How a Bilingual LM Becomes Bilingual: Tracing Internal Representations with Sparse Autoencoders

Tatsuro Inaba[1*], **Go Kamoda**[2,3], **Kentaro Inui**[1,4,5], **Masaru Isonuma**[6,4], **Yusuke Miyao**[7,6], **Yohei Oseki**[7,6], **Yu Takagi**[8†], **Benjamin Heinzerling**[5,4†]

[1]MBZUAI, [2]NINJAL, [3]SOKENDAI, [4]Tohoku University, [5]RIKEN,
[6]NII LLMC, [7]University of Tokyo, [8]Nagoya Institute of Technology

Correspondence: inaba@nii.ac.jp

Abstract

This study explores how bilingual language models develop complex internal representations. We employ sparse autoencoders to analyze internal representations of bilingual language models with a focus on the effects of training steps, layers, and model sizes. Our analysis shows that language models first learn languages separately, and then gradually form bilingual alignments, particularly in the mid layers. We also found that this bilingual tendency is stronger in larger models. Building on these findings, we demonstrate the critical role of bilingual representations in model performance by employing a novel method that integrates decomposed representations from a fully trained model into a mid-training model. Our results provide insights into how language models acquire bilingual capabilities[1].

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable multilingual capabilities (OpenAI et al., 2024; Dubey et al., 2024; Team et al., 2025). However, it is not yet clear how such capabilities emerge during pre-training. Specifically, do LLMs initially learn each language separately before aligning them? Is cross-lingual alignment distributed across layers or concentrated in specific components? How does model size affect this alignment process? These are not just theoretical questions; they directly impact our understanding of model scalability and the emergence of generalization abilities (Wei et al., 2022).

To address these questions, in this study, we explore the internal mechanisms through which LLMs develop their internal representations; namely, we trace when, where, and how bilingual alignment (English-Japanese) emerges during pre-training. For this purpose, we use sparse autoencoders (SAEs; Bricken et al., 2023; Huben et al., 2024) as a tool for our analysis, which enable us to extract interpretable latent features from hidden representations. Unlike previous approaches (Bricken et al., 2023; Huben et al., 2024; Balcells et al., 2024; Wang et al., 2025), our method captures fine-grained distinctions between language-specific and bilingual features, as well as semantic features, and allows analysis of their emergence across training stages and model layers.

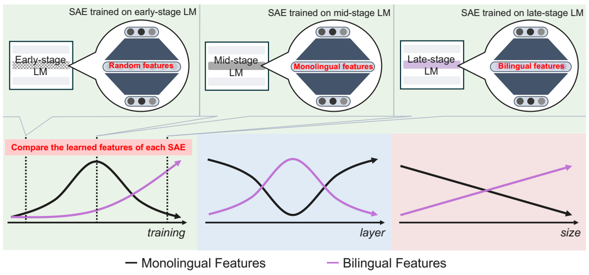


Figure 1: Illustration of the experimental setup (top) and the key findings (bottom). In the top panel, SAEs are trained independently on language models at each training stage, layer, and model size. The bottom panel visualizes the evolution of bilingual alignment, derived from comparisons of the features learned by each SAE.

We conduct experiments on decoder-only models with a variety of sizes, pretrained on an English-Japanese bilingual corpus. Our observations highlight three key findings, as illustrated in Figure 1.

- LLMs initially learn languages independently, and gradually develop bilingual alignment over training (Section 4.1).
- Bilingual alignments are more prominently captured in the mid-layers of the model (Section 4.2).
- Larger models exhibit stronger bilingual alignment than smaller ones (Section 4.3).

[*] Work completed as a research assistant at NII LLMC.

[†] Co-last authors

[1] Our code is publicly available at `https://github.com/llm-jp/llm-jp-sae`

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 13458–13470
November 4-9, 2025 ©2025 Association for Computational Linguistics

Beyond these observations, we introduce an SAE-based method to identify which types of representations are most essential to the model. We first decompose the representations of a fully trained model into three distinct types: English-specific, Japanese-specific, and bilingual. These components are then selectively injected into the model at a mid-training stage, allowing us to evaluate their importance by analyzing the resulting changes in the model's behavior.

Our results demonstrate that bilingual representations from a fully trained model enhance the performance of a mid-training model (Section 5). Beyond simply using SAEs to interpret language models, we harnessed them to directly manipulate internal representations, demonstrating their versatility as tools for both analysis and intervention. We believe that our approach can be further extended to investigate beyond bilinguality in language models, providing valuable insights to the broader research community.

2 Sparse Autoencoders

A sparse autoencoder (SAE) is an autoencoder that enforces a sparsity constraint on its hidden layer. In this study, we adopt a variant called TopK-SAE (Makhzani and Frey, 2014), where the TopK activation function is applied at the hidden layer. Compared to a ReLU-based SAE (Bricken et al., 2023; Huben et al., 2024), TopK-SAE has been shown to be easier to train while maintaining sparsity and achieving higher reconstruction performance (Gao et al., 2025).

Let $x \in \mathbb{R}^d$ be the input vector of an SAE and n be the dimension of its hidden layer. The encoder E and decoder D are defined as follows:

$$E(x) = \mathrm{TopK}\big(W_{\mathrm{enc}}(x - b_{\mathrm{pre}})\big), \quad (1)$$

$$\hat{x} = D(E(x)) = W_{\mathrm{dec}}E(x) + b_{\mathrm{pre}}, \quad (2)$$

where $W_{\mathrm{enc}} \in \mathbb{R}^{n \times d}$ and $W_{\mathrm{dec}} \in \mathbb{R}^{d \times n}$ are learned linear layers, and $b_{\mathrm{pre}} \in \mathbb{R}^d$ is a learnable bias parameter. W_{dec} is initialized as the transpose of W_{enc}, and b_{pre} is initialized to the geometric median of the input data.

The training objective is the following mean squared error (MSE) loss:

$$L = \|x - \hat{x}\|_2^2. \quad (3)$$

In this study, we control TopK-SAE by two hyperparameters: n, the dimension of the hidden layer, and K, the number of hidden dimensions to keep active. Interpreting W_{dec} as n distinct vectors in $\mathbb{R}^d$, TopK-SAE can be seen as selecting K vectors from n and using their weighted sum to reconstruct the input. In this study, we denote each dimension of the encoder output $E(x) \in \mathbb{R}^n$ as a *feature*. We say the feature is *activated* when it is selected during the TopK operation (i.e., utilized in reconstruction).

3 Experiments

In this section, we describe our experimental setup for analyzing the internal representations of bilingual language models using SAEs. We detail the language models, datasets, and SAE training procedure (Section 3.1); the procedure to find activation patterns of individual features (Section 3.2), and the evaluation of language and concept selectivity of individual features (Sections 3.3 and 3.4).

3.1 Experimental Setup

Language Models We used the models in the LLM-jp family (150M, 440M, 980M, 1.8B, 3.7B) as our focus for analysis (Aizawa et al., 2024). These models were trained on the LLM-jp Corpus v3[2], which contains 1.7T tokens: 950B in English, 592B in Japanese, 114B in code, 0.8B in Korean, and 0.3B in Chinese. We chose the LLM-jp family because (i) its intermediate checkpoints are (or available upon request) publicly available, (ii) it offers a range of model sizes, and (iii) it demonstrates bilingual capabilities in both English and Japanese. We analyzed all of the layers of each language model. For additional details about the models, please refer to the original repository[3].

Datasets We train SAEs with the Japanese and English Wikipedia subsets in the LLM-jp Corpus v3. For each document, we extract the first 64 tokens as the input to the language model, discard the [BOS] token representation, and apply L2 normalization to the remaining 63 representations ($\in \mathbb{R}^{63 \times d}$), which serve as inputs to the SAE. We use 100M tokens (50M in Japanese and 50M in English) for training, and 10M tokens (5M in Japanese and 5M in English) for evaluation.

TopK-SAE We use TopK-SAE and set the sparsity parameter $K = 32$ and the hidden layer's dimension $n = 32,768$ for all our experiments. The

[2] https://gitlab.llm-jp.nii.ac.jp/datasets/llm-jp-corpus-v3

[3] https://huggingface.co/llm-jp/llm-jp-3-3.7b

batch size is fixed at 32,768, with a warm-up phase of 500 steps. We perform a grid search to optimize the learning rate (Appendix A.1). Training a single SAE takes approximately 10 minutes to 1.5 hours on a single A100 40GB GPU. This variation is primarily due to the size of the Language Model (LM), as we simultaneously obtain intermediate activations through an LM while training an SAE. Our implementation leverages the activation buffer to temporarily store a batch of LM activations, which are then used for SAE training (Nanda, 2023; Samuel et al., 2024). The number of stored activations is adjusted according to the model size (see Appendix A.2 for details).

3.2 Finding Activation Patterns

We collect tokens that strongly activate each feature. Specifically, we first determine the maximum activation value of each feature. The threshold is then set at 70% of this maximum value, and all tokens that exceed this threshold are collected from the evaluation set.

Next, we define token attribution distribution for feature i, denoted $f(v|i)$ for $1 \leq i \leq n$, as the probability that an activation of feature i was caused by token v. This is defined by the count of v activating feature i divided by the total number of feature i being activated, satisfying $\sum_{v \in V} f(v|i) = 1$

We also assess the language distribution conditioned on the activation of each feature i. Specifically, we define $p(\text{en}|i)$ and $p(\text{ja}|i)$ as the probabilities that the input of the LM was in English or Japanese, respectively, given that the feature was activated, satisfying $p(\text{en}|i) + p(\text{ja}|i) = 1$.

3.3 Language Selectivity Metrics

We classify each feature into three categories — English Feature, Japanese Feature, and Mixed Feature — based on the calculated language probability $p(\text{en}|i)$ and $p(\text{ja}|i)$. The i-th feature is classified as an English Feature if $p(\text{en}|i) > 0.9$, a Japanese Feature if $p(\text{ja}|i) > 0.9$, and a Mixed Feature if neither condition is met. This classification reflects the dominant language context in which each feature is most strongly activated.

3.4 Concept Selectivity Metrics

To quantitatively evaluate the semantic alignment of feature-activating tokens (i.e., tokens that activate a certain feature) over languages, we use three metrics: Token Entropy, Semantic Entropy, and Monosemanticity.

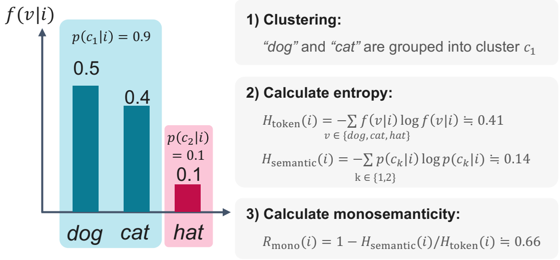

$f(v|i)$
$p(c_1|i) = 0.9$
0.5
0.4
$p(c_2|i)$
$= 0.1$
0.1
dog
cat
hat
1) Clustering:
"dog" and "cat" are grouped into cluster c_1
2) Calculate entropy:
$H_{\text{token}}(i) = -\sum_{v \in \{dog, cat, hat\}} f(v|i) \log f(v|i) \risingdotseq 0.41$
$H_{\text{semantic}}(i) = -\sum_{k \in \{1,2\}} p(c_k|i) \log p(c_k|i) \risingdotseq 0.14$
3) Calculate monosemanticity:
$R_{\text{mono}}(i) = 1 - H_{\text{semantic}}(i)/H_{\text{token}}(i) \risingdotseq 0.66$

Figure 2: The procedure for calculating Monosemanticity ($R_{\text{mono}}(i)$) from Token Entropy ($H_{\text{token}}(i)$) and Semantic Entropy ($H_{\text{semantic}}(i)$) for the i-th feature.

Token Entropy Token Entropy measures the diversity of tokens that activate a given feature. For the i-th feature, it is calculated as:

$$H_{\text{token}}(i) = -\sum_{v \in V} f(v|i) \log f(v|i) \qquad (4)$$

A high Token Entropy $H_{\text{token}}(i)$ value indicates that a wide variety of tokens can activate the feature, while a low value suggests that only a limited set of tokens do so.

Semantic Entropy Semantic Entropy quantifies the diversity of semantic meanings among the tokens that activate each feature. Calculating Semantic Entropy consists of three steps: embedding tokens, clustering based on cosine similarity, and computing the entropy of the resulting clusters.

1. **Token Embedding**: Token embeddings of feature-activating tokens, or tokens that activated feature i at least once, are extracted from the embedding layer of the 3.7B model.

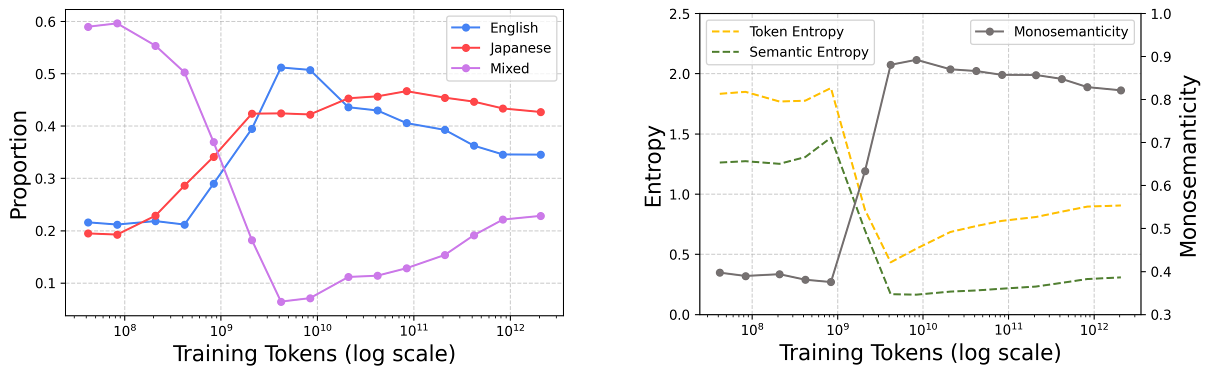
2. **Semantic Clustering**: Using the extracted embeddings, tokens with a cosine similarity above a predefined threshold are grouped into the same semantic cluster[4].

3. **Entropy Calculation**: Similar to Token Entropy, we compute the entropy over these semantic clusters using the formula:

$$H_{\text{semantic}}(i) = -\sum_{c \in C_i} p(c|i) \log p(c|i) \qquad (5)$$

where C_i is the set of semantic clusters for the i-th feature, and $p(c|i)$ is the probability that an activation of feature i was caused by a token belonging to cluster c.

[4] We set the cosine similarity threshold at 0.1 because it effectively balances capturing semantically related tokens and avoiding over-clustering of unrelated tokens.



(a) Language Distribution

(b) Semantic Distribution

Figure 3: (a) Language Distribution and (b) Semantic Distribution of SAE's features at the 14th layer of the 3.7B model across training stages. During early training ($\leq 4 \times 10^8$ tokens), the model exhibits a high proportion of mixed language features and low monosemanticity, indicating that features are activated by tokens from both languages without clear semantic coherence. As training continues (4×10^8 – 4×10^9 tokens), the mixed language proportion decreases while monosemanticity increases, reflecting more language-specific and semantically coherent features. In the late training stage ($\geq 4 \times 10^9$ tokens), the mixed-language proportion rises again, but high monosemanticity is maintained, suggesting the emergence of bilingual semantic representations.

A high value of $H_{\text{semantic}}(i)$ indicates that the activating tokens are semantically diverse, while a low value suggests they are semantically consistent. For example, in Figure 2, "dog" and "cat" are grouped into the same cluster, resulting in a relatively low semantic entropy of $H_{\text{semantic}} = 0.14$. This entropy effectively captures the degree of semantic diversity in token activation patterns.

This quantification is based on the approach proposed by Farquhar et al. (2024). While they used semantic entropy to assess the semantic diversity among sentences and leveraged LLMs to cluster these sentences, our method applies semantic entropy to measure semantic diversity among tokens.

Monosemanticity Monosemanticity provides a normalized measure that quantifies the relationship between the semantic diversity and token diversity. It is defined as the complement of the ratio of semantic entropy to token entropy:

$$R_{\text{mono}}(i) = 1 - \frac{H_{\text{semantic}}(i)}{H_{\text{token}}(i)} \tag{6}$$

This ratio ranges between 0 and 1: A value close to 1 suggests that although the feature is activated by a wide variety of tokens (high Token Entropy), these tokens are semantically similar (low Semantic Entropy). A value close to 0 indicates that the activating tokens are both diverse in form and meaning (high token entropy and high semantic entropy) or they are both consistent in form and meaning (low token entropy and low semantic entropy). In the special case where $H_{\text{token}}(i) = 0$ (i.e., only one token activates the feature), we define $R_{\text{mono}}(i) = 1$.

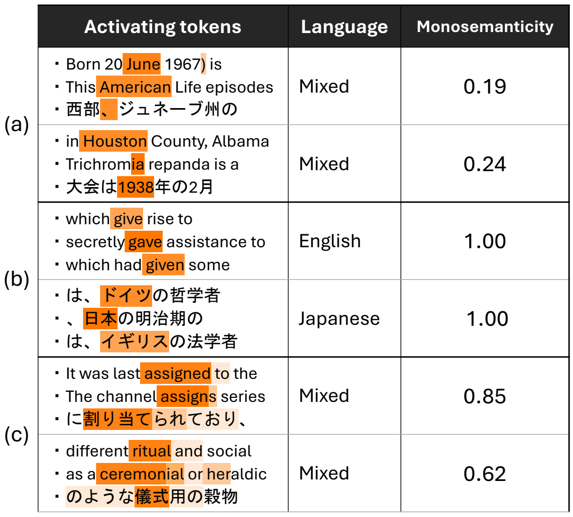
4 Observations

We first examine the internal representations of the model by analyzing the distribution of each SAE's features trained on various checkpoints (Section 4.1), layers (Section 4.2), and model sizes (Section 4.3).

4.1 LLMs first learn languages independently before aligning them bilingually

Figure 3 presents the evolution of language and semantic distributions for SAE's features at the 14th layer of the 3.7B model across different training stages. In the early training phase ($\leq 4 \times 10^8$ tokens), most features are categorized as mixed features and exhibit low monosemanticity. This indicates that individual features are activated by tokens from both Japanese and English without any consistent semantic pattern, effectively behaving as random activation patterns. This observation is consistent with the activation patterns shown in Figure 4(a), where activated tokens lack any clear semantic or linguistic coherence.

As training progresses into the middle phase (4×10^8 – 4×10^9 tokens), the proportion of mixed language features sharply declines, while monosemanticity markedly increases. This shift suggests that features become more language-specific, activating on tokens within a single language that share



	Activating tokens	Language	Monosemanticity
(a)	・Born 20 June 1967) is ・This American Life episodes ・西部、ジュネーブ州の	Mixed	0.19
	・in Houston County, Albama ・Trichromia repanda is a ・大会は1938年の2月	Mixed	0.24
(b)	・which give rise to ・secretly gave assistance to ・which had given some	English	1.00
	・は、ドイツの哲学者 ・、日本の明治期の ・は、イギリスの法学者	Japanese	1.00
(c)	・It was last assigned to the ・The channel assigns series ・に割り当てられており、	Mixed	0.85
	・different ritual and social ・as a ceremonial or heraldic ・のような儀式用の穀物	Mixed	0.62

Figure 4: Activation patterns of features at the 14th layer of the 3.7B model across training stages. (a) In the early training stage (4×10^6 tokens), features are activated by random tokens without any clear semantic structure. (b) In the mid-training stage (4×10^9 tokens), features become more language-specific, with tokens activating on semantically similar words in a single language. (c) In the fully trained model (2×10^{12} tokens), features exhibit bilingual activation, with semantically related tokens appearing in both Japanese and English.

coherent semantic meanings. For instance, Figure 4(b) illustrates two representative examples: the first feature is activated by English tokens "give," "gave," and "given," which are grammatical variations of the same verb, while the second feature is activated by Japanese tokens representing country names ("ドイツ" for Germany, "日本" for Japan, and "イギリス" for the United Kingdom). These patterns demonstrate that the model is beginning to organize and align semantics within each language independently.

In the late training stage ($\geq 4 \times 10^9$ tokens), the model exhibits a resurgence of mixed-language features while maintaining high monosemanticity. This phase signifies a transition from language-specific semantics to bilingual semantic alignment, where features activate on semantically similar tokens across both languages. As shown in Figure 4(c), one feature is activated by "assigned," "assign," and "割り当て" (the Japanese term for "assign"), while another is activated by "ritual," "ceremon," and "儀式" (the Japanese term for "ritual"). These examples confirm that the model now captures semantic correspondences between languages, functioning as a bilingual representation.

These findings suggest that LLMs learn in two distinct stages.

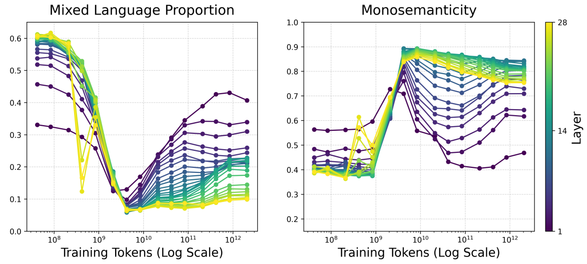


Figure 5: Layer-wise evolution of mixed language proportion and the monosemanticity in 3.7B model across training stages.

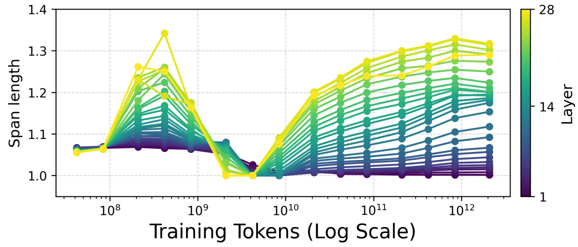


Figure 6: Layer-wise evolution of span length average in 3.7B model across training stages.

1. During the early to mid-training phase, they develop independent semantic representations within each language.

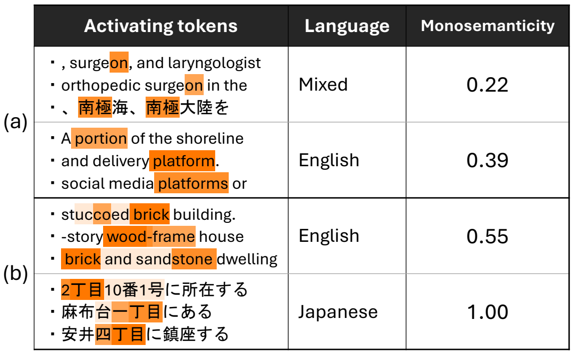
2. In the subsequent mid-to-late training phase, they begin to align these semantic representations across languages

4.2 Mid-layers capture more bilingual alignments

Figure 5 illustrates the layer-wise evolution of the mixed language proportion and the monosemanticity of SAEs' features in the 3.7B model across training stages. In the early to mid-training phase ($\leq 4 \times 10^9$ tokens), all layers exhibit a decrease in the mixed language proportion and an increase in monosemanticity. This suggests that the model initially learns the semantics within each language in all layers.

As training progresses into the later stages, layer behaviors begin to diverge. The mid layers (green) align with the behavior of the 14th layer described in Section 4.1, while the lower (purple) and upper layers (yellow) follow distinct patterns.

In the lower layers, particularly the initial layers, the mixed language proportion increases, while monosemanticity decreases compared to the mid



	Activating tokens	Language	Monosemanticity
(a)	• , surgeon, and laryngologist • orthopedic surgeon in the • 、南極海、南極大陸を	Mixed	0.22
	• A portion of the shoreline • and delivery platform. • social media platforms or	English	0.39
(b)	• stuccoed brick building. • -story wood-frame house • brick and sandstone dwelling	English	0.55
	• 2丁目10番1号に所在する • 麻布台一丁目にある • 安井四丁目に鎮座する	Japanese	1.00

Figure 7: Activation patterns of features in the 3.7B model. (a) In the lower layer (2nd layer), features exhibit activation across multiple meanings. (b) In the upper layer (26th layer), features primarily activate on long-span tokens.

layers. This suggests a tendency toward polysemanticity, where a single feature is activated by multiple meanings. As illustrated in Figure 7(a), the first feature is activated on both the English word "on" and "南極" (Japanese for "Antarctica"), and the second feature is activated on " portion", " platform", and " platforms". Although these activation patterns are less random than in the early training stages, they still occur across multiple tokens, reflecting the model's polysemantic nature in these layers. Such behavior can be attributed to the model's proximity to the input layer, where it must distinguish between a vast vocabulary of approximately 100,000 tokens, which exceeds the dimension n of the intermediate layers of the SAE.

On the other hand, the upper layers consistently maintain a lower mixed language proportion than the mid layers, while their monosemanticity declines even further as training progresses. Analyzing span length — the number of consecutive tokens each feature activates — reveals that these deeper layers increasingly focus on longer spans (Figure 6), indicating that features are not monosemantic at the token level because they span multiple, contextually connected tokens. For instance, as shown in Figure 7(b), the first feature is activated on phrases such as "uccoed brick", "wood-frame", and "brick and sandstone", all referring to building materials with spans of around three tokens. The second feature activates on Japanese addresses such as "2丁目10番1号" (similar to "Block 2, No. 10-1"), "一丁目" ("Block 1"), "四丁目" ("Block 4"), each spanning multiple tokens.

From these findings, it can be inferred that

- Mid layers specialize in learning bilingual representations, balancing monosemanticity and mixed language proportion.

- Lower layers exhibit polysemanticity, distinguishing a wide variety of tokens in the vocabulary.

- Upper layers focus on multi-token concepts by capturing longer spans rather than individual tokens.

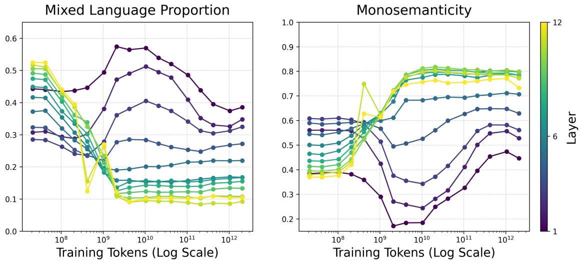


Figure 8: Layer-wise evolution of the mixed language proportion and the monosemanticity in the 150M model across training stages.

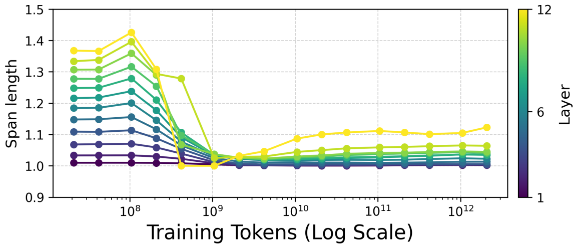


Figure 9: Layer-wise evolution of span length average in 150M model across training stages.

4.3 Larger LMs develop more bilingual alignments

Figure 8 illustrates the layer-wise evolution of the mixed language proportion and the monosemanticity in the 150M model. Figures 12 to 14 also show the result of other sizes (440M, 980M, and 1.8B). In the early to mid-training phase ($\leq 4 \times 10^9$ tokens), the behavior of around mid layers mirrors that of the 3.7B model: the mixed language proportion decreases while monosemanticity increases. This indicates that even in smaller models, the early training stage primarily involves learning languages individually.

However, a divergence becomes apparent in three aspects: (1) within mid layers during the late training phase ($\geq 4 \times 10^9$ tokens), (2) within upper layers during the late training phase, and (3) within the lower layers during all training phases.

In the mid layers, the smaller model shows a smaller increase in mixed language proportion com-

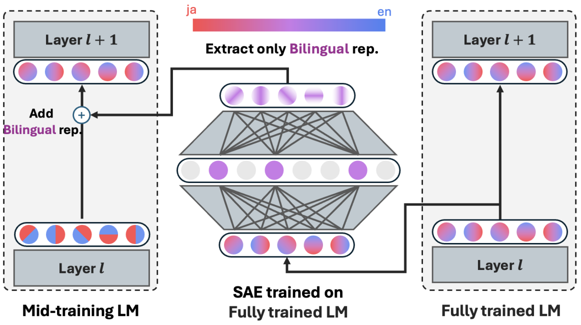


Figure 10: Illustration of adding bilingual representations from a fully trained model into a mid-training model.

pared to the larger model, as described in Section 4.2. The features learned by the smaller model around the mid layers are less inclined to exhibit high monosemanticity across languages. This suggests that a much lower capacity for learning bilingual features compared to the larger model.

In the upper layers, monosemanticity is smaller compared to larger models at the late training stages. As shown in Figure 9, the layer-wise change in span length in the 150M model indicates that the increase in span length in the upper layers is also smaller than in larger models. These observations suggest that the upper layers in smaller models cannot capture the context-level meanings.

In lower layers, the smaller model retains a relatively high mixed language proportion and low monosemanticity. This indicates a failure to adequately capture semantics even within individual languages, unlike the larger model, where lower layers effectively acquire intra-language semantics.

In summary, two key observations can be drawn:

- Larger models exhibit a greater ability to learn bilingual features in the mid layers, while smaller models struggle to do so.
- Although smaller models may acquire some degree of semantic alignments within individual languages in certain layers, they lack a strong tendency to generalize these features towards bilingual representations in the later stages of training.

5 Intervention

We hypothesize that bilingual representations, which correspond to bilingual features, play a crucial role in the performance of a fully trained model. If this is true, integrating these representations into a mid-training model should significantly enhance its performance. To test this, here, we extract bilingual representations from a fully trained model using a TopK-SAE and inject them into the intermediate representations of a mid-training model. This process is illustrated in Figure 10.

5.1 Method

Mathematically, let $\boldsymbol{X}^{\ell}_{\text{full}}, \boldsymbol{X}^{\ell}_{\text{mid}} \in \mathbb{R}^{T\times d}$ denote the outputs of the ℓ-th layer of the fully trained and mid-training models, respectively, where T is the sequence length and d is the model dimension. We also denote $E : \mathbb{R}^d \to \mathbb{R}^n$ and $D : \mathbb{R}^n \to \mathbb{R}^d$ as an encoder and a decoder of TopK-SAE trained on the fully trained model. A binary mask $\textbf{mask} \in \mathbb{R}^n$ is also defined, with m elements set to 1 and others to 0, forcing only the bilingual features to get activated. The intervention is formulated as follows:

$$\boldsymbol{X}^{\ell}_{\text{mid}} \leftarrow \boldsymbol{X}^{\ell}_{\text{mid}} + \alpha \cdot D(\textbf{mask} \odot E(\boldsymbol{X}^{\ell}_{\text{full}})) \quad (7)$$

where α is a hyperparameter controlling the strength of the intervention, set to 0.1 in our experiments (see Appendix B for the result of other values). This method allows us to assess the direct impact of the bilingual representation incorporation.

5.2 Setup

We conducted experiments using the 14th layer of the 3.7B model. As the mid-training model, we selected the checkpoint at 10,000 (approximately 40B training tokens), where the mixed language proportion in this layer is relatively low (Figure 3). We evaluated the effects of three feature types: English, Japanese, and Bilingual (Mixed). The number of selected feature dimensions was set to $m = 5,000$ (see Appendix B for the result of other values). Each setting was evaluated five times, and the results were averaged.

5.3 Results & Discussion

Table 1 shows the results. Adding English-specific representations mainly improved English performance, while adding Japanese-specific representations primarily enhanced Japanese performance. In contrast, adding bilingual representations significantly improved both languages' performances. This performance boost holds even when varying the hyperparameters α and m as shown in Table 4. These results support our hypothesis that the bilingual alignments acquired by the model in the later

	Perplexity (↓)			COMET-22 (↑)	
Add	En	Ja	all	En → Ja	Ja → En
-	18.70	25.78	22.43	61.1	56.4
En	18.53	25.64	22.28	61.4	56.9
Ja	18.65	25.33	22.17	61.3	**57.2**
Bi	**18.36**	**25.20**	**21.96**	**62.5**	**57.2**

Table 1: Baseline denotes the perplexity (PPL) of the mid-training model without any intervention. Adding mixed (bilingual) representations leads to a greater reduction in PPL compared to adding Japanese or English representations.

training stages play a crucial role in its performance.

Note that this method requires the output of Layer ℓ from the fully trained model, meaning that the SAE alone cannot directly enhance the performance of a mid-training model. However, our findings reveal that the bilingual information encoded in the later training stages is more critical for performance than monolingual information. This suggests that designing a training schedule that encourages the acquisition of bilingual knowledge in the later stages of pre-training could be beneficial.

6 Related Work

Understanding the internal mechanisms of LLMs has become a major focus of the research community. Recent studies show neural networks can represent more features than their dimensions (Elhage et al., 2022). To disentangle these representations, SAEs have emerged as a key tool for decomposing them into interpretable components (Huben et al., 2024; Olshausen and Field, 1997). While early work primarily focused on a single SAE, recent studies have shifted toward comparing SAE features across layers (Balcells et al., 2024; Balagansky et al., 2025), model architectures (Lan et al., 2024; Lindsey et al., 2024), or fine-tuning stages (Lindsey et al., 2024; Wang et al., 2025). Xu et al. (2024) concurrently tracks feature formation during training, but lacks quantitative evaluation.

Another line of research has explored the multilingual capability of language models. Zeng et al. (2025) explored the formation of multilingual capabilities through neuron-level analysis and showed that as models become larger and training progresses, they exhibit an increasing degree of multilingual understanding. This result aligns with our SAE-based analysis results. Wang et al. (2024) identified neurons shared across languages and tasks, while Tang et al. (2024) and Kojima et al. (2024) highlighted language-specific neurons, demonstrating their impact on model performance and language output.

Our research builds on these foundations and contributes to them in three key ways: (1) we investigate the formation process of bilingual capabilities within a bilingual language model, (2) we conduct a comparative analysis across training stages, model sizes, and layers, and (3) we exmploy SAEs to perform direct interventions on bilingual representations, offering novel insights on the dynamics of bilingual representation in language models.

7 Conclusion

In this study, we investigated the evolution of internal representations in language models using SAEs. Our analysis revealed that bilingual language models initially learn languages independently and later develop bilingual alignments, particularly in the mid-layers of larger models. We further demonstrated the importance of bilingual representations by conducting targeted interventions with SAEs. Beyond using SAEs solely for interpreting language models, we leveraged them to manipulate internal representations, showcasing their potential as a tool for both analysis and intervention. We believe that our approach can be extended to explore beyond analyzing the bilinguality of language models and offer valuable insights for the broader research community.

8 Limitations

This study explored the internal mechanisms of bilingual language models, specifically focusing on English, Japanese, and their bilingual interactions. While this provides insights into cross-lingual representation between these two typologically distinct languages, the findings may not generalize to all language pairs. Future research should investigate a wider range of language pairs to validate and extend our observations.

Another limitation is the interpretability of the SAEs used in our analysis. While SAEs allowed us to investigate the types of information that models tend to encode as features, recent studies have raised concerns about the reliability and interpretability of them. Additionally, given that the reconstruction accuracy was not perfect, our analysis is based on an approximation of the model's internal representations. As a direction for future work, combining SAEs with other analytical methods could lead to a more robust and comprehensive understanding of the model's behavior.

Acknowledgements

This work was supported by JSPS KAKENHI (Grant Number JP25KJ06300) and JST BOOST, Japan (Grant Number JPMJBS2412). We used the "mdx: a platform for building a data-empowered society" (Suzumura et al., 2022). ChatGPT was employed to assist with manuscript preparation and code generation, and all outputs were thoroughly reviewed and validated by the authors.

References

Akiko Aizawa, Eiji Aramaki, Bowen Chen, Fei Cheng, Hiroyuki Deguchi, Rintaro Enomoto, Kazuki Fujii, Kensuke Fukumoto, Takuya Fukushima, Namgi Han, Yuto Harada, Chikara Hashimoto, Tatsuya Hiraoka, Shohei Hisada, Sosuke Hosokawa, Lu Jie, Keisuke Kamata, Teruhito Kanazawa, Hiroki Kanezashi, and 62 others. 2024. Llm-jp: A cross-organizational project for the research and development of fully open japanese llms. *arXiv preprint arXiv:2407.03963*.

Nikita Balagansky, Ian Maksimov, and Daniil Gavrilov. 2025. Mechanistic permutability: Match features across layers. In *The Thirteenth International Conference on Learning Representations*.

Daniel Balcells, Benjamin Lerner, Michael Oesterle, Ediz Ucar, and Stefan Heimersheim. 2024. Evolution of sae features across layers in llms. *arXiv preprint arXiv:2410.08869*.

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*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, and others. 2024. The Llama 3 herd of models. *arXiv [cs.AI]*.

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. *Transformer Circuits Thread*.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. 2025. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*.

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*.

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. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6919–6971, Mexico City, Mexico. Association for Computational Linguistics.

Michael Lan, Philip Torr, Austin Meek, Ashkan Khakzar, David Krueger, and Fazl Barez. 2024. Sparse autoencoders reveal universal feature spaces across large language models. *arXiv preprint arXiv:2410.06981*.

Jack Lindsey, Adly Templeton, Jonathan Marcus, Thomas Conerly, Joshua Batson, and Christopher Olah. 2024. Sparse crosscoders for cross-layer features and model diffing. *Transformer Circuits Thread*.

Alireza Makhzani and Brendan Frey. 2014. k-sparse autoencoders. *arXiv preprint arXiv:1312.5663*.

Neel Nanda. 2023. Open source replication & commentary on anthropic' s dictionary learning paper. *AI Alignment Forum*.

Bruno A. Olshausen and David J. Field. 1997. Sparse coding with an overcomplete basis set: a strategy employed by v1? *Vision Research*, 37(23):3311–3325.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Marks Samuel, Karvonen Adam, and Mueller Aaron. 2024. dictionary_learning. `https://github.com/saprmarks/dictionary_learning`.

Toyotaro Suzumura, Akiyoshi Sugiki, Hiroyuki Takizawa, Akira Imakura, Hiroshi Nakamura, Kenjiro Taura, Tomohiro Kudoh, Toshihiro Hanawa, Yuji Sekiya, Hiroki Kobayashi, Yohei Kuga, Ryo Nakamura, Renhe Jiang, Junya Kawase, Masatoshi Hanai, Hiroshi Miyazaki, Tsutomu Ishizaki, Daisuke Shimotoku, Daisuke Miyamoto, and 13 others. 2022. mdx: A cloud platform for supporting data science and cross-disciplinary research collaborations. In *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pages 1–7.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Xin Zhao, Furu Wei, and Ji-Rong Wen. 2024. Language-specific neurons: The key to multilingual capabilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5701–5715, Bangkok, Thailand. Association for Computational Linguistics.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Junxuan Wang, Xuyang Ge, Wentao Shu, Qiong Tang, Yunhua Zhou, Zhengfu He, and Xipeng Qiu. 2025. Towards universality: Studying mechanistic similarity across language model architectures. In *The Thirteenth International Conference on Learning Representations*.

Weixuan Wang, Barry Haddow, Minghao Wu, Wei Peng, and Alexandra Birch. 2024. Sharing matters: Analysing neurons across languages and tasks in llms. *Preprint*, arXiv:2406.09265.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*.

Yang Xu, Yi Wang, and Hao Wang. 2024. Tracking the feature dynamics in llm training: A mechanistic study. *arXiv preprint arXiv:2412.17626*.

Hongchuan Zeng, Senyu Han, Lu Chen, and Kai Yu. 2025. Converging to a lingua franca: Evolution of linguistic regions and semantics alignment in multilingual large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10602–10617, Abu Dhabi, UAE. Association for Computational Linguistics.

A Training Details

A.1 Learning Rate Selection

We determined the optimal learning rate for training SAEs on each LM size by a grid search. Specifically, we tested several learning rates (1e-4, 2e-4, 5e-4, 1e-3, 2e-3, 5e-3) for each LM, the last checkpoint, and the middle layer (maxlayer // 2), and selected one that resulted in the lowest reconstruction loss (Eq. 3) on the validation set.

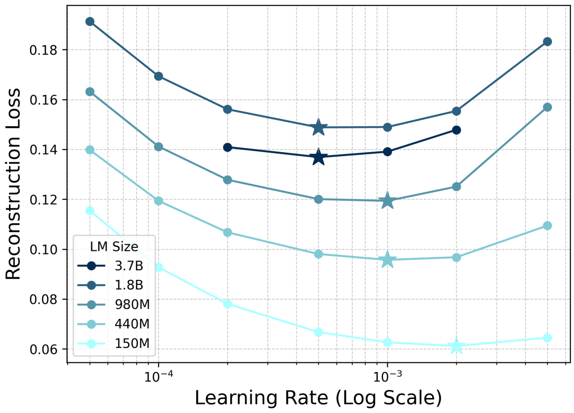


Figure 11: Learning Rate vs. Reconstruction Loss for SAEs on Various Model Sizes. The star markers indicate the lowest loss points for each model.

Figure 11 shows the result. Our experiments revealed that smaller learning rates were more effective for training SAEs on larger LMs. The selected learning rates for each model size are summarized in Table 2.

LM Size	Optimal Learning Rate
150M	2e-3
440M	1e-3
980M	1e-3
1.8B	5e-4
3.7B	5e-4

Table 2: Optimal learning rates for training SAEs across different LM sizes

A.2 Time for training SAEs & the number of stored activations

Table 3 shows the details.

B Ablation Study of Adding Bilingual Features

Table 4 shows the ablation result of different *alpha* and m.

LM Size	Training Time	N of act.
150M	20min	10M
440M	25min	5M
980M	40min	2M
1.8B	60min	1M
3.7B	90min	0.5M

Table 3: The training time for each SAE and the number of buffered activations for each model size.

			Perplexity (dif.)		
α	m	Add	En	Ja	all
		-	17.57	19.54	15.39
0.05	1000	En	−0.08	−0.06	−0.07
		Ja	−0.07	−0.09	−0.08
		Bi	−0.10	−0.10	−0.10
	3000	En	−0.10	−0.07	−0.09
		Ja	−0.08	−0.15	−0.12
		Bi	−0.16	−0.19	−0.18
	5000	En	−0.12	−0.08	−0.10
		Ja	−0.09	−0.21	−0.15
		Bi	−0.21	−0.28	−0.25
0.10	1000	En	−0.08	−0.07	−0.08
		Ja	−0.06	−0.12	−0.09
		Bi	−0.12	−0.15	−0.14
	3000	En	−0.13	−0.09	−0.11
		Ja	−0.08	−0.25	−0.17
		Bi	−0.23	−0.34	−0.29
	5000	En	−0.16	−0.11	−0.14
		Ja	−0.10	−0.36	−0.24
		Bi	−0.33	−0.50	−0.42
0.20	1000	En	+0.13	+0.14	+0.13
		Ja	+0.18	+0.03	+0.10
		Bi	+0.05	−0.03	+0.01
	3000	En	+0.01	+0.10	+0.06
		Ja	+0.15	−0.25	−0.06
		Bi	−0.18	−0.40	−0.30
	5000	En	−0.07	+0.05	−0.01
		Ja	+0.10	−0.49	−0.21
		Bi	−0.37	−0.72	−0.56

Table 4: Baseline denotes the perplexity (PPL) of the mid-training model without any intervention.

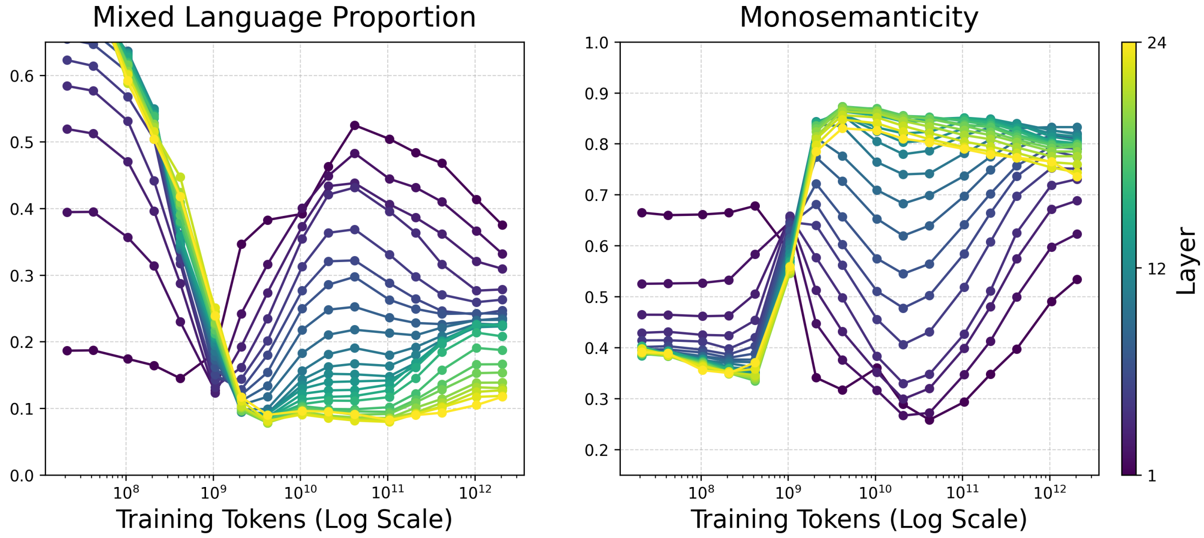


Figure 12: Layer-wise evolution of the mixed language proportion and the monosemanticity in the 1.8B model across training stages.

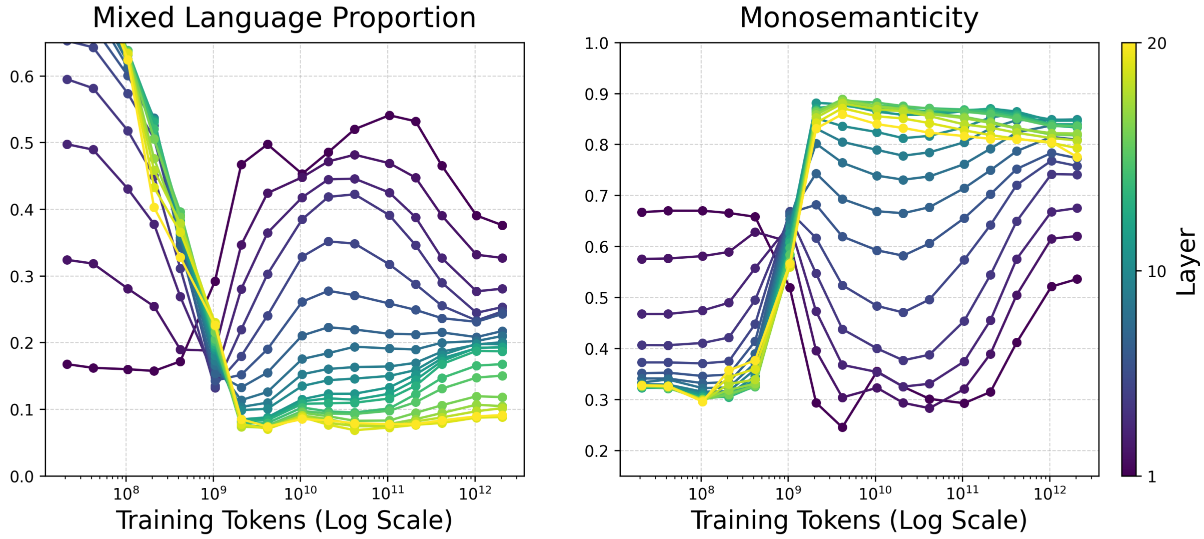


Figure 13: Layer-wise evolution of the mixed language proportion and the monosemanticity in the 980M model across training stages.

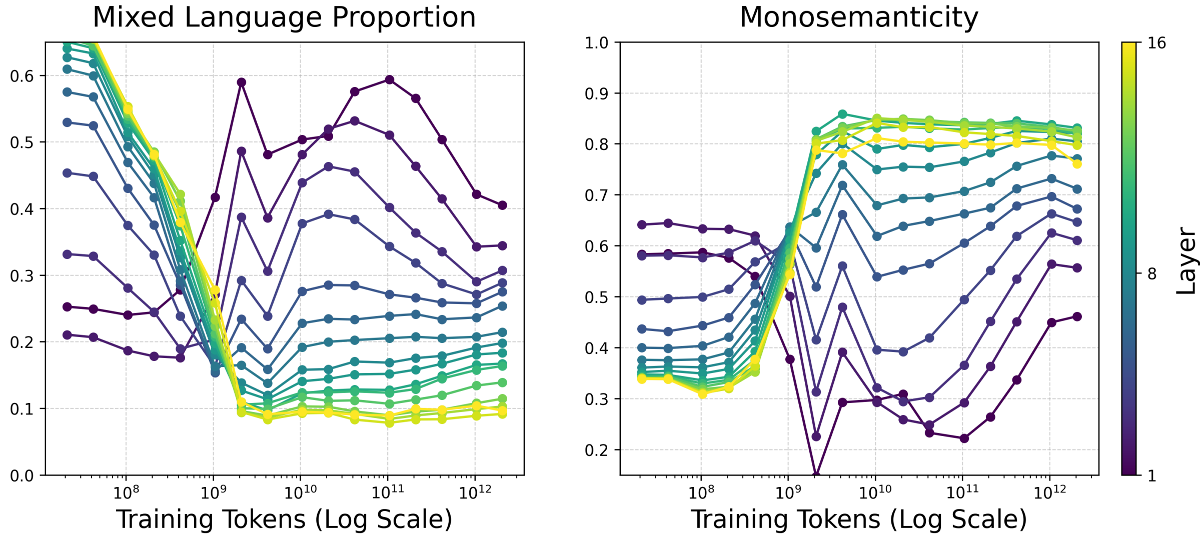


Figure 14: Layer-wise evolution of the mixed language proportion and the monosemanticity in the 440M model across training stages.

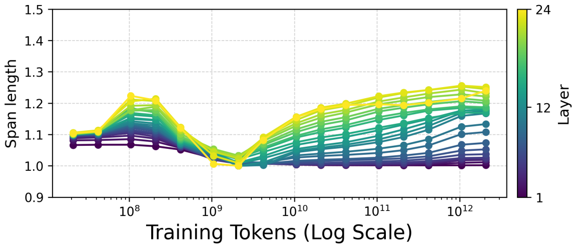


Figure 15: Layer-wise evolution of the span length average in the 1.8B model across training stages.

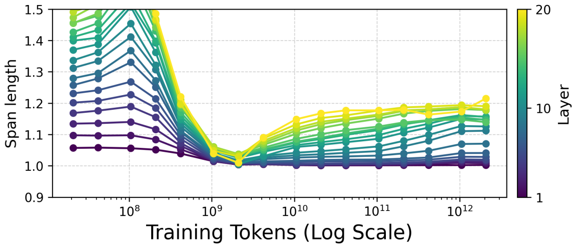


Figure 16: Layer-wise evolution of the span length average in the 980M model across training stages.

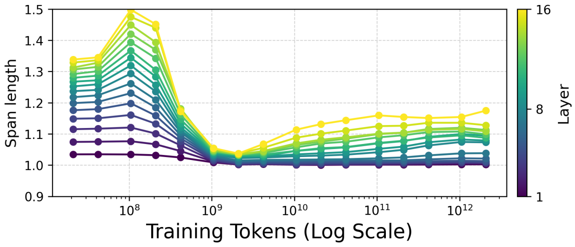


Figure 17: Layer-wise evolution of the span length average in the 440M model across training stages.