert Dadashi, and Alek Andreev. 2024. Gemma 2: Improving open language models at a practical size.

A. Templeton, T. Conerly, J. Marcus, J. Lindsey, T. Bricken, B. Chen, A. Pearce, C. Citro, E. Ameisen, A. Jones, H. Cunningham, N. L. Turner, C. McDougall, M. MacDiarmid, C. D. Freeman, T. R. Sumers, E. Rees, J. Batson, A. Jermyn, S. Carter, C. Olah, and T. Henighan. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. 2022. Interpretability in the wild: a circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*.

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.

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.

A Appendix

A.1 Details on Pooling Methods

Top-N Feature Selection per Token

In our approach, the top-N feature selection per token is performed as follows:

Step 1 For each token t in a sequence, we consider its corresponding SAE activation vector:

$$f_t \in \mathbb{R}^m, \quad t = 1, 2, \ldots, n,$$

where m is the SAE dimension and n is the sequence length.

Step 2 For each token-level activation vector f_t, we keep only the top N largest activation values, setting all other activations to zero:

$$\tilde{f}_t[i] = \begin{cases} f_t[i], & \text{if } f_t[i] \text{ is among the top } N \text{ values in } f_t, \\ 0, & \text{otherwise.} \end{cases}$$

Step 3 We then aggregate these sparse vectors across all tokens by summation to obtain a fixed-size sequence-level representation F:

$$F = \sum_{t=1}^{n} \tilde{f}_t.$$

Thus, the selection is performed *per token* independently (not across the entire dataset at once). This ensures each token contributes its most salient features, and then we aggregate token-level sparse activations into a sequence-level vector.

Top-N Mean-Difference Selection

The top-N mean-difference selection method is a supervised feature selection approach performed at the dataset level, as follows:

Step 1 For each SAE dimension i, compute the absolute difference between the mean activation for the positive class C^+ and the negative class C^- over the entire training set:

$$d_i = \left| \frac{1}{|C^+|} \sum_{x \in C^+} F_x(i) - \frac{1}{|C^-|} \sum_{x \in C^-} F_x(i) \right|,$$

where $F_x(i)$ is the aggregated activation of dimension i for instance x.

Step 2 Select the top N SAE dimensions with the largest d_i values. This selection is done once at the dataset level using the training data.

Step 3 For subsequent classification, keep only these top N dimensions for all instances.

In other words, the mean-difference selection is computed using activations aggregated across all tokens and all instances in the training dataset to identify globally discriminative SAE dimensions.

A.2 Models and Dataset Information

Table 1 describes the configurations of the Gemma 2 models under study, including which layers are analyzed, the width of our SAE, and whether the model is base or instruction-tuned. These particular layers were selected based on availability of SAE widths across model sizes, and to reflect progression throughout the model.

Table 2 outlines each dataset used, specifying the type of task, a brief description, and the corresponding number of classes. These datasets focus on safety based tasks such as toxicity detection, and the multimodal datasets use the vision task such as CIFAR-100. Our goal was to test each model's robustness across both domain (language vs. vision) and complexity (binary vs. multi-class classification), thereby evaluating classifiers applicability.

Table 1: Model Configurations and SAE Specifications. We analyze select intermediate layers (see *Layers Analyzed*) to extract representations for the Stacked Autoencoder, whose width is indicated.

Model	Layers Analyzed	SAE Width	Model Type
Gemma 2 2B	5, 12, 19	2^{14}, 2^{16}	Base
Gemma 2 9B	9, 20, 31	2^{14}, 2^{17}	Base
Gemma 2 9B-IT	9, 20, 31	2^{14}, 2^{17}	Instruction-tuned

Table 2: Dataset Specifications, Task Descriptions, and Class Information. Each dataset is evaluated based on its primary task and class distribution. V) noted for vision tasks otherwise are pure text classification tasks

Dataset	Description	Classes
Multilingual Toxicity (Dementieva et al., 2024)	Cross-lingual toxicity detection	2
Election Questions (Anthropic, 2023)	Classify election-related claims	2
Reject Prompts (Arditi et al., 2024)	Detect unsafe instructions	2
Jailbreak Classification (Hao, 2023)	Detect model jailbreak attempts	2
MASSIVE Intent (FitzGerald et al., 2023)	Massive intent classification	60
MASSIVE Scenario (FitzGerald et al., 2023)	Massive scenario classification	18
Banking77 (Casanueva et al., 2020)	Banking-related queries intent classification	77
TweetEval Stance Abortion (SetFit, 2023)	Stances on abortion: favor, against, neutral	3
NQ-Swap-original (Longpre et al., 2022)	Robustness testing with correct or incorrect factual information swapped QA	2
V) CIFAR-100 (Krizhevsky and Hinton, 2009)	General image classification	100
V) Oxford Flowers (Nelorth, 2023)	Classification of 102 flower types	102
V) Indian Food Images (Rajistics, 2023)	Classification of Indian dishes	20

A.3 Performance variation on the width

We conduct an analysis of the effect of width scaling on full SAE features among our safety text classification tasks. The evaluation compares models with and without max pooling, as well as binarized and non-binarized activations, to determine their impact on classification performance. Consistently increasing the width results in decline in the mean score across all configurations, with the steepest drop observed in non-binarized cases, which is surprisingly different from Kantamneni et al. (2024) demonstrate the opposite using mean-diff feature SAE selection. A complete table of our results across variations are available at the following anonymous link `https://docs.google.com/spreadsheets/d/1zUTXBdsorzthBLwMUoXNBP-X5lrUysnNL0iLYdBZ1HU/edit?usp=sharing`.

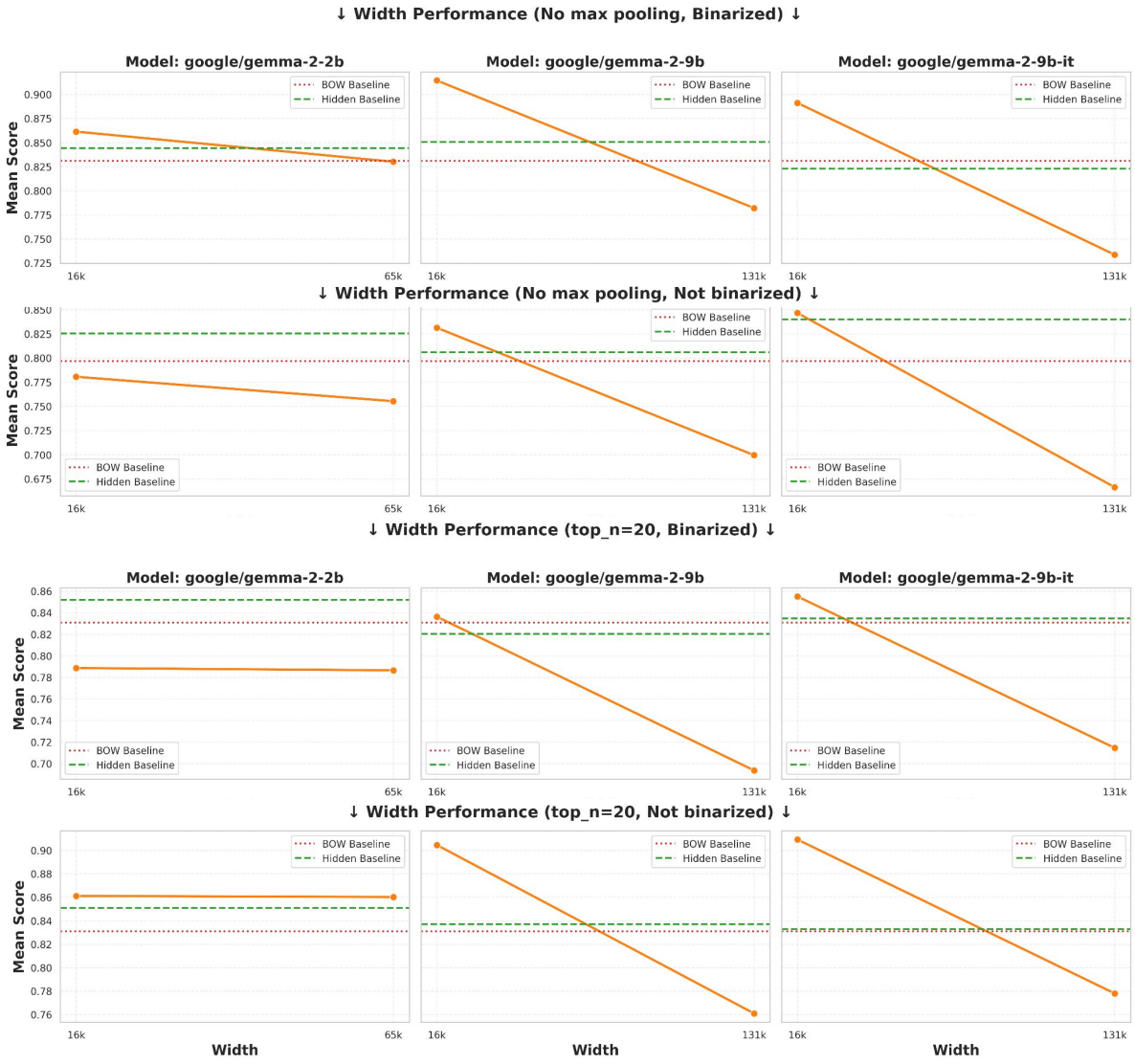


Figure 7: Performance evaluation of SAE feature transfer across different model widths for Gemma-2 models. Results are presented under different binarization and pooling settings, demonstrating a decline in mean score as width increases. The observed trends indicate that larger widths may reduce feature discriminability, particularly in non-binarized settings.

A.4 Multimodal performance

We also implemented an unsupervised approach and analyzed the retrieved features to evaluate whether meaningful features could be identified through this transfer method among other models and pretrained SAEs. Initially, features were cleaned to remove those overrepresented across instances, which could add noise or reduce interpretability.

Considering the CIFAR-100 dataset again, which comprises 100 labels with 100 instances per label, the expected maximum occurrence of any feature under uniform distribution is approximately 100. To address potential anomalies, a higher threshold of 1000 occurrences was selected as the cutoff for identifying and excluding overrepresented features. This conservative threshold ensured that dominant, potentially less informative features were removed while retaining those likely to contribute meaningfully to the analysis.

In this study, we also tried the Intel Gemma-2B LLaVA 1.5-based model (Intel/llava-gemma-2b) (Hinck et al., 2024) as the foundation for our experiments. For feature extraction, we incorporate pre-trained SAEs from jbloom/Gemma-2b-Residual-Stream-SAEs (RELU-based), trained on the Gemma-1-2B model.

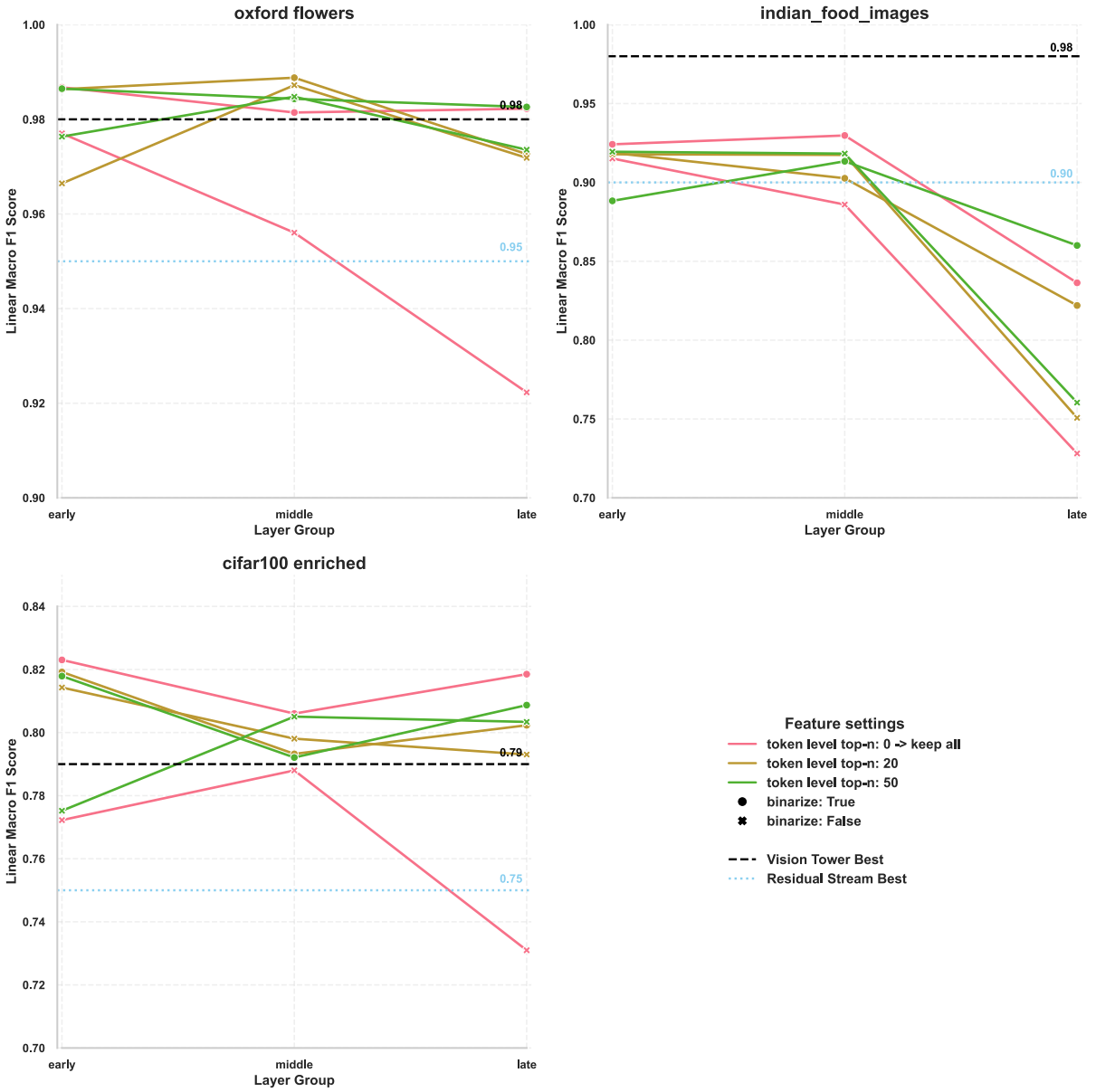


Figure 8: Performance of SAE features from gemmascope being utilised on activations derived from Peligemma 2 models. Token-n = 0 and binarization yielded overall best performance. These results also demonstrate the promise on direct SAE transfer in multimodal settings.

These SAEs include 16,384 features (an expansion factor of 8 × 2048) and are designed to capture sparse and interpretable representations.

After cleaning, we examined the retrieved features across different model layers (0–12 of 19 layers). We found that deeper layers exhibited increasingly useful/relevant features.

Below, we provide some examples of retrieved features from both high-performing and underperforming classes, demonstrating the range of interpretability outcomes.

A.5 Top retrieved features

Category	Layer	Top 2 Features (Occurrences)
Dolphin	Layer 0	Technical information related to cooking recipes and server deployment (30/100)
		Continued on next page

– Continued from previous page –

Category	Layer	Top 2 Features (Occurrences)
		References to international topics or content (26/100)
Dolphin	Layer 6	Phrases related to a specific book title: *The Blue Zones* (25/100) Mentions of water-related activities and resources in a community context (17/100)
Dolphin	Layer 10	Terms related to underwater animals and marine research (88/100) Actions involving immersion, dipping, or submerging in water (61/100)
Dolphin	Layer 12	Terms related to oceanic fauna and their habitats (77/100) References to the ocean (53/100)
Dolphin	Layer 12-it	Mentions of the ocean (60/100) Terms related to maritime activities, such as ships, sea, and naval battles (40/100)
Skyscraper	Layer 0	Information related to real estate listings and office spaces (11/100) References to sports teams and community organizations (7/100)
Skyscraper	Layer 6	Details related to magnification and inspection, especially for physical objects and images (32/100) Especially for physical objects and images (28/100)
Skyscraper	Layer 10	References to physical structures or buildings (68/100) Character names and references to narrative elements in storytelling (62/100)
Skyscraper	Layer 12	References to buildings and structures (87/100) Locations and facilities related to sports and recreation (61/100)
Skyscraper	Layer 12-it	Terms related to architecture and specific buildings (78/100) References to the sun (57/100)
Boy	Layer 0	References to sports teams and community organizations (17/100) Words related to communication and sharing of information (10/100)
Boy	Layer 6	Phrases related to interior design elements, specifically focusing on color and furnishings (52/100) Hair styling instructions and descriptions (25/100)
Boy	Layer 10	Descriptions of attire related to cultural or traditional clothing (87/100)

Continued on next page

– Continued from previous page –

Category	Layer	Top 2 Features (Occurrences)
		References to familial relationships, particularly focusing on children and parenting (83/100)
Boy	Layer 12	Words associated with clothing and apparel products (89/100) Phrases related to parental guidance and involvement (60/100)
Boy	Layer 12-it	Patterns related to monitoring and parental care (88/100) Descriptions related to political issues and personal beliefs (67/100)
Cloud	Layer 0	Possessive pronouns referring to one's own or someone else's belongings or relationships (4/100) References to sports teams and community organizations (3/100)
Cloud	Layer 6	Descriptive words related to weather conditions (24/100) Mentions of astronomical events and celestial bodies (21/100)
Cloud	Layer 10	Terms related to aerial activities and operations (62/100) References and descriptions of skin aging or skin conditions (59/100)
Cloud	Layer 12	Themes related to divine creation and celestial glory (92/100) Terms related to cloud computing and infrastructure (89/100)
Cloud	Layer 12-it	The word "cloud" in various contexts (80/100) References to the sun (47/100)

A.6 Performance Tables

Below we present the full results for evaluating our multilingual toxicity classification experiments, focusing on different feature extraction methods, top-n feature selection, and the overall experimental design.

Table 4: Multilingual Toxicity Classification Performance Comparison

Model	Transfer	SAE Features			Hidden States		
		Layer 9	20	31	Layer 9	20	31
Gemma2 - 9B	Original	0.759	**0.794**	0.766	0.772	0.792	0.765
	Translated	0.763	**0.798**	0.771	0.771	0.794	0.766
Gemma2 - 9B IT	Original	0.754	**0.784**	0.751	0.755	0.770	0.755
	Translated	0.761	**0.778**	0.753	0.761	0.776	0.747

SAE Features vs. Hidden States. Table 4 reports macro F1 scores for two Gemma2 9B model variants (base and instruction-tuned), comparing:

1. *SAE Features:* Representations learned by a Sparse Autoencoder at specific layers.
2. *Hidden States:* Direct residual stream outputs/hidden states from the same transformer layers.

We evaluate both *Original* (multilingual) and *Translated* (All translated to English) test sets. Across most settings, SAE-based features at layer 20 or 31 produce competitive (often superior) results, suggesting that deeper layers encode richer semantic information for toxicity detection. The instruction-tuned model (Gemma2 - 9B IT) also benefits from SAE features, although its absolute scores are slightly lower than the base model's best results, surprisingly, on both using full SAE features and hidden states.

Table 5: Comparison of F1 scores across different layers and top-N token selections. **top-N** indicates evaluation on the top 10, 20, or 50 **mean top-diff SAE features**. **Original** refers to the original input language, while **Translated** corresponds to translated input to English. Bold values highlight the highest scores for each row.

	Transfer	Top 10			Top 20			Top 50		
Model	**Setting**	L9	L20	L31	L9	L20	L31	L9	L20	L31
Gemma2 - 9B	Original	0.72	0.76	**0.79**	0.72	0.76	**0.79**	0.72	0.76	**0.79**
	Translated	0.72	0.76	**0.78**	0.72	0.76	**0.78**	0.72	0.76	**0.78**
Gemma2 - 9B IT	Original	0.72	0.74	**0.77**	0.72	0.74	**0.77**	0.72	0.74	**0.77**
	Translated	0.72	0.73	**0.76**	0.72	0.73	**0.76**	0.72	0.73	**0.76**

In the table above, we investigate selecting only the top 10, 20, or 50 most salient SAE features. Interestingly, reduced features can maintain or sometimes even slightly improve macro F1 performance.

A.7 Cross Lingual Transfer of Feature Activations

A more detailed set of visualizations are provided below showing how feature extraction methods perform when transferring across different languages. We first show a high-level summary of cross-lingual transfer via a heatmap (Figure 9), then we provide a series of line plots (Figures 10–13) illustrating performance versus sampling rate for five target languages. These plots compare *Native SAE Training* with *English SAE Transfer* under three feature extraction strategies: *full SAE features*, *hidden states*, and **mean difference** *top-n SAE features*.

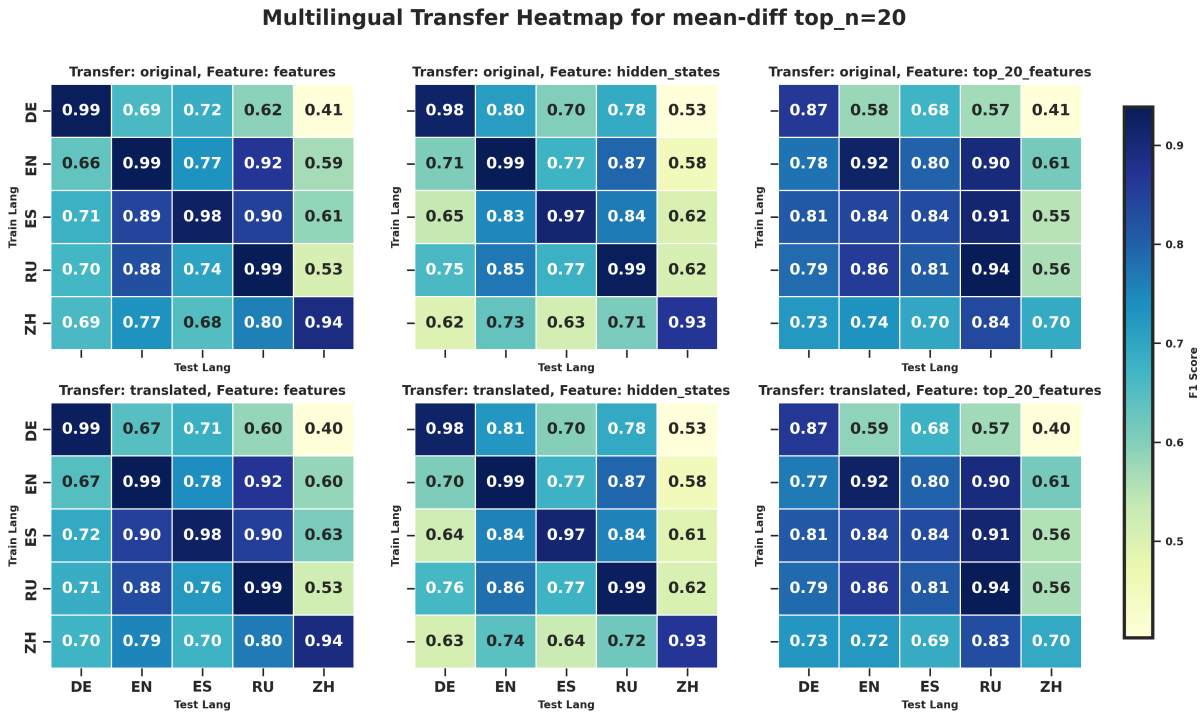


Figure 9: Average F1 scores for each training language (y-axis) versus test language (x-axis). We compare hidden states, SAE features, and top-n feature selection. The top row shows models trained on native-language datasets; the bottom row uses English-translated data for training. Darker cells indicate higher F1 performance.

Analysis of Feature Overlap. Table 6 compares F1 scores for different training approaches (*English Transfer*, *Native*, and *Translated SAE*) across five languages (DE, EN, ES, RU, ZH). The *Overlap* columns indicate how many of the top 20 SAE features are shared with each respective training scheme. As expected, each model has a complete overlap (1.000) with its own native features. In contrast, cross-lingual overlaps (e.g., *Overlap English* for Spanish or Chinese) are comparatively low (often around 0.06–0.26). Top 20 features were stored for each model trained on a language. Overlaps were calculated as standard jaccard similarities measures between train and test language sites, where we compare the features from the training set of one language to that of the top 20 features derived during training on the test language. For example, English-Spanish overlap is calculated using the top 20 SAE features derived from logistic regression training on the English dataset, and the top 20 features derived from logistic regression training on the Spanish Dataset. We then compute the similarity metric between the two.

Despite relatively small overlaps in top features, the *English Transfer* and *Translated SAE* configurations can still yield competitive F1 scores (e.g., RU with English Transfer at 0.888 or 0.903 for instruction-tuned). This suggests that, although the top features in one language are not strictly identical to those in another, a significant subset of high-impact features appears useful across languages. At the same time, the strongest performance generally occurs under *Native* training.

A.7.1 Full SAE learns classifiers find different features than Mean-diff top-N features

As we have seen in Figure 10-13, our Full SAE learns features outperform the Mean-diff Top-20 features. This makes sense because our features are learned through supervision, while the other method is done by naive clustering. You can also see that the top-20 "useful features" found by two different method from 9B-IT model is different in Figure 14. As we use more data, the overlap fully got washed out.

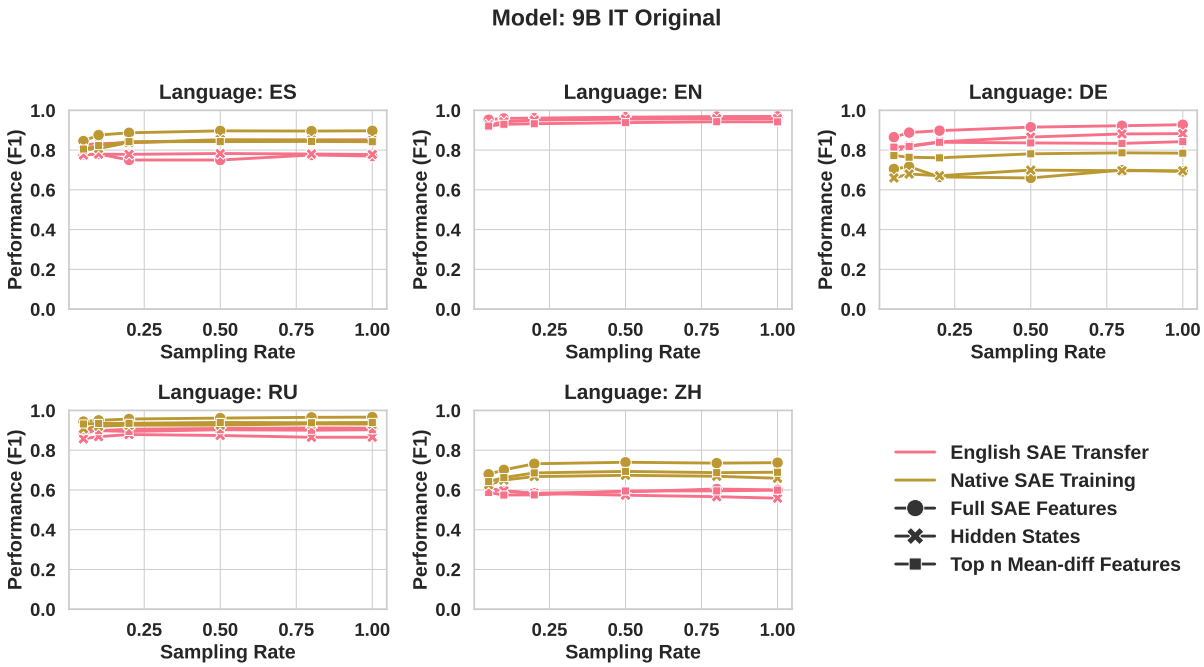


Figure 10: Performance vs. sampling rate for the 9B *instruction-tuned* model on *original-language* data. The x-axis is the sampling rate (from 0.25 to 1.0), and the y-axis is F1 score. Each subplot corresponds to a different language (ES, DE, EN, RU, ZH), while line colors distinguish *Native SAE Training* from *English SAE Transfer*. Markers reflect the feature extraction approach (*features*, *hidden_states*, or **mean difference** *top_n_features*).

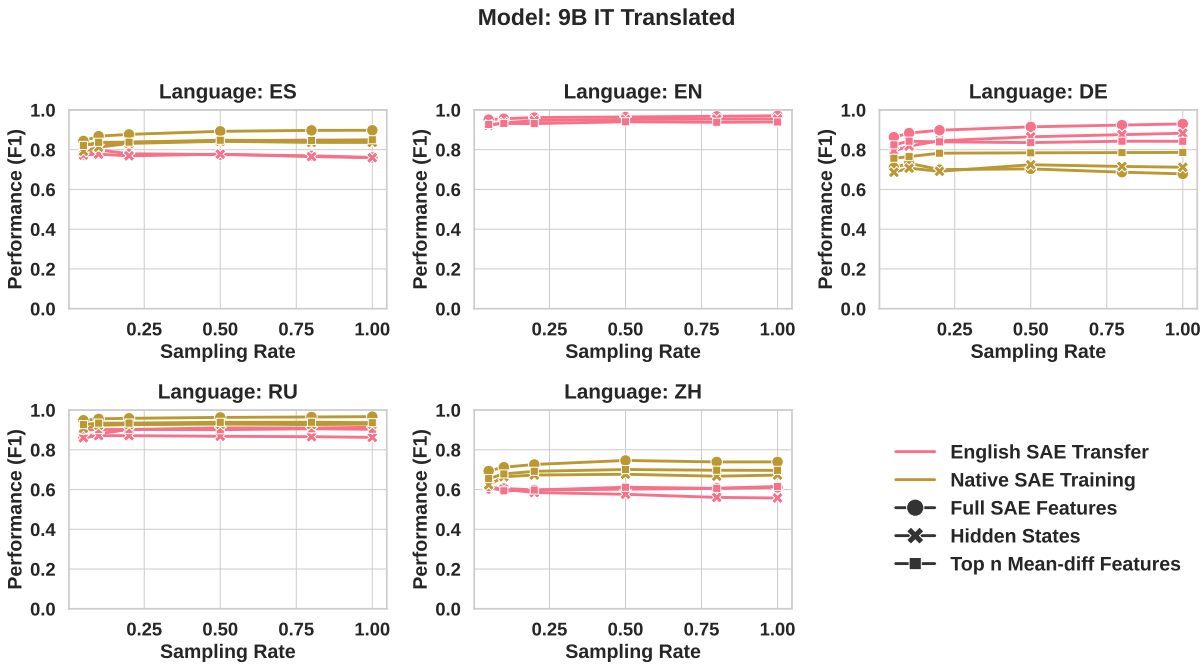


Figure 11: Performance vs. sampling rate for the 9B *instruction-tuned* model on different *translated-language* data. As in Figure 10, the x-axis shows sampling rate, the y-axis is F1, and subplots detail performance across ES, DE, EN, RU, and ZH. Lines and markers compare *English SAE Transfer* to *Native SAE Training* under different feature types.

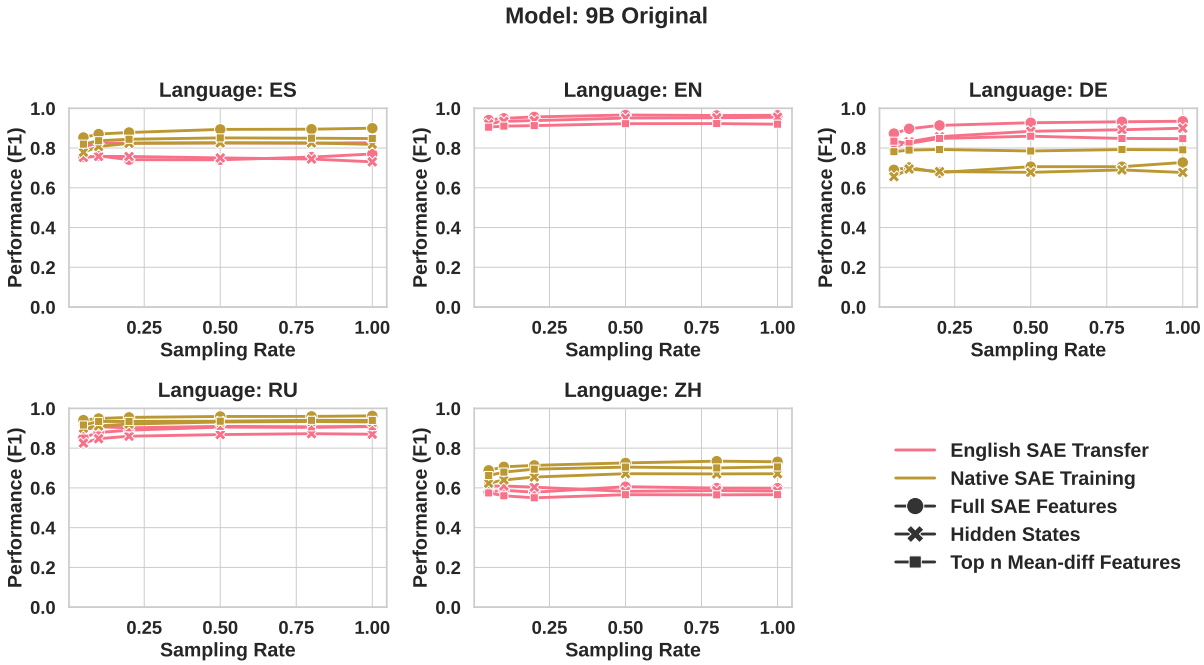


Figure 12: Performance vs. sampling rate for the 9B *base* model using *original-language* data. Subplots again separate ES, DE, EN, RU, and ZH. The curves illustrate how training type (Native vs. English transfer) and feature extraction (full features, hidden states, **mean difference** top n features) affect F1 across varying sampling rates.

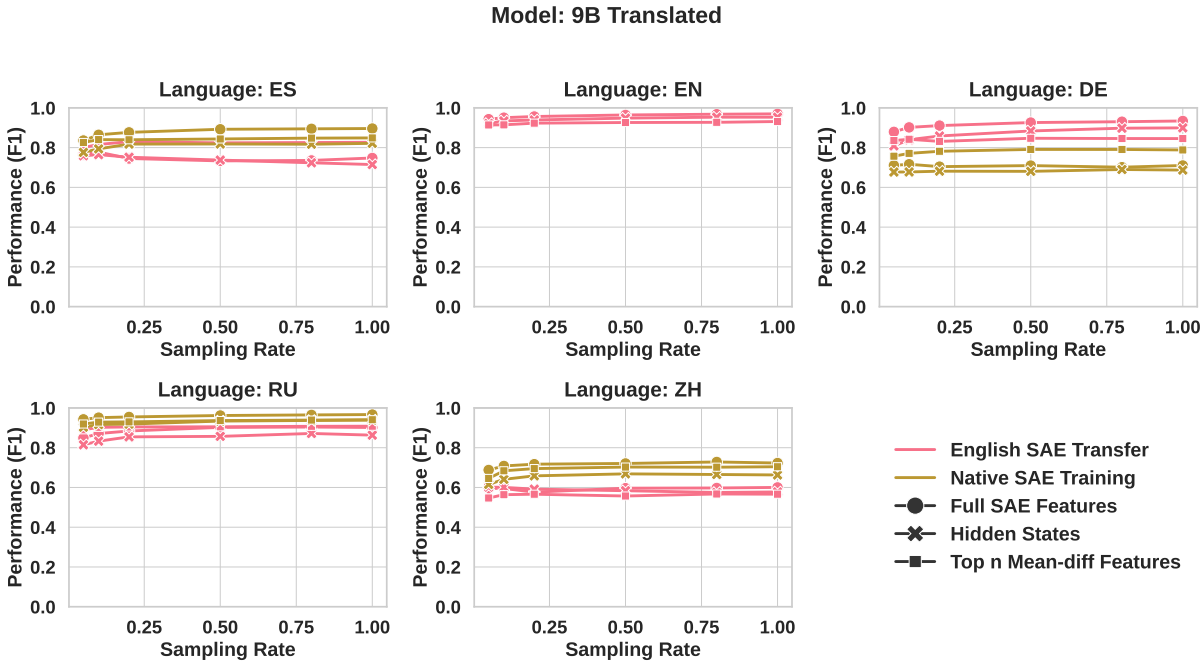


Figure 13: Performance vs. sampling rate for the 9B *base* model using *translated* datasets. The x-axis is sampling rate, the y-axis is F1, and each subplot is a distinct target language. Color and marker styles reflect training type and feature extraction, as in prior figures.

Table 6: F1 Scores and Overlap for Models and Test Languages. F1 scores are reported for three evaluation strategies: **F1 (EN-T)**: Trained on English SAE features and tested on other languages (Transfer), **F1 (N)**: Trained and tested natively, **F1 (Tr-SAE)**: Trained on translated inputs with extracted SAE features. Overlap measures indicate representation similarity: **Ovlp (EN)**: Overlap with English Transfer, **Ovlp (Tr)**: Overlap with Translated SAE.

Model	Lang	F1 (EN-T)	F1 (N)	F1 (Tr-SAE)	Ovlp (EN)	Ovlp (Tr)
9b	DE	0.710	0.945	0.708	0.098	0.099
9b	EN	–	0.969	–	–	–
9b	ES	0.768	0.926	0.771	0.212	0.200
9b	RU	0.888	0.973	0.886	0.237	0.221
9b	ZH	0.592	0.856	0.593	0.061	0.064
9b it	DE	0.722	0.941	0.723	0.093	0.089
9b it	EN	–	0.969	–	–	–
9b it	ES	0.792	0.928	0.790	0.207	0.209
9b it	RU	0.903	0.973	0.903	0.263	0.253
9b it	ZH	0.599	0.858	0.602	0.086	0.071

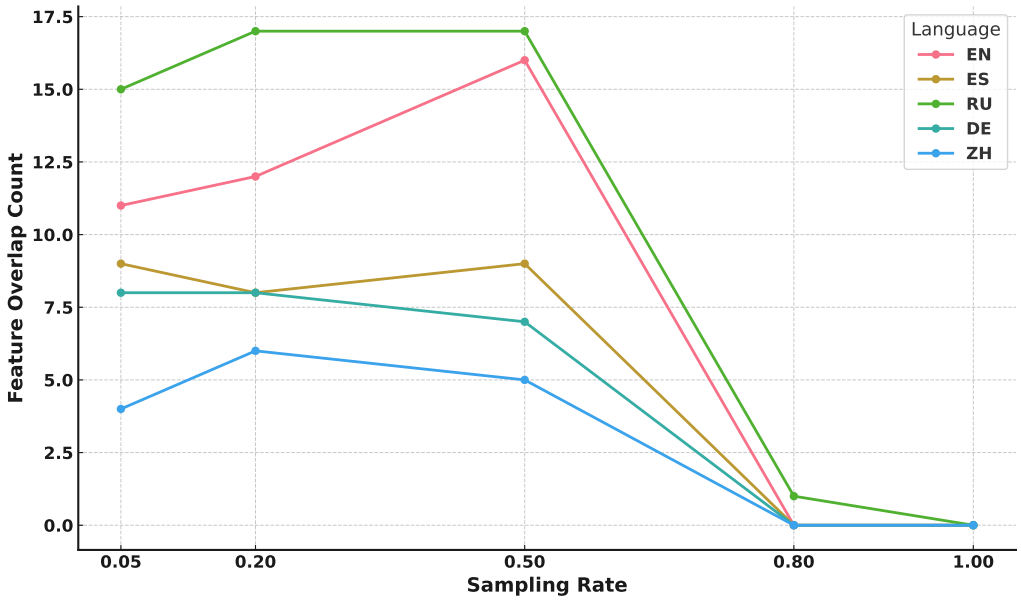


Figure 14: Feature overlap count between Full SAE Top-20 and Mean-Difference Top-20 feature selection across sampling rates among native-language trained SAEs (from 9B-IT, layer 31). Higher overlap suggests greater consistency in feature selection between the two methods.

A.8 Action prediction

below we show the disaggregated performance of SAE features vs. hidden states ability to predict a model's actions or behaviors across multiple task scenarios. Specifically, we focus on the 9B instruction-tuned model (*9b it*) under three dataset conditions:

1. **Original questions without context:** Queries posed directly with no additional background.
2. **Questions with correct context:** Queries augmented by relevant information aligned with the true scenario.
3. **Questions with incorrect context:** Queries intentionally combined with misleading or contradictory statements.

Figure 15 presents a paired bar plot that compares *hidden states* (gold bars) and *SAE features* (pink bars) for predicting whether the model will respond with a particular action or behavior. Each subplot corresponds to a different dataset, illustrating how these features perform under various context conditions. Notably, the SAE-based classifier often achieves performance levels on par with or superior to the raw hidden-state baseline, suggesting that SAE features may help isolate key aspects of the model's decision-making process. This pattern holds across original questions (no context) as well as questions provided with correct or incorrect context, indicating that SAE features can enhance interpretability and robustness in action prediction tasks.

A.9 Action Features

To further investigate how these learned representations drive action prediction, we highlight in the tables below the top classifier features for the original and no context scenario in the middle layer setting, reflecting the core layers from which features are extracted.

The goal would be to identify if similar concepts are activated across model sizes e.g. are features from the 2b similar to the concepts on the 9b-it that is trying to predict its own behaviour? These tables help reveal whether similar conceptual features emerge across different context conditions (e.g., *No Context* vs. *original*) or whether the model learns context-specific indicators tied to the question setup.

Table 7: Feature Comparison for Dataset: No Context, Layer: middle

Feature (Model google/gemma-2-2b)	Feature (Model google/gemma-2-9b)	Feature (Model google/gemma-2-9b-it)
10: terms related to programming languages	11: terms related to competition and ranking	319: phrases that denote parts of a whole
444: phrases indicating a scarcity or lack of something	3143: expressions of pride and accomplishments	1513: phrases related to raising awareness and advocacy for various social issues
632: car dealership and financing-related terminology	4152: technical terms and concepts related to data streaming and manipulation	2032: topics related to societal norms and expectations
1373: conjunctive phrases that express relationships or connections between multiple elements	4316: authenticity and sincerity in relationships and choices	7597: references to publishers and publication details
4214: phrases relating to economic inequality and socio-political commentary	4771: terms related to the emission of light and radiation in various contexts	8568: legal terminology and concepts related to administrative and tax liability
5593: terms related to switching or transitions	8741: instances of the verb "pass" and its variations in context	9520: references to applications, their requirements, and the processes involved in their submission and approval
10177: references to procedures and protocols	9153: phrases related to approaching critical points or thresholds	9912: elements and methods related to API request handling and asynchronous processing
10316: terms related to study design and data analysis methods	12185: references to sanctions and their implications	12025: references to meetings and discussions
13181: phrases that refer to taking or maintaining control or responsibility	13192: references to biblical imagery and themes related to prophecy and divine intervention	13586: common phrases or templates in written dialogues
15360: periods at the end of sentences	13510: code-related terminology and concepts in programming languages	14004: occurrences of specific events and their frequency in a legal or conversational context

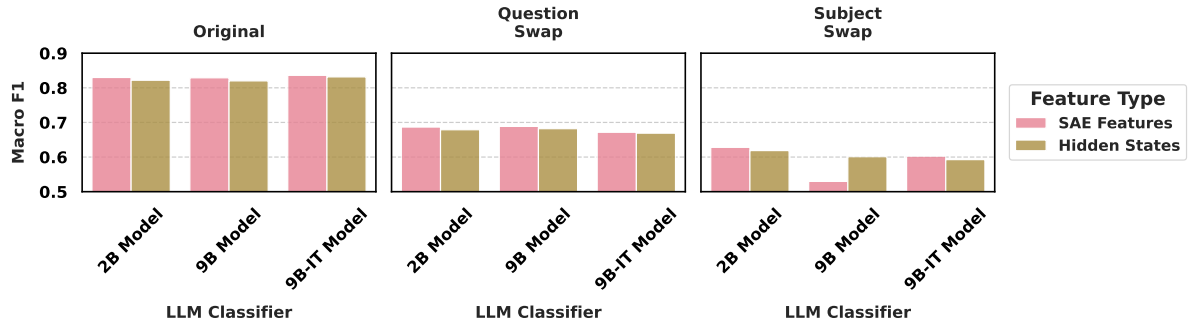


Figure 15: Paired bar plot for hidden state compared to SAE feature performance for behavior prediction across datasets.

Table 8: Feature Comparison for Dataset: Original, Layer: middle

Feature (Model google/gemma-2-2b)	Feature (Model google/gemma-2-9b)	Feature (Model google/gemma-2-9b-it)
1189: commands or instructions related to processing data or managing functions	1976: technical terms and phrases related to experimental setups and measurements	557: mentions of personal identity and name references
3563: syntax related to resource management and context management in programming (e.g., using "with" and "using" statements)	4864: cooking-related terms and ingredients	1489: instances of dialogue and conversational exchanges
4705: numerical and alphanumeric sequences, likely related to coding or technical details	5181: components of code related to database operations and responses	2297: technical programming concepts and syntax elements
5382: phrases related to customer engagement and interactions in a business context	6672: medical terminology related to women's health conditions	3084: contact information and email addresses
7360: elements related to function and method definitions	6729: mathematical symbols and notations	4110: code structure and syntax elements in programming
10140: elements related to programming structures and their definitions	7656: punctuation and formatting markers typical in academic citations	5465: phrases related to legal and ethical violations
10421: references to programming languages, libraries, and frameworks related to system and web development	7926: terms related to weights and their configurations in neural networks	6645: references to mathematical variables and parameters associated with functions and their behaviors
12396: assignment operations in code	9384: terms related to exercise and physical activity	7196: references to upcoming events or competitions
13999: array declarations and manipulations in code	9708: terms related to crime and legal issues	9384: proper nouns related to people, places, and institutions
14399: currency symbols and monetary values	13547: programming-related syntax and structure	13338: words related to programming or software-related language components

High-Level Consistencies Across Models. Across the tables comparing 2B, 9B, and 9B-IT, we see frequent mentions of programming-related features (e.g., code syntax, function definitions, data structures). Such technical elements dominate many of the top features identified by our *autointerpretable* definitions. However, we also observe several non-programming references (e.g., legal terminology, societal or economic concepts) shared across models—particularly at middle or late layers.

An example we observe is the presence of *Economic and Socio-Political Commentary* across models. The 2B model identifies phrases relating to "economic inequality and socio-political commentary" (Feature 4214), whereas 9B-IT surfaces "legal terminology and concepts related to administrative and tax liability" (Feature 8568). Both target broader sociopolitical or legal contexts.

It is important to note that our similarity claims are constrained by the level of granularity in *autointerpretable* annotations. Different feature IDs may describe related or overlapping real-world concepts, even if they are not labeled identically. At a high level, these tables suggest that all three Gemma-2 variants (2B, 9B, and 9B-IT) learn to capture similar domains, with broad thematic parallels (legal frameworks, social dynamics, etc.) emerging beyond mere code-based patterns. Thus, even though the precise feature names differ, it appears plausible that many of these salient features reflect similar underlying concepts.