Crosscoding Through Time: Tracking Emergence & Consolidation Of Linguistic Representations Throughout LLM Pretraining

Deniz Bayazit[1], **Aaron Mueller**[2], **Antoine Bosselut**[1]
[1]EPFL, [2]Boston University
Correspondence: {deniz.bayazit,antoine.bosselut}@epfl.ch

Abstract

Large language models (LLMs) learn non-trivial abstractions during pretraining, such as detecting irregular plural noun subjects. However, because traditional evaluation methods (*e.g.*, benchmarking) fail to reveal how models acquire these concepts and capabilities, it is not well understood when and how these specific linguistic abilities emerge. To bridge this gap and better understand model training at the concept level, we use sparse crosscoders to discover and align features across model checkpoints. Using this approach, we track the evolution of linguistic features during pretraining. We train crosscoders between open-sourced checkpoint triplets with significant performance and representation shifts, and introduce a novel metric, Relative Indirect Effects (RELIE), to trace training stages at which individual features become causally important for task performance. We show that crosscoders can detect feature emergence, maintenance, and discontinuation during pretraining. Our approach is architecture-agnostic and scalable, offering a promising path toward more interpretable and fine-grained analysis of representation learning throughout pretraining.[1]

1 Introduction

Among the foundational advances in deep learning is the ability to learn useful internal features through gradient-based optimization, rather than relying on hand-crafted representations (Rumelhart et al., 1986). This principle underlies much of the success of modern large language models (LLMs), where learned features can capture complex linguistic patterns during training (Manning et al., 2020). However, this unstructured learning comes at the expense of interpretability (Mueller et al., 2024), motivating new methods that measure whether particular concept representations are learned by LMs.

[1]The code, crosscoders, and annotations are available at https://github.com/bayazitdeniz/crosscoding-through-time

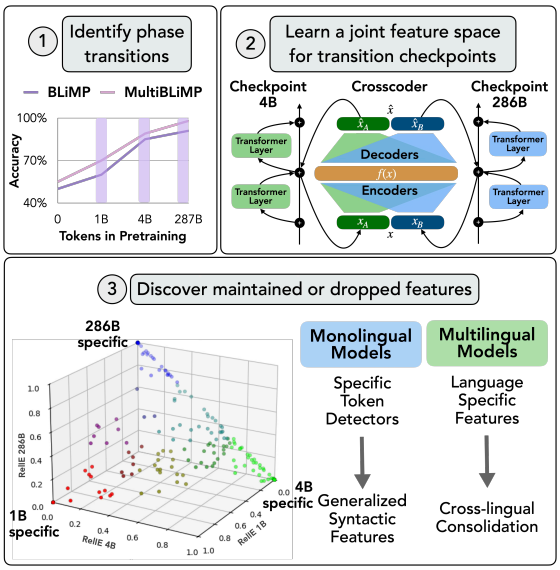


Figure 1: **Capturing the evolution of features.** Given a task, our pipeline selects the relevant checkpoints during pretraining, learns a joint feature space with crosscoders, and then analyzes causal feature importance across checkpoints. This allows to understand how models learn, maintain, or unlearn particular representations over time.

In particular, we lack a clear understanding of when and how specific linguistic abilities emerge during pretraining—a gap that, if bridged, would allow us to better understand LLM pretraining at the concept level. Common methods to estimate the acquisition of a concept include measuring the performance on tasks that act as proxies for the concept (Olsson et al., 2022; Chen et al., 2024), and identifying changes in the model's activation or parameter spaces (Wu et al., 2020). However, these measures only reveal *when* changes occur and fail to shed light on the mechanism by which a model internalizes particular linguistic concepts (*e.g.*, subject–verb agreement; Lovering et al., 2021; Bunzeck and Zarrieß, 2024; Kangaslahti et al., 2025).

Recently, sparse autoencoders (SAEs) have been adopted to study *how* such linguistic concepts are represented by models. SAEs project a model's

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1353–1377
July 2-7, 2026

dense internal representation at particular layers onto large, sparsely activated feature spaces (Bricken et al., 2023; Huben et al., 2024), thereby discretizing activations into linear combinations of one-dimensional features. However, using SAEs to better understand the evolution of concepts during pretraining would require training unique SAEs for all checkpoints. These separately learned sparse feature spaces would preclude direct feature comparisons across training stages. To address these limitations, sparse crosscoders were introduced to learn a single *joint* feature space across layers or models simultaneously (Lindsey et al., 2024). This framework provides a structured lens for analyzing how linguistic concepts evolve over checkpoints: shared features indicate concepts that persist, while unique features reveal concepts that emerge or disappear. However, prior work has only used crosscoders to study features that arise during post-training (Minder et al., 2025; Baek and Tegmark, 2025).

In this work, we use crosscoders to track the evolution of syntactic concept representations across pretraining checkpoints, using the pipeline illustrated in Fig. 1. First, we learn crosscoders across checkpoint triplets showing behavioral and representational shifts on benchmarks such as BLiMP (Warstadt et al., 2020). We compare checkpoints from the same training run so that representational changes can be attributed to pretraining dynamics rather than to differences in tokenizer, data mixture, or objective. Then, we introduce the Relative Indirect Effect (RELIE) metric to causally quantify per-feature attribution over training checkpoints and annotate the role of the features. We validate RELIE through ablation and interpretability studies, assessing whether it accurately traces how and when features gain or lose task relevance.

We show that pairing crosscoders with our RELIE metric enables us to pinpoint linguistic concept representations at individual checkpoints and trace their development over time. This architecture-agnostic framework—validated on Pythia, BLOOM, and OLMo—scales easily to billion-parameter models. Qualitatively, we find that LLMs progressively build higher-level abstractions, as evidenced in how token and language-specific concepts gradually become abstracted into more universal concepts.

2 Related Work

Language Model Interpretability There has recently been significant progress in scaling unsupervised interpretability methods, including dictionary learning (Bricken et al., 2023; Huben et al., 2024), circuit discovery (Wang et al., 2023; Conmy et al., 2023; Bayazit et al., 2024), and work that combines the two (Marks et al., 2025). These have been instrumental in revealing *how* a final checkpoint performs tasks like subject–verb agreement (Marks et al., 2025), parenthesis matching (Huben et al., 2024), garden-path sentence processing (Hanna and Mueller, 2025), and crosslingual morphosyntactic generalization (Brinkmann et al., 2025). However, they offer limited insight into *when* specific concepts emerge. Crosscoders (Lindsey et al., 2024) address this gap by mapping joint feature spaces between models (*e.g.*, pretrained vs. instruction-tuned; Minder et al., 2025; Baek and Tegmark, 2025). We extend this approach to pretraining checkpoints to trace concept-level feature evolution and uncover both *when* and *how* representations emerge.

Training Dynamics A parallel line of work examines the learning trajectories of models via model performance, parameter shifts, and activation patterns across training steps (Saphra and Lopez, 2019; Wu et al., 2020; Kaplan et al., 2020; Liu et al., 2021). Some research aligns these dynamics with cognitive signals (*e.g.*, brain activity; Nakagi et al., 2025; AlKhamissi et al., 2025a,b; Constantinescu et al., 2025), or tracks knowledge acquisition over time (Liu et al., 2021; Ou et al., 2025; Cao et al., 2024; Zucchet et al., 2025). Such studies shed light on what general internal changes occur, but they do not enable precise concept-level claims and rarely tie these changes back to discrete or human-interpretable conceptual representations.

More recently, Kangaslahti et al. (2025) introduce POLCA, a method that analyzes training loss patterns to uncover hidden phase transitions among conceptually similar data samples. While POLCA reveals when a concept emerges through *loss dynamics*, our approach aims to trace how such a concept's role evolves over time by using the model's *hidden states*. Together, these complementary paradigms deepen our understanding of how pretraining shapes model behavior.

3 Preliminaries

Crosscoders SAEs[2] learn mappings from activations of model layers to feature spaces $\mathbf{f}$. Consequently, these mappings are unique to the corresponding model layer activations that are used as input during training, and cannot be used to disentangle what concepts might be shared or unique across different activation spaces (such as those in different model layers, or from different model checkpoints). Sparse crosscoders instead learn a joint feature space for activations from multiple sources, *e.g.*, from multiple checkpoints, denoted $C = \{c_1, c_2, \dots\}$. Crosscoders introduce three key modifications to the SAE paradigm: (1) dedicated encoder and decoder weights per source $c \in C$ (W^c_{dec}, $\mathbf{b}^c_{\text{dec}}$ and W^c_{enc}) to capture source-specific concepts;[3] (2) a joint reconstruction loss that is averaged across the different sources $\mathbf{x}_c$ and their reconstructions $\hat{\mathbf{x}}_c$; (3) an aggregated sparsity penalty summed across the sources to encourage the inclusion of both shared and unique features in the joint feature space. The final loss is then:

$$\mathbf{f} = \text{ReLU}\Big(\sum\nolimits_{c\in C} W^c_{\text{enc}}\mathbf{x}_c + \mathbf{b}_{\text{enc}}\Big) \quad (1)$$

$$\hat{\mathbf{x}}_c = W^c_{\text{dec}}\mathbf{f} + \mathbf{b}^c_{\text{dec}} \quad (2)$$

$$\mathcal{L} = \sum\nolimits_{c\in C}\|\mathbf{x}_c - \hat{\mathbf{x}}_c\|_2^2 + \sum\nolimits_{c\in C}\sum\nolimits_i \mathbf{f}_i\|W^c_{\text{dec},i}\|_2, \quad (3)$$

where i indexes a particular feature in $\mathbf{f}$ and the column W^c_{dec} that scales $\mathbf{f}_i$ when reconstructing $\hat{\mathbf{x}}_c$.

Measuring feature changes To determine whether a crosscoder feature $\mathbf{f}_i$ is unique to a particular checkpoint or shared between checkpoints, prior work proposes the relative decoder norm (RELDEC) $\in [0, 1]$ (Lindsey et al., 2024), computed per feature $\mathbf{f}_i$:

$$\text{RELDEC}_i = \frac{\left\|W^{c_2}_{\text{dec},i}\right\|_2}{\sum\limits_{c\in\{c_1,c_2\}}\left\|W^c_{\text{dec},i}\right\|_2} \quad (4)$$

For each checkpoint c and dictionary feature $\mathbf{f}_i$, the ℓ_2 norm is computed across the hidden model activation dimension that we aim to reconstruct. Then, the norms are scaled across features by their strength per model to find which feature is more specific to a given model, or shared. Values closer to 0 mean the feature is more present in c_1; those closer to 1 mean the feature is more present in c_2.

[2]We provide a formal definition of SAEs in Appendix A.

[3]Note that $\mathbf{b}_{\text{enc}}$ is shared across checkpoints.

Indirect Effect In the tasks we study, for a given prefix, *i.e.* a token set x, a single token identifies a correct (t_{correct}) or wrong (t_{wrong}) completion. To quantify each hidden unit's contribution to the correct completion—whether it is a neuron (*i.e.*, a dimension in the residual/layer output vector) or a crosscoder feature—we compute its indirect effect (IE) at each checkpoint by zero-ablating the feature (*i.e.*, setting its activation to 0) and measuring the change in a selected metric. To measure the significance of the feature towards the correct model behavior we compute the following log-probability difference as the primary metric m:

$$m(x) = \log p(t_{\text{wrong}} \mid x) - \log p(t_{\text{correct}} \mid x)$$

Specifically, IE is defined as the difference in $m(x)$ after and before the ablation $\mathbf{a}_{\text{patch}}$ (Pearl, 2001):

$$\text{IE}(m; \mathbf{a}; x) = m\big(x \mid \text{do}(\mathbf{a} = \mathbf{a}_{\text{patch}})\big) - m(x), \quad (5)$$

where the do operator replaces the original activation $\mathbf{a}$ with $\mathbf{a}_{\text{patch}}$. Intuitively, a positive IE indicates that ablating the unit pushes the model's prediction away from the correct class, while a negative IE means the ablation reinforces the correct prediction. In our work, we use integrated gradients ($\hat{\text{IE}}_{\text{ig}}$; Sundararajan et al., 2017; Marks et al., 2025) to approximate the IE of crosscoder features and use zero-ablation as patching. For more details on our IE implementation, see Appendix F.1.

4 Methodology

To understand how model representations evolve across training, we aim to identify which features emerge, persist, or disappear over time. This requires attributing representations to *specific* training checkpoints, a task we refer to as checkpoint representation attribution. We tackle this in three steps: (1) Identify the critical checkpoints via performance and activation correlation analyses; (2) Learn a crosscoder between these critical checkpoints; (3) Attribute features using the Relative Indirect Effect (see Eq. 6 below), and track emerging, maintained, or vanishing representations.

Phase Transition Identification Building on prior work that flags sudden jumps (*i.e.*, phase transitions) in validation accuracy, loss, or activation-pattern similarity (Wu et al., 2020; Chen et al., 2024; Nakagi et al., 2025), we track two signals from different checkpoints in tandem: (1) each checkpoint's accuracy on the target task, and (2) the

pairwise correlation of mid-layer activations across all checkpoints, averaged across task inputs. We focus on mid-layer activations because prior work has shown that they capture higher-level linguistic and cross-lingual abstractions (Tenney et al., 2019; Liu and Niehues, 2025; Brinkmann et al., 2025), while earlier and later layers are more closely tied to input processing and output prediction (Lad et al., 2024; Csordás et al., 2025). This makes the mid-layer activations a natural site for analyzing how such representations evolve over training. By plotting accuracy and the mid-layer activation similarity across training steps (later shown in Fig. 2), we identify when the model undergoes representational shifts.

Crosscoder Training for Checkpoint Evolution We train crosscoders under two regimes to trace the evolution of linguistic representations. First, we conduct triplet comparisons of phase-transition checkpoints (denoted by the number of tokens they have been trained on, *e.g.*, 1B ↔ 4B ↔ 286B) to gauge which features persist between phases.[4] Then, we analyze pairwise comparisons to verify that the takeaways from triplet comparisons are consistent with pairwise observations. In each setting, we also include the final training checkpoint of a particular model to see which features are present in the model's fully trained state.

Feature Selection & Annotation with RELIE The task-agnostic RELDEC (Eq. 4) separates checkpoint-specific features from shared ones by comparing *every* feature in the dictionary. While this yields a broad, task-agnostic view, it makes it difficult to isolate the features that actually drive performance on a target task and to interpret their role. Instead, we compute each feature's Indirect Effect (IE) per checkpoint at each token step using integrated gradients (Eq. 5), which directly quantifies a feature's contribution for the correct behavior. We then define the Relative Indirect Effect (RELIE) as the ratio of the absolute approximated IEs ($\hat{\text{IE}}_{\text{ig}}$) for checkpoints c_1 and c_2 for each feature $\mathbf{f}_i$, following the same normalization as Eq. 4 but applied to IEs:[5]

$$\text{RELIE}_{\text{2-way},i} = \frac{|\hat{\text{IE}}^{c_2}_{\text{ig},i}|}{|\hat{\text{IE}}^{c_1}_{\text{ig},i}| + |\hat{\text{IE}}^{c_2}_{\text{ig},i}|} \tag{6}$$

For three-checkpoint crosscoders, we compute a one-versus-all RELIE via:

$$\text{RELIE}_{\text{3-way},i} = \frac{\left(|\hat{\text{IE}}^{c_1}_{\text{ig},i}|, |\hat{\text{IE}}^{c_2}_{\text{ig},i}|, |\hat{\text{IE}}^{c_3}_{\text{ig},i}|\right)}{\sum_{c\in\{c_1,c_2,c_3\}} |\hat{\text{IE}}^{c}_{\text{ig},i}|} \tag{7}$$

We then select each checkpoint's top-10 IE features for the particular task, annotate them by the pretraining sequences that maximally activate them, and use RELIE to trace how their task relevance shifts across checkpoints.

5 Experimental Setup

Models & Crosscoders We evaluate three open-source LLM families with publicly released checkpoints: Pythia 1B (Biderman et al., 2023), OLMo 1B (Groeneveld et al., 2024), and BLOOM 1B (Scao et al., 2023). Pythia's dense checkpoint logging lets us understand early linguistic feature emergence more precisely; OLMo's extended training helps us study feature maintenance over longer pretraining; and BLOOM's multilingual corpus allows us to trace crosslingual representation development. Following prior SAE work at similar model scale (Lieberum et al., 2024), our crosscoders use a dictionary size of 2^{14} features. We train crosscoders on the output hidden states extracted from the middle transformer block of each checkpoint. Additional details are provided in Appendix C & D.

Crosscoder Training Datasets Guided by the principle that a system's behavior reflects its training distribution (McCoy et al., 2024), we train each crosscoder on a subset of its model's original pretraining data subsampled to 400M tokens. For Pythia, we sample from the Pile (Gao et al., 2020); for OLMo, we sample from Dolma (Soldaini et al., 2024); and for BLOOM, we subsample mC4 (Xue et al., 2021) in proportion to the top ten languages represented in ROOTS (Laurençon et al., 2023).

Linguistic Tasks To chart the acquisition of subject–verb agreement representations in LLMs, we use the BLiMP (Warstadt et al., 2020), MultiBLiMP (Jumelet et al., 2025), and CLAMS (Mueller

[4] A notable challenge is that very early checkpoints often resist sparse mapping and accurate reconstruction; prior work shows SAEs falter on fully random models (Karvonen et al., 2024), but their behavior on *partially-trained* checkpoints remains underexplored. We verify that our crosscoders remain robust even when incorporating these early checkpoints (§6.2).

[5] Ablations in Appendix E show that by focusing on task-relevant signal rather than the entire feature set, RELIE uncovers more meaningful task-specific features.

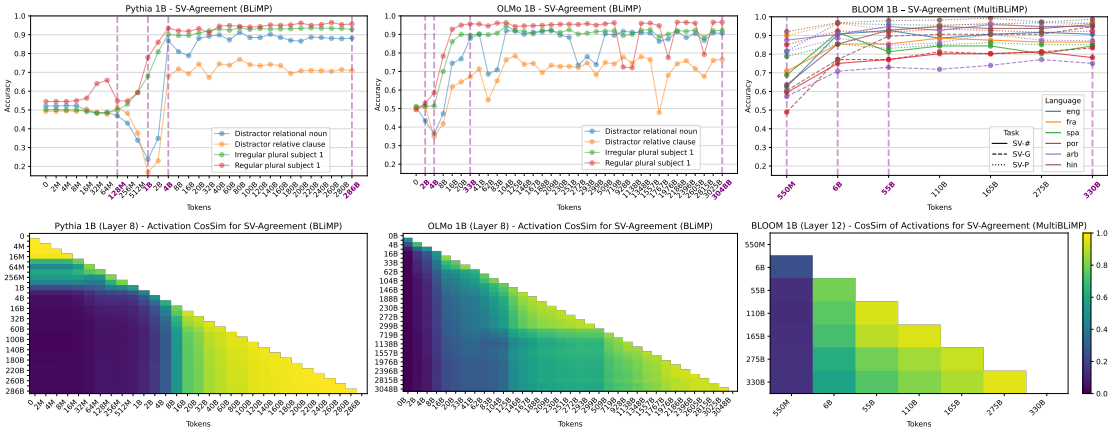


Figure 2: **Checkpoint selection with task performance (top) and average middle-layer activation cosine similarity (bottom).** The Pythia-1B (left) and OLMo-1B (middle) performance and activations patterns are calculated using BLiMP whereas BLOOM-1B (right) uses MultiBLiMP. All columns have the same x-axis (number of training tokens). We highlight checkpoints identified as critical in purple vertical lines. While some activation shifts align with performance jumps, others reveal continued representational change even after accuracy plateaus.

et al., 2020) benchmarks. These benchmarks are especially useful for our causal analysis because they are built around controlled minimal pairs, allowing us to compare model behavior on sentence pairs that differ only in the targeted grammatical phenomenon. At the same time, they cover a range of agreement-related subtasks that vary in grammatical case, language, and difficulty (*e.g.*, BLiMP includes verb agreement cases with regular vs. irregular plural noun subjects). We preprocess them by finding examples where the difference between a correct and wrong completion is determined by a single token for a given prefix. See Appendix B for processing details and examples.

6 Crosscoder Learnability for Intermediate Checkpoints

6.1 Checkpoint Selection via Phase Transition Identification

Before training crosscoders, we identify checkpoints where phase transitions in model behavior or representations occur. To uncover phase transitions, we track subject–verb agreement accuracy (BLiMP, MultiBLiMP) and the cosine similarity of middle-layer activations across checkpoints for Pythia-1B, OLMo-1B, and BLOOM-1B (Fig. 2).[6]

In Pythia-1B, a significant phase transition unfolds from 128M to 4B tokens: accuracy on BLiMP vaults from near-chance (∼50%) to above 90%, and the activation similarity heatmap shows a significant change compared to earlier near-random checkpoints. These indicators signal the emergence of syntactic representations. A smaller inflection at 1B hints that certain subtasks (*e.g.*, agreement with irregular plural subjects) are learned before more complex ones at 4B (*e.g.*, agreement with distractor clauses), after which performance remains stable until 286B. Hence, we study checkpoints that have been trained on {128M, 1B, 4B, 286B} tokens.

OLMo-1B follows a more staggered trajectory: an initial milestone at 2B boosts accuracy and activation similarity, followed by a second adjustment phase at 4B, and a consolidation near 33B tokens. After 33B, while accuracy mostly plateaus, activations continue refining through 3T, indicating ongoing representational changes. Thus we select the {2B, 4B, 33B, 3T} checkpoints for OLMo-1B.

Applying the same analysis to BLOOM-1B on MultiBLiMP, while accuracy jumps for all subtasks at 6B, the extent differs for particular languages like English vs. Arabic. Beyond 55B tokens, performance plateaus but continues to undergo subtle refinement of the activation space until 341B, hence the choice of {550M, 6B, 55B, 341B} checkpoints.

6.2 Crosscoder Learnability

We train crosscoders on pairs and triplets of checkpoints using subsampled model training data (§5). As seen in Table 6 (Appendix D), across three random seeds and checkpoints at different training stages, crosscoders consistently reconstruct intermediate activations with little increase in cross entropy loss compared to the original loss (mostly $\Delta\text{CE} < 0.2$), even for earlier checkpoints that are

[6] All observed average pairwise cosine similarity values were non-negative, so we restrict the heatmap range to $[0, 1]$.

RelIE	FeatID	Interpreted Function	Top Activating Sequence
1B-4B shared			
[0.53, 0.33, 0.15]	1067	Detects subtoken *-ans* in various contexts	... pick-ups, vans, and larger vehicles such ...
[0.45, 0.41, 0.14]	4897	Detects subtoken *-ists*, (*e.g.*, protagonist, capitalist, pharmacist)	... a conspiracy by elitists within government and big business ...
1B-286B shared			
[0.52, 0.01, 0.46]	3852	Detects *man* subtoken,often the singular noun but sometimes not (*e.g.*, manned aircraft)	... bad omens: The man in charge of the reactor ...
[0.55, 0.10, 0.34]	7489	Detects singular *woman* noun	... other people –; including a woman with far too many cats ...
4B specific			
[0.00, 1.00, 0.00]	11274	Multi-word noun or compound noun detector in particular for scientific writing	... two-dimensional pair correlation function and the structure factor
[0.02, 0.68, 0.30]	10523	Detects plural nouns depicting humans (*e.g.*, people, students, bloggers)	Most older scientists said such a thing is ...
286B specific			
[0.08, 0.15, 0.77]	14228	Multi-word named entity and title detectors (*e.g.*, proper nouns, locations etc.)	Buick Lesabre Ownership Costs There
[0.00, 0.18, 0.82]	6746	Detects deverbal nouns and nominalizations formed from verbs	Though my reactions tend to be tepid ... However its performance degrades with ...
[0.00, 0.01, 0.99]	14623	Detects prepositions	...so that an inclusion bias due to restricted use in specific

Table 1: **Subset of 3-way Crosscoder Annotations for Pythia-1B | 1B↔4B↔286B tokens.** RELIE gives the one-versus-all attribution vector; Interpreted function describes any detected linguistic role. Rows are grouped by whether features are unique to one checkpoint, shared between two, or common to all. Top activating sequence shows an example yielding a high activation, where color density shows the activation intensity for individual tokens. Features evolve from token-specific detectors to group-level abstract concepts (*e.g.*, deverbal noun detectors), even after accuracy plateaus.

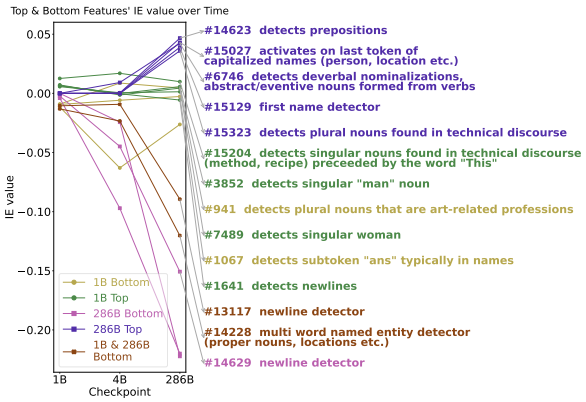


Figure 3: **IE evolution of Top-5 & Bottom-5 IE Features for Pythia checkpoints 1B & 286B.** IEs are calculated using BLiMP subject–verb agreement tasks. Missing annotation means the feature was not interpretable. We observe that low-level or uninterpretable features fade over time, while high-level grammar detectors emerge and strengthen by 286B.

trained on an order-of-magnitude less data. Larger differences in tokens used for pretraining lead to more dead features (*i.e.*, hidden units in the joint feature space that never get activated) and slightly higher ΔCE ($\sim$0.35), but they remain largely recoverable.

7 Emergence of Agreement Generalizations in Monolingual Models

Having established crosscoder mappings at critical checkpoints, we examine whether monolingual models develop broader syntactic intuitions. Specifically, we investigate whether LLMs initially rely primarily on surface-level token matching mechanisms and then progressively internalize deeper grammatical abstractions. To evaluate this, we track how features evolve across training stages, annotating and quantifying the top and bottom IE features at each checkpoint.

7.1 From Specific Token Detectors to High-Level Syntactic Features

Fig. 3 tracks the top-5 and bottom-5 IE features from Pythia's early (1B) to final (286B) checkpoints. We observe a sharp decline in low-level token detectors (*e.g.*, subtokens or non-interpretable features, depicted as those that do not have annotations) alongside a rise in grammatical detectors (*e.g.*, prepositions, plural-noun classes). A similar but longer-horizon trend appears in OLMo (Fig. 6, Appendix G), where for the rest of the training until 3T tokens, more abstract grammatical concept detectors emerge, such as those identifying plural nouns that depict jobs or skill attributes.

The manual annotations for Pythia-1B in Table 1 illustrate this shift more in detail. Early checkpoints (1B-4B) employ detectors for specific tokens (*e.g.*, the token *-ans* in different contexts) or irregular forms (*e.g.*, *woman* vs. *women*). By 4B, the model instead favors abstract, group-level concepts, such as detecting multi-word nouns common in scientific writing or nouns denoting groups of humans. Between 4B and 286B, although overall performance plateaus, features with

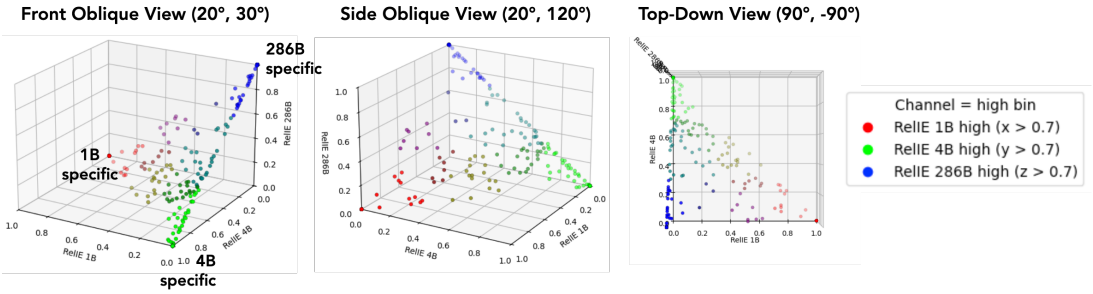


Figure 4: **RELIE of Top-100 IE features on BLiMP for Pythia-1B checkpoints 1B, 4B, 286B.** Distinct clusters at each corner show attribution of checkpoint-specific features. 4B–286B share more abstract features, while 1B–286B overlaps are sparse and lower-level features.

RelIE	FeatID	Activated Languages	Interpreted Function	Top Activating Sequence
6B specific				
[1.00, 0.00, 0.00]	3672	arb,eng,fra,hin,por,spa	Detects ellipsis and question/exclamation marks	... ou mesurage sur place, Ville/Région :
[0.78, 0.14, 0.07]	7122	eng	Main-verb head detector	... one day, Ananya opens up about herself and her
[0.62, 0.17, 0.20]	15288	fra,por,spa	Detects *Ev*, *ev*, *év* subtokens (only in latin languages)	... les résultats ont été moins évidents.
6B-55B shared				
[0.56, 0.27, 0.17]	15758	eng	Detects head of multi-token or compound nouns	The nanogenerator generates output ...
[0.36, 0.36, 0.28]	12525	eng,fra,hin,por,spa	Boss concept detector (*e.g.*, *chief*, *jefe*, *chefs*, *chefe*, मुख्य - boss)	O chefe do departamento, Prof. ...
[0.35, 0.41, 0.24]	15248	eng	Detects the token *that* (only in English)	good number of analogies that can apply to ...
55B-341B shared				
[0.05, 0.31, 0.64]	6997	arb,eng,fra,hin,por,spa	Proper-noun/ID detector that activates on named-entity heads	observers, the June 2015 King v. Burwell decision
6B-55B-341B shared				
[0.35, 0.32, 0.32]	12140	arb,eng,fra,por,spa	Multilingual relative pronoun detector (*e.g.*, *que*, *that*, *who*, *aladhi*)	predijo que los editores que prosperarán en el futuro
[0.32, 0.31, 0.37]	4610	eng,fra,por,spa	Phrasal-verb/PP-complement detector that fires on first token of verb-plus-particle or adjective-plus-preposition pattern	... army was accustomed to in Europe. In the ...
[0.39, 0.26, 0.35]	5819	arb,eng,fra,spa	Activates most on new beginning of clauses right after a punctuation and wanes until a new clause	some readers may recall I cut my night photography

Table 2: **Subset of 3-way Crosscoder CLAMS French/English Annotation for BLOOM-1B | 6B↔55B ↔341B.** Languages are those that appeared when observing the feature's top-activating sentences. Early features are often language-specific, but over time these consolidate into crosslingual detectors capturing shared syntax and semantics.

targeted linguistic functions—such as detectors for deverbal nominalizations—continue to emerge and strengthen (see Appendix F for a complete list of list annotation). We additionally observe the same trend with the pairwise crosscoder comparisons (Table 8), where the earlier checkpoint 128M employs more token-specific and edge-case detecting features while later checkpoints such as 4B develop detectors for functional token groups (*e.g.*, plural quantifiers, such as *many*, driving verb agreement).

7.2 Feature Trajectories: Quantifying Emerging, Persistent, and Vanishing Causal Features via RELIE

To track how the causal importance of features redistributes across three checkpoints (*e.g.*, 1B, 4B, and 286B), we conduct a three-way RELIE analysis (Eq. 7), shown in Fig. 4. Each axis corresponds to the one-versus-all RELIE score for one checkpoint: a value of 1.0 means that a feature only plays a causal role for that checkpoint, while 0.0 means it has no effect on model behavior at that checkpoint. When we visualize the top-100[7] IE features using this method, several patterns emerge.

High-importance features are primarily shared between 4B and 286B, consistent with their matching performance. However, many features also cluster near the 4B- and 286B-specific corners, indicating that even after performance plateaus, the model's internal concepts continue to evolve and new features continue to arise.

There is also substantial overlap across all three checkpoints and between 1B and 4B, where annotations indicate a shift from token-specific patterns to more abstract grammatical role detectors. Interestingly, despite being a sparser region, some features appear to be shared between 1B and 286B, at (0.5, 0.0, 0.5) coordinates. However, these shared features are generally less interpretable than those in other regions, often activating on sequences of random tokens, punctuation, or newlines. A version of the plot highlighting the top-10 IE features is provided in Appendix G.

[7]Given the min IE threshold 0.1, some tasks yield fewer than 100 features; in those cases, we use all that surpass it.

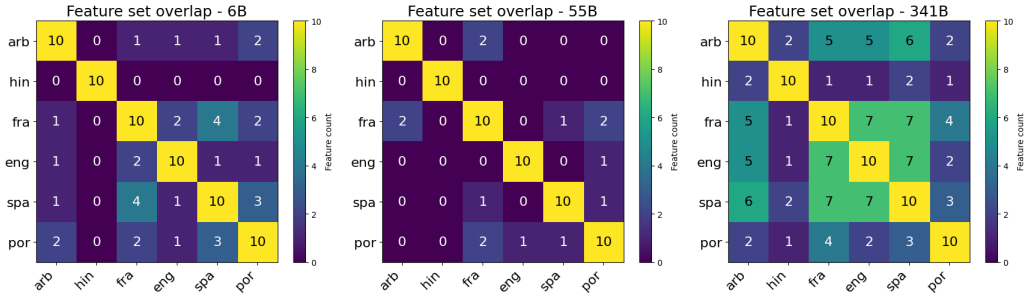


Figure 5: **Top-10 MultiBLiMP Number Agreement Task Feature Overlap per Checkpoint for Languages with 3-way comparisons.** Cross-lingual feature overlap increases by 341B, especially for Latin-script languages, reflecting shared syntactic patterns, while greater morphological complexity in Arabic and Hindi limits alignment.

8 Crosslingual Alignment of Syntactic Features in Multilingual Models

Building on our findings in monolingual models—where lower-level, token-specific detectors give way to more abstract grammatical features—we now examine how multilingual LMs like BLOOM learn to share features across languages.

8.1 Consolidation of Monolingual Features into Multilingual Ones

In Table 2, we show a subset of annotations for features significant for the CLAMS task in English and French. We observe that features of early checkpoints, (*e.g.*, 6B) often have individual features for specific languages. For instance, instead of a single crosslingual feature that detects main-verb heads, the model maintains a separate detector for English. The crosslingual features found at this early stage are punctuation and delimeter detectors, which can be easier to abstract due to shared punctuation scripts. According to our pairwise checkpoint comparison, monolinguality of features in early phases also holds for conjunction and relative pronoun detecting features. As training progresses, these features merge into crosslingual ones. Checkpoints 6B and 55B, for example, share many such features among which we note the emergence of higher-level semantic detectors, such as a crosslingual *boss* concept feature, which likely reflects the prominence of that noun in our IE dataset. In the final stages we find crosslingual features for more complex constructions, such as one for detecting adjective-plus-preposition patterns.

8.2 Quantifying Cross-Lingual Feature Alignment and its Limits

To quantify the number of crosslingual features over time, we calculate their overlap across five languages available in BLOOM and MutliBLiMP (Arabic, Hindi, English, French, Spanish, and Portuguese) and three subtasks (number, person, and gender agreement — SV-#, SV-P, and SV-G). In particular, we compute an IE score for each language–subtask pair and then measure how many of their top-10 features intersect. We repeat this for each model checkpoint in our 3-way comparison crosscoder (6B $\leftrightarrow$ 55B $\leftrightarrow$ 341B). Fig. 5 illustrates the SV-# trend: first, a moderate overlap in the beginning of the training (6B) followed up by a mild overlap decrease for French, English, Spanish, and Portuguese at 55B (a pattern that we also observe for the remaining tasks in Fig. 8 in Appendix G). Despite the 55B checkpoint having similar performance on the task as 341B, the latter has a significantly higher number of overlapping features, especially for Latin script languages. We hypothesize two reasons: (1) the shared script leading to high overlap of token-specific features from the beginning of pretraining; and (2) having verbs that agree with their subjects in fairly predictable ways where nouns also often agree with adjectives in gender/number. On the other hand, Hindi and Arabic have more complex agreement systems (*e.g.*, in Hindi, verbs agree with both subject and object in person/number/gender depending on verb aspect).

This raises an important question: how do LMs handle agreement in languages with greater morphological complexity? To unpack why Hindi has lower overlap, we examine the top IE-annotated features per language in Appendix Tables 13 (English), 14 (French), and 15 (Hindi). We find that high IE features for Hindi encode more information on the verbal aspect and the object than English and French. Importantly, we find that a majority of the overlap between Hindi and other languages is due to punctuation or parenthesis detecting low-level features. Our analysis shows that while multi-

lingual models can learn a joint feature space for languages with similar morphological systems, languages with more complex or under-represented agreement mechanisms—like Hindi and Arabic—may retain language-specific representations even at larger scales for this particular middle layer in BLOOM. Future work should further examine whether and how such language-specific representations persist across layers.

9 Conclusion

We deployed crosscoders to learn joint feature spaces between model checkpoints and introduced RELIE to show when individual features become causally important for task performance. This lets us trace fine-grained changes in linguistic representations over pretraining, including which features emerge, persist, and disappear. We find that monolingual models transition from detecting specific tokens to high-level syntactic patterns, while multilingual models consolidate these into universal crosslingual features, reflecting increasingly shared representations. This approach generalizes across architectures and scales to billion-parameter models. Future work could extend this analysis from individual features to circuits (Tigges et al., 2024; Hakimi et al., 2025), tracing how distributed patterns of computation evolve over training and across layers, rather than relying solely on static slices of the model.

Acknowledgements

We are grateful to Madhur Panwar, Negar Foroutan, Badr AlKhamissi, Gail Weiss, and Anja Surina for their helpful discussions and feedback on our manuscript. Antara Bhattacharya and Yugesh Kothari provided essential support in verifying the Hindi feature annotations. We also thank Julian Minder for insights on training crosscoders and Michael Hanna for the LaTeX code used to visualize sequence activations in tables.

We also gratefully acknowledge the support of the Swiss National Science Foundation (No. 215390), the European Research Council (Starting grant no. 101222478, RESPECT-LM), the AI2050 program at Schmidt Sciences (Grant #G-25-69783), Sony Group Corporation, and the Swiss National Supercomputing Center (CSCS) in the form of an infrastructure engineering and development project.

Limitations

Our analysis is dependent on checkpoint selection, which can influence the takeaways drawn from the study. Early pretraining stage checkpoints are particularly hard to interpret, as features derived from them are generally less human-interpretable; thus, not all checkpoints will yield immediate insights with this setup. Additionally, our reliance on benchmarks like BLiMP may not fully capture the diversity of real-world linguistic variation; this limits the generalizability of our findings.

The annotation process itself involves a degree of subjectivity, and our use of gradient attribution methods, such as integrated gradients, does not strictly guarantee causal relationships.[8] Finally, there is a risk of misinterpretation: it would be easy to falsely alias features into stable or human-like conceptual spaces, which could mislead downstream use or public understanding (Saphra, 2022). In the other direction, it is also not guaranteed that all atomic concepts used by the model are human understandable or explainable with natural language (Hewitt et al., 2025), similar to AlphaGo Zero's surprising "nonstandard strategies beyond the scope of traditional Go knowledge" (Silver et al., 2017).

Finally, the computational cost of training crosscoders is non-trivial, although it is relatively short for typical SAE training due to our focus on the middle layer activations: we coarsely estimate 6 hours on an A100 80GB GPU for 2-way comparisons and 12 hours for 3-way comparisons in 1B models. Finally, scaling beyond 7B parameters across many checkpoint comparisons presents challenges, though this can potentially be mitigated by computing each source separately and aggregating the crosscoder latents iteratively.

Ethics Statement

Our findings on crosslingual feature consolidation may help reveal language-specific underperformance and inequities, contributing to better understanding of multilingual model fairness. However, sharing detailed mappings of internal model representations carries a dual-use risk: while it can aid safety researchers, it may also enable malicious actors to design more sophisticated adversarial attacks or exploitations.

[8] That said, past work has observed strong correlations between gradient attributions and exact interventions, especially when using integrated gradients (Marks et al., 2025)

References

Badr AlKhamissi, Greta Tuckute, Antoine Bosselut, and Martin Schrimpf. 2025a. The LLM language network: A neuroscientific approach for identifying causally task-relevant units. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10887–10911, Albuquerque, New Mexico. Association for Computational Linguistics.

Badr AlKhamissi, Greta Tuckute, Yingtian Tang, Taha Binhuraib, Antoine Bosselut, and Martin Schrimpf. 2025b. From language to cognition: How llms outgrow the human language network. *Preprint*, arXiv:2503.01830.

David D. Baek and Max Tegmark. 2025. Towards understanding distilled reasoning models: A representational approach. In *ICLR 2025 Workshop on Building Trust in Language Models and Applications*.

Deniz Bayazit, Negar Foroutan, Zeming Chen, Gail Weiss, and Antoine Bosselut. 2024. Discovering knowledge-critical subnetworks in pretrained language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6549–6583, Miami, Florida, USA. Association for Computational Linguistics.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, and 1 others. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Dan Braun, Jordan Taylor, Nicholas Goldowsky-Dill, and Lee Sharkey. 2024. Identifying functionally important features with end-to-end sparse dictionary learning. *Preprint*, arXiv:2405.12241.

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

Jannik Brinkmann, Chris Wendler, Christian Bartelt, and Aaron Mueller. 2025. Large language models share representations of latent grammatical concepts across typologically diverse languages. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6131–6150, Albuquerque, New Mexico. Association for Computational Linguistics.

Bastian Bunzeck and Sina Zarrieß. 2024. Fifty shapes of BLiMP: syntactic learning curves in language models are not uniform, but sometimes unruly. In *Proceedings of the 2024 CLASP Conference on Multimodality and Interaction in Language Learning*, pages 39–55, Gothenburg, Sweden. Association for Computational Linguistics.

Bart Bussmann, Patrick Leask, and Neel Nanda. 2024. Batchtopk sparse autoencoders. *Preprint*, arXiv:2412.06410.

Boxi Cao, Qiaoyu Tang, Hongyu Lin, Shanshan Jiang, Bin Dong, Xianpei Han, Jiawei Chen, Tianshu Wang, and Le Sun. 2024. Retentive or forgetful? diving into the knowledge memorizing mechanism of language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14016–14036, Torino, Italia. ELRA and ICCL.

Angelica Chen, Ravid Shwartz-Ziv, Kyunghyun Cho, Matthew L Leavitt, and Naomi Saphra. 2024. Sudden drops in the loss: Syntax acquisition, phase transitions, and simplicity bias in MLMs. In *The Twelfth International Conference on Learning Representations*.

Arthur Conmy, Augustine Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, volume 36, pages 16318–16352. Curran Associates, Inc.

Ionut Constantinescu, Tiago Pimentel, Ryan Cotterell, and Alex Warstadt. 2025. Investigating critical period effects in language acquisition through neural language models. *Transactions of the Association for Computational Linguistics*, 13:96–120.

Róbert Csordás, Christopher D. Manning, and Christopher Potts. 2025. Do language models use their depth efficiently? *Preprint*, arXiv:2505.13898.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

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*.

Dirk Groeneveld, Iz Beltagy, Evan Walsh, Akshita Bhagia, Rodney Kinney, Oyvind Tafjord, Ananya Jha, Hamish Ivison, Ian Magnusson, Yizhong Wang,

Shane Arora, David Atkinson, Russell Authur, Khyathi Chandu, Arman Cohan, Jennifer Dumas, Yanai Elazar, Yuling Gu, Jack Hessel, and 24 others. 2024. OLMo: Accelerating the science of language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15789–15809, Bangkok, Thailand. Association for Computational Linguistics.

Ahmad Dawar Hakimi, Ali Modarressi, Philipp Wicke, and Hinrich Schuetze. 2025. Time course MechInterp: Analyzing the evolution of components and knowledge in large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12633–12653, Vienna, Austria. Association for Computational Linguistics.

Michael Hanna and Aaron Mueller. 2025. Incremental sentence processing mechanisms in autoregressive transformer language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3181–3203, Albuquerque, New Mexico. Association for Computational Linguistics.

Michael Hanna, Sandro Pezzelle, and Yonatan Belinkov. 2024. Have faith in faithfulness: Going beyond circuit overlap when finding model mechanisms. In *First Conference on Language Modeling*.

Dan Hendrycks and Kevin Gimpel. 2023. Gaussian error linear units (gelus). *Preprint*, arXiv:1606.08415.

John Hewitt, Robert Geirhos, and Been Kim. 2025. We can't understand ai using our existing vocabulary. *Preprint*, arXiv:2502.07586.

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*.

Jaap Jumelet, Leonie Weissweiler, and Arianna Bisazza. 2025. Multiblimp 1.0: A massively multilingual benchmark of linguistic minimal pairs. *Preprint*, arXiv:2504.02768.

Sara Kangaslahti, Elan Rosenfeld, and Naomi Saphra. 2025. Hidden breakthroughs in language model training. *Preprint*, arXiv:2506.15872.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *Preprint*, arXiv:2001.08361.

Adam Karvonen, Benjamin Wright, Can Rager, Rico Angell, Jannik Brinkmann, Logan Smith, Claudio Mayrink Verdun, David Bau, and Samuel Marks. 2024. Measuring progress in dictionary learning for language model interpretability with board game models. In *Advances in Neural Information Processing Systems*, volume 37, pages 83091–83118. Curran Associates, Inc.

János Kramár, Tom Lieberum, Rohin Shah, and Neel Nanda. 2024. Atp*: An efficient and scalable method for localizing llm behaviour to components. *Preprint*, arXiv:2403.00745.

Vedang Lad, Wes Gurnee, and Max Tegmark. 2024. The remarkable robustness of LLMs: Stages of inference? In *ICML 2024 Workshop on Mechanistic Interpretability*.

Hugo Laurençon, Lucile Saulnier, Thomas Wang, Christopher Akiki, Albert Villanova del Moral, Teven Le Scao, Leandro Von Werra, Chenghao Mou, Eduardo González Ponferrada, Huu Nguyen, Jörg Frohberg, Mario Šaško, Quentin Lhoest, Angelina McMillan-Major, Gerard Dupont, Stella Biderman, Anna Rogers, Loubna Ben allal, Francesco De Toni, and 35 others. 2023. The bigscience roots corpus: A 1.6tb composite multilingual dataset. *Preprint*, arXiv:2303.03915.

Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, Janos Kramar, Anca Dragan, Rohin Shah, and Neel Nanda. 2024. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 278–300, Miami, Florida, US. Association for Computational Linguistics.

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

Danni Liu and Jan Niehues. 2025. Middle-layer representation alignment for cross-lingual transfer in fine-tuned LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15979–15996, Vienna, Austria. Association for Computational Linguistics.

Zeyu Liu, Yizhong Wang, Jungo Kasai, Hannaneh Hajishirzi, and Noah A. Smith. 2021. Probing across time: What does RoBERTa know and when? In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 820–842, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Charles Lovering, Rohan Jha, Tal Linzen, and Ellie Pavlick. 2021. Predicting inductive biases of pretrained models. In *International Conference on Learning Representations*.

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

Christopher D. Manning, Kevin Clark, John Hewitt, Urvashi Khandelwal, and Omer Levy. 2020. Emergent linguistic structure in artificial neural networks trained by self-supervision. *Proceedings of the National Academy of Sciences*, 117(48):30046–30054.

Samuel Marks, Can Rager, Eric J Michaud, Yonatan Belinkov, David Bau, and Aaron Mueller. 2025. Sparse feature circuits: Discovering and editing interpretable causal graphs in language models. In *The Thirteenth International Conference on Learning Representations*.

R. Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D. Hardy, and Thomas L. Griffiths. 2024. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences*, 121(41):e2322420121.

Julian Minder, Clément Dumas, Caden Juang, Bilal Chugtai, and Neel Nanda. 2025. Overcoming sparsity artifacts in crosscoders to interpret chat-tuning. *Preprint*, arXiv:2504.02922.

Aaron Mueller, Jannik Brinkmann, Millicent Li, Samuel Marks, Koyena Pal, Nikhil Prakash, Can Rager, Aruna Sankaranarayanan, Arnab Sen Sharma, Jiuding Sun, Eric Todd, David Bau, and Yonatan Belinkov. 2024. The quest for the right mediator: A history, survey, and theoretical grounding of causal interpretability. *Preprint*, arXiv:2408.01416.

Aaron Mueller, Garrett Nicolai, Panayiota Petrou-Zeniou, Natalia Talmina, and Tal Linzen. 2020. Cross-linguistic syntactic evaluation of word prediction models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5523–5539, Online. Association for Computational Linguistics.

Yuko Nakagi, Keigo Tada, Sota Yoshino, Shinji Nishimoto, and Yu Takagi. 2025. Triple phase transitions: Understanding the learning dynamics of large language models from a neuroscience perspective. *Preprint*, arXiv:2502.20779.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2025. 2 olmo 2 furious. *Preprint*, arXiv:2501.00656.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Scott Johnston, Andy Jones, Jackson Kernion, Liane Lovitt, and 7 others. 2022. In-context learning and induction heads. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2022/in-context-learning-and-induction-heads/index.html.

Yixin Ou, Yunzhi Yao, Ningyu Zhang, Hui Jin, Jiacheng Sun, Shumin Deng, Zhenguo Li, and Huajun Chen. 2025. How do llms acquire new knowledge? a knowledge circuits perspective on continual pre-training. *Preprint*, arXiv:2502.11196.

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

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*.

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

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *nature*, 323(6088):533–536.

Naomi Saphra. 2022. Interpretability creationism.

Naomi Saphra and Adam Lopez. 2019. Understanding learning dynamics of language models with SVCCA. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3257–3267, Minneapolis, Minnesota. Association for Computational Linguistics.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, and 373 others. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Noam Shazeer. 2020. Glu variants improve transformer. *Preprint*, arXiv:2002.05202.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy P. Lillicrap, Fan Hui, L. Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. 2017. Mastering the game of go without human knowledge. *Nature*, 550:354–359.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai

Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, and 17 others. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3319–3328. PMLR.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. 2019. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy. Association for Computational Linguistics.

Curt Tigges, Michael Hanna, Qinan Yu, and Stella Biderman. 2024. LLM circuit analyses are consistent across training and scale. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

Alex Warstadt, Alicia Parrish, Haokun Liu, Anhad Mohananey, Wei Peng, Sheng-Fu Wang, and Samuel R. Bowman. 2020. BLiMP: The benchmark of linguistic minimal pairs for English. *Transactions of the Association for Computational Linguistics*, 8:377–392.

John Wu, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2020. Similarity analysis of contextual word representation models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4638–4655, Online. Association for Computational Linguistics.

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

Nicolas Zucchet, Jörg Bornschein, Stephanie Chan, Andrew Lampinen, Razvan Pascanu, and Soham De. 2025. How do language models learn facts? dynamics, curricula and hallucinations. *Preprint*, arXiv:2503.21676.

A Preliminaries Continued

Sparse Autoencoders Sparse Autoencoders (SAE) learn to to reconstruct a model's dense internal representation by projecting it onto a larger yet sparsely activating feature space with an encoder, and then decoding it back into the original activations. Formally, one way to implement SAEs is to enforce a sparsity and a reconstruction objective, such as the ℓ_1 sparsity and ℓ_2 reconstruction loss:

$$\mathbf{f} = \text{ReLU}(W_{\text{enc}}\mathbf{x} + \mathbf{b}_{\text{enc}}) \tag{8}$$

$$\hat{\mathbf{x}} = W_{\text{dec}}\mathbf{f} + \mathbf{b}_{\text{dec}} \tag{9}$$

$$\mathcal{L} = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 + \lambda \sum_i \mathbf{f}_i \|W_{\text{dec},i}\|_2 \tag{10}$$

Beyond this implementation, there are a variety of alternative objectives. Some change how the ReLU activation function is applied (*e.g.*, replacing it with JumpReLU; Rajamanoharan et al., 2024). Others modify the sparsity objective by directly applying a top-k activation constraint instead of a open-ended ℓ_1 sparsity loss (Makhzani and Frey, 2014; Bussmann et al., 2024; Gao et al., 2025). Some also replace the direct reconstruction objective with a KL divergence loss between the model's original output distribution as the reference and its output when using the reconstructed activation in the forward pass (Braun et al., 2024).

B Dataset Details

In this section we describe the datasets and the preprocessing steps. Note that we did not collect these datasets ourselves; they may contain personally identifying or offensive content.

Pile and DOLMA Subsampling To train crosscoders for Pythia and OLMo, we use the Pile and DOLMA datasets respectively (Gao et al., 2020; Soldaini et al., 2024). Specifically, we randomly subsample around 400M tokens from each dataset for training and around 120'000 tokens for the validation split. All reported metrics are calculated with the validation split.

mC4 Subsampling with ROOTS Ratios We also subsample 400M tokens for BLOOM's crosscoder training dataset. However, because the original training dataset ROOTS is spread across multi-

Dataset	Subtask	Prefix	Correct	Wrong
BLiMP	Distractor relational noun	The granddaughters of every customer ___	don	doesn
	Irregular plural subject	The octopi ___	have	has
MultiBLiMP	Spanish	Desde algún lugar donde habita el recuerdo ___	fue	fui
	French	Cette proposition d'amendement a pour but que l'on tienne compte de les grands froids que _	peuvent	pouvez
CLAMS	Simple Agreement	The farmer ___	is	are
	VP Coord	Les clients retournent et démén ___	agent	age

Table 3: **Examples from subject–verb agreement tasks,** showing shared prefix with correct and wrong single-token completions. For BLiMP, we use the Pythia tokenizer, and for the other two multilingual datasets, we use BLOOM. BLiMP and CLAMS subtasks differ in terms of grammatical case and difficulty, while MultiBLiMP and CLAMS additionally include multilingual examples.

ple repositories on Huggingface Hub[9] (Laurençon et al., 2023), we instead extract the same ten most-frequent languages from mc4 in identical proportions (Xue et al., 2021): English (35%), Chinese (19%), French (15%), Spanish (13%), Portuguese (6%), Arabic (5%), Vietnamese (3%), Hindi (2%), Indonesian (1%), and Bengali (1%).

Subject–Verb Agreement Task Examples We evaluate subject–verb agreement using three datasets: BLiMP (Warstadt et al., 2020), MultiBLiMP (Jumelet et al., 2025), and CLAMS (Mueller et al., 2020). Across all three benchmarks we apply a unified preprocessing pipeline: we identify a common prefix and isolate the tokens whose prediction reflects a correct versus incorrect agreement, which makes the process model-specific. We provide two examples from each task in Table 3.

In BLiMP, we focus on four subtasks: (1) `Distractor agreement relational noun`, (2) `Distractor agreement relative clause`, (3) `Regular plural subject verb agreement 1`, (4) `Irregular plural subject verb agreement 1`. For the latter two subtasks, we omit their version 2 as they do not match our requirements for a single token completion given a single shared prefix. The processing yields a dataset containing 3577 entries for both Pythia and OLMo, with roughly similar amounts of examples from each subtask.

For MultiBLiMP, we restrict to the six languages that both appear in BLOOM's top ten and have sufficient data (English, French, Spanish, Portuguese, Arabic, and Hindi). We use all valid examples when computing language-specific Indirect Effects (IEs): 575 examples for English, 1593 for French, 1242 for Spanish, 1481 for Portuguese, 692 for Arabic, and 964 for Hindi), but sample uniformly when we learn the IEs for several languages at once. When uniform sampling, each language has 100 examples from SV-#, 100 examples from SV-G (except English, French, and Spanish, where no examples exist in MultiBLiMP), and 290 from SV-P.

For CLAMS, we only include the two overlapping languages with BLOOM, which are English and French, and we subsample from both using subtasks (1) `long_vp_coord` (300 for both), (2) `obj_rel_across_anim` (400), (3) `obj_rel_within_anim` (400), (4) `prep_anim` (400), (5) `simple_agrmt` (80), (6) `subj_rel` (400), and (7) `vp_coord` (400), resulting in a total of 4760 examples.

C Model Details

All experiments use the 1B-parameter variants of Pythia, OLMo(1), and BLOOM. We choose to use OLMo(1)-1B instead of OLMo2-1B (OLMo et al., 2025) as the former has more frequent checkpointing during pretraining. These models differ in depth and hidden dimension size, as shown in Table 5. Pythia and OLMO's 1B version have 16 layers, whereas BLOOM has 24, hence why we learn a crosscoder at the 8th layer for Pythia and OLMo, and at the 12th layer for BLOOM.

They also differ in their tokenizers, position embeddings, attention implementation, and activation functions, which can affect how models process their mid-layer output. All three models use a BPE style tokenizer (Sennrich et al., 2016). In addition, OLMo and Pythia uses RoPE embeddings (Su et al., 2024), while BLOOM uses ALiBi (Press et al., 2022). BLOOM also implements a multi-query attention where each head has its own query but shares key and value projections, whereas the other models have separate key/value/query per heads. Finally Pythia and BLOOM use a GeLU

[9] https://huggingface.co/bigscience-data/datasets

Hyperparameter	Value
seeds	[124, 153, 6582]
num_train_tokens	400M
train_batch_token_num	4096
val_batch_token_num	8184
num_val_batches	30
dict_size	16384
dec_init_norm	0.08
enc_dtype	fp32
lr	5e-05
l1_warmup_pct	0.05
l1_coeff	2
beta1	0.9
beta2	0.999

Table 4: **Hyperparameters for Crosscoder Training.**

Hyperparameter	Pythia-1B and OLMo-1B	BLOOM-1B
hidden_dim	2048	1536
num_layers	16	24
mid_layer	8	12

Table 5: **Model-specific Hyperparameters.** Note that the HF model names we used to load these models are `pythia-1b`, `OLMo-1B-0724-hf`, `bloom-1b1-intermediate`.

activation function (Hendrycks and Gimpel, 2023), whereas OLMo uses SwiGLU (Shazeer, 2020).

D Crosscoder Training Details

Hardware For training we use a single 80GB NVIDIA A100 GPU. Under our chosen hyperparameters, pairwise (2-way) comparisons converge in about 6 hours, while 3-way experiments require closer to 12. By contrast, training sparse autoencoders for several layers can stretch into multiple days. Because our batch sizes, learning rates, and model dimensions fit comfortably within one 80GB GPU's memory, we did not pursue any data or model parallelization schemes.

Hyperparameters We share our hyperparameters in Tables 4 and 5. All crosscoders are trained using three fixed random seeds, $\{124, 153, 6582\}$. Training proceeds until approximately 400M tokens have been processed. Optimization is performed with Adam, using a peak learning rate of 5×10^{-5}, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The ℓ_1 penalty on decoder-weighted activations is ramped up linearly over the first 5% of training, reaching a coefficient of $\lambda = 2$. Decoder weights are randomly initialized with norm 0.08, and we employ a dictionary size of $16\,384$.

		ℓ_1 Sparsity Crosscoder				
Model	Comparison	ℓ_0	Dead Feats	ΔCE A	ΔCE B	ΔCE C
Pythia-1B Layer 8	128M ↔ 1B	88	9	0.00	0.07	-
	1B ↔ 4B	214	1	0.05	0.18	-
	4B ↔ 286B	190	9	0.15	0.48	-
	1B ↔ 4B ↔ 286B	215	19	0.03	0.16	0.54
OLMo-1B Layer 8	2B ↔ 4B	184	0	0.08	0.20	-
	4B ↔ 33B	227	0	0.14	0.21	-
	33B ↔ 3048B	182	425	0.16	0.35	-
	4B ↔ 33B ↔ 3048B	225	101	0.12	0.18	0.43
BLOOM-1B Layer 12	550M ↔ 6B	211	6	0.10	0.20	-
	6B ↔ 55B	112	8	0.14	0.29	-
	55B ↔ 341B	96	12	0.18	0.18	-
	6B ↔ 55B ↔ 341B	118	19	0.13	0.20	0.22

Table 6: **Crosscoder statistics.** Results averaged over three seeds on validation set. ΔCE is the change in cross-entropy loss when doing a forward pass using the original output versus the crosscoder reconstruction. A, B, C refer to the 1st, 2nd and 3rd checkpoints used for loss computation. ℓ_0 and dead feature averages are rounded to integers. Less trained models (*e.g.*, 1B) get smaller ΔCE values than further trained models (*e.g.*, 286B) due to the former's high original CE loss.

Crosscoder Learning Results In Table 6 we show the averaged metrics of trained crosscoders across three seeds. ℓ_0 records how many crosscoder features fire on average per token. Dead features refer to the number of features that were never activated across all validation batches. ΔCE shows the difference in cross-entropy loss when doing a forward pass using the original mid-layer output versus the crosscoder reconstruction. A, B, C refer to the first, second and third checkpoints used for loss computation. For a detailed analysis, see §6.2 in the main paper.

E Attribution Correlation via Feature Ablation

To assess whether RELIE's focus on task-relevant signals yields a more targeted identification of significant features than RELDEC, we perform an ablation study on the top-10 features deemed most important by IE on each BLiMP subtask (see Appendix B). For each checkpoint, we measured the change in log probability difference upon ablating each feature (one-by-one), denoted Δc, and then computed the ratio $|\Delta c_2|/|\Delta c_1|$ to quantify which checkpoint's predictions were more adversely affected. Given that 1.0 means more relevant for c_2 and 0.0 for c_1 for both RELIE and RELDEC, we expect a high positive correlation between the two if the feature was indeed more important for one checkpoint versus the other.

Table 7 presents Spearman correlations (ρ) between the ratios of log-probability-difference-differences $|\Delta c_2|/|\Delta c_1|$ and two checkpoint-

Task	Pythia-1B			OLMo-1B		
	Comparison	$\rho(\frac{\lvert\Delta c_2\rvert}{\lvert\Delta c_1\rvert}, \text{RelDec})$	$\rho(\frac{\lvert\Delta c_2\rvert}{\lvert\Delta c_1\rvert}, \text{RelIE})$	Comparison	$\rho(\frac{\lvert\Delta c_2\rvert}{\lvert\Delta c_1\rvert}, \text{RelDec})$	$\rho(\frac{\lvert\Delta c_2\rvert}{\lvert\Delta c_1\rvert}, \text{RelIE})$
Distractor Relational Noun	128M ↔ 1B	0.316	**0.934**	2B ↔ 4B	0.920	**0.972**
	1B ↔ 4B	0.930	**0.958**	4B ↔ 33B	**0.949**	0.897
	4B ↔ 286B	0.691	**0.964**	33B ↔ 3048B	0.770	**0.973**
Distractor Relative Clause	128M ↔ 1B	0.788	**0.966**	2B ↔ 4B	0.961	**0.979**
	1B ↔ 4B	0.901	**0.922**	4B ↔ 33B	**0.956**	0.938
	4B ↔ 286B	0.784	**0.941**	33B ↔ 3048B	0.771	**0.989**
Irregular Plural Subject	128M ↔ 1B	0.941	**0.979**	2B ↔ 4B	**0.982**	0.966
	1B ↔ 4B	0.777	**0.937**	4B ↔ 33B	0.898	**0.905**
	4B ↔ 286B	0.843	**0.954**	33B ↔ 3048B	0.785	**0.810**
Regular Plural Subject	128M ↔ 1B	0.794	**0.948**	2B ↔ 4B	0.838	**0.966**
	1B ↔ 4B	0.874	**0.930**	4B ↔ 33B	**0.961**	0.919
	4B ↔ 286B	0.806	**0.908**	33B ↔ 3048B	0.862	**0.956**
Avg by Comparison	Avg 128M ↔ 1B	0.710	**0.957**	Avg 2B ↔ 4B	0.925	**0.971**
	Avg 1B ↔ 4B	0.870	**0.937**	Avg 4B ↔ 33B	**0.941**	0.915
	Avg 4B ↔ 286B	0.781	**0.942**	Avg 33B ↔ 3048B	0.797	**0.932**
Overall Avg	-	0.787	**0.945**	-	0.843	**0.952**

Table 7: **Top-10 significant feature ablation for Pythia-1B and OLMo-1B.** Spearman correlations ρ between the ratio of log-probability-difference metrics and model-attributing scores (RELDEC, RELIE), across four subject–verb agreement phenomena and various phase transition comparisons. RELIE shows consistently higher correlations than RELDEC across tasks and model comparisons, indicating that focusing on task-relevant signal uncovers more meaningful and stable task-specific features.

attributing scores—RELDEC and RELIE—across four BLiMP subtasks. For both Pythia-1B and OLMo-1B, RELIE has higher correlation with the ratio (avg ρ = 0.945 and 0.952, respectively). In short, by focusing on task-relevant signal rather than the entire feature set, RELIE uncovers more interpretable and task-relevant features more effectively.

F Feature Annotations

F.1 Indirect Effect Implementation

As mentioned in §3, we approximate the Indirect Effect (IE) of each crosscoder feature using integrated gradients (IG; Sundararajan et al., 2017; Hanna et al., 2024; Marks et al., 2025). We use integrated gradients because they provide a principled attribution method with desirable axiomatic guarantees (Sundararajan et al., 2017), and prior work suggests that IG-based estimates of IE better match ablation-based measurements than alternative attribution methods (Kramár et al., 2024; Marks et al., 2025). Formally, we approximate IE as:

$$\hat{\text{IE}}_{\text{ig}}(m; \mathbf{a}; x) = (\mathbf{a}_{\text{patch}} - \mathbf{a}) \times \frac{1}{N} \sum_{\alpha} \nabla_{\mathbf{a}} m|_{\alpha \mathbf{a}_{\text{clean}} + (1-\alpha)\mathbf{a}_{\text{patch}}} \tag{11}$$

where the sum ranges over $N = 10$ equally-spaced $\alpha \in \{0, \frac{1}{N}, \ldots, \frac{N-1}{N}\}$ steps. This gradient-based formulation allows us to estimate the IE for all features simultaneously using only a fixed number of passes, rather than scoring each feature via a separate intervention. Thus, it is an $O(1)$ algorithm rather than $O(n)$, where n is the number of features. After summing the gradients, we average the resulting IE scores across the batch (following Marks et al., 2025). Finally, we threshold the batch-averaged IE values, setting any below 0.1 to zero.

F.2 Annotation Instructions

Annotations were done by the authors. For languages that none of the authors spoke natively, the annotation was completed with the use of translators and verified with at least one native speaker. In particular, during the annotation process, the expert was not exposed to the RELIE values of the features; rather, they were asked to answer 4 questions adapted from Marks et al.:

1. **Description:** To the best of your extent, describe the behavior of this feature's activation.
2. **Interpretability:** On a scale of 0.0 to 1.0, how coherent are the examples shown with the description you wrote? Is it consistently activating on similar tokens or promoting/demoting similar tokens?
3. **Complexity:** On a scale of 0.0 to 1.0, how complex is the feature behavior? How broad is the topic that the feature fires on? Does the feature activate on or promote/demote diverse tokens or similar tokens all over again?
4. (if BLOOM) **Languages:** Which languages have this feature activated most on?

F.3 Complete Annotations

We provide here the full set of annotation tables omitted from the main paper: Pythia's two-way and three-way comparisons (Tables 8 and 10), OLMo's three-way comparison (Table 11), and BLOOM's two-way and three-way comparisons (Tables 9 and 12). Language-specific annotations for English, French, and Hindi appear in Tables 13, 14 and 15, respectively. We transliterated the Arabic text instead of rendering it in its native script, because the PDF LaTeX compiler could not render Devanagari and Arabic script simultaneously.

G Additional Analyses

In Fig. 6, we provide an additional plot showing the IE evolution of OLMo's top and bottom five IE features at the 4B and 3T checkpoints, complementing the Pythia evolution figure in the main paper (Fig. 3).

We also include an additional monolingual overlap plot in Fig. 7, which shows the top 10 IE features in addition to the top 100 shown in the main paper (Fig. 4).

Finally, we present the top-10 significant feature overlap counts from the multilingual two-way and three-way comparisons for all MultiBLiMP subtasks (number, person, and gender agreement) in Figures 8 and 9, respectively.

RelIE	FeatID	Interpreted Function
Comparison: 128M ↔ 1B		
128M specific		
0.19	5667	-
0.28	14250	Detects token *-ese* at the end of a word
0.29	440	Detects token *-ara* at the end of a name, promotes possession or verbs
128M-1B shared		
0.38	8636	-
0.43	3164	Detects *-us* ending, often for a Latin origin single noun and promotes verb *is*
0.52	12683	Detects the noun *analysis*
0.53	1749	Detects irregular plural noun *people*
0.56	5032	Detects irregular plural noun *men*, promotes EOS, conjunction, or verbs
0.57	7072	-
1B specific		
0.71	15882	Detects singular *man*, promotes preposition completion
0.83	4118	Detects singular *woman*, not necessarily as a subject
0.89	6381	-
0.91	10069	Detects nouns that end with *-ists* and promotes plural verb completion or prepositions
0.92	14897	Detects nouns that end with *-ans*
0.94	16118	Detects words ending with *-ias*
0.95	8757	-
1.00	3811	Detects regular plural nouns and promotes conjunction or prepositions
1.00	7483	Detects singular nouns preceded by *this*
Comparison: 1B ↔ 4B		
1B specific		
0.00	12677	Detects regular plural nouns that refer to groups of people and promotes plural verb completion or prepositions
0.10	5778	Detects *man* starting words but promotes multi-token word completions (*e.g.*, *-hood*, *-ned*, *-hattan*)
0.17	1440	Detects words containing the mid-token *-es* and promotes medical term completions (*e.g.*, *-es-ophagus*)
0.28	3737	Detects singular *woman*, not necessarily as a subject
1B-4B shared		
0.37	12685	Detects nouns that end with *-ans*
0.48	7616	Detects nouns that end with *-ists* and promotes plural verb completion or prepositions
0.53	14814	Detects singular nouns preceded by *this* in front of them, and promotes singular verb conjugation
0.68	11799	Detects regular plural nouns that refer to objects/science concepts and promotes plural verb completion or prepositions
4B specific		
0.82	9385	-
0.85	744	-
0.88	14210	Detecs regular plural nouns
1.00	13102	-
1.00	1868	Detects punctuations and newline to promote BOS words
1.00	4050	Detects nouns that are preceded by plural quantifiers (*e.g.*, *most*, *many*, *majority of*, *some*)
1.00	9326	Detects plural regular nouns, promotes plural conjugated verbs
1.00	9414	Comma detector
1.00	11088	Detects the final token of first names
1.00	14546	Detects HTML/code-related regular plural object nouns
Comparison: 4B ↔ 286B		
4B specific		
0.00	4368	-
0.00	2307	Detects regular plural nouns that refer to science concepts and promotes plural verb completion or prepositions
0.14	479	-
0.19	680	-
0.27	13452	Detects (regular and irregular) plural nouns, promotes plural verb completion
0.27	10514	Detects regular plural nouns that can be followed up with *themselves*
4B-286B shared		
0.37	15084	Detects (regular and irregular) plural nouns that refer to groups of people, promotes plural verb completion
0.56	5268	-
0.58	5129	-
286B specific		
0.79	14815	-
0.85	6511	-
0.89	5588	Detects newlines
0.92	12003	-
0.98	13244	Detects first names that are not followed up a last name
0.98	12108	Detects HTML/code-related plural object nouns
1.00	2139	Detects a larger variety of prepositions and complementizers (*e.g.*, *by*, *from*, *due*, *with*, *concerning*)
1.00	5138	Detects last token of multi-token first names, promotes last names

Table 8: **2-way L1-Sparsity Crosscoder Annotation for Pythia-1B.** Each block is one pairwise comparison. RELIE is sorted from 0.00 to 1.00, where < 0.3 gets attributed to the first checkpoint; > 0.7 to second; shared otherwise). Interpreted function gives a description if a linguistic role was detected, "–" otherwise. Pairwise comparisons reveal finer-grained feature shifts from one checkpoint to another, but cannot assess persistence like triplet analyses. This shows that early checkpoints (*e.g.*, 128M) capture low-level lexical and morphological patterns, while slightly further trained ones (*e.g.*, 1B) detect slightly more abstract patterns, such as irregular plurals.

RelIE	FeatID	Interpreted Function	Languages
Comparison: 550M ↔ 6B			
550M specific			
0.00	8760	Detects administrative/government-related nouns	fra
0.00	14133	Detects nouns and verbs that convey key actions, entities, or ideas in a sentence	eng
0.05	12275	Detects conjunction token *et*	fra
0.10	8341	Detects subtoken *et* typically conjunction but also for *et al.*	fra
0.20	15852	Detects head nouns and their modifiers that signal prominent participants or components	fra
0.22	14697	Detects plural nouns, promotes plural verb conjugation and *who* pronoun	eng,fra,spa
550M-6B shared			
0.36	1474	Detects plural French articles (*e.g.*, *les*, *nos*, *certains*), promotes plural single token noun completions	fra
0.39	8223	Promotes *-age* completion for nouns	eng,fra,spa
0.44	7882	Detects English relative pronoun *that*, promotoes pronoun follow ups	eng
0.69	14645	Detects *Ev* at BOS, to be completed with French adverbs or nouns	fra
6B specific			
0.95	12523	-	-
0.96	10853	Detects noun and nominal expressions representing abstract entities, events, or processes	eng,fra,por,spa
0.97	2189	Detects multi-word nouns (*e.g.*, compound nouns or with adjectives)	eng,fra,por,spa
1.00	9386	-	-
1.00	9813	*who* and *that* detector	arb,eng,fra,por,spa
1.00	10337	Punctuation and newline detector	arb,eng,fra,por,spa
1.00	14067	-	-
1.00	15428	Detects verbs with the concept to like/love/appreciate	eng,fra,hin*,por,spa
Comparison: 6B ↔ 55B			
6B specific			
0.00	11469	Detects punctuation and newline	eng,fra,spa
0.17	942	Verbs that depict dynamic, agentive actions	eng
0.20	10311	Detects *ev* or *Ev* tokens to be completed with french adverbs or nouns	fra
0.28	15632	Detects verbs with the concept to like/love/appreciate	arb*,eng,fra,por,spa
6B-55B shared			
0.31	11920	Detects head of noun phrases denoting concrete or informational entities (*e.g.*,*data*, *system*, *text*)	eng,fra,por,spa
0.32	12000	Detects noun that depict occupational or social roles like researchers, engineers, physician, and journalists	arb*,eng,fra,hin*,por,spa
0.32	2792	-	-
0.38	14748	-	-
0.40	5763	Detects the concept boss in different languages (*e.g.*, *chef*, *boss*, *jefe*) probably because it is a common noun in the IE dataset	eng,fra,spa
0.56	9817	Detects verbs and promotes preposition/conjunction/punctuation	eng,fra,por,spa
0.56	425	Detects relative pronoun and prepositions (*e.g.*, *that*, *que*, *at*, *of*, *de*)	eng,fra,spa
0.59	4863	Detects relative pronoun *that* and promotes verb/pronoun completions	eng
55B specific			
0.79	5345	Nouns and verbs related to consultants and consulting	arb*,eng,fra,por,spa
Comparison: 55B ↔ 341B			
55B specific			
0.27	15249	Nouns and verbs related to consultants and consulting	arb*,eng,fra,por,spa
0.27	12734	-	-
55B-341B shared			
0.35	11458	Detects *that* and promotes verb or pronoun completion in English	eng
0.39	11920	Detects nouns to be completed by *de*, *of*	eng,fra,por,spa
0.39	7339	Determiners and quantifiers in noun phrases, such as articles (*e.g.*, *a*, *os*), possessives (*e.g.* our), and universal quantifiers (*e.g.* every, todo, cada)	eng,por,spa
0.44	6063	Detects the concept boss in different languages (*e.g.*, *chef*, *boss*, *jefe*) probably because it is a common noun in the IE dataset and promotes *of* (*e.g.*, *de*, *do*, *du*) completions	eng,fra,por,spa
0.48	15083	Relative pronouns and the syntactic material inside relative clauses	eng,fra,por,spa
0.54	10325	Detects *ev* or *Ev* tokens to be completed with french adverbs or nouns	fra,por,spa
0.55	425	Detects relative pronouns/subordinators (*e.g.*, *that*, *que*, *qui*, *which*, *who*, *où*) to introduce a new clause; also activates on the verbs inside the subordinate clause	arb*,eng,fra,por,spa
0.57	8729	-	-
341B specific			
1.00	794	-	-
1.00	7419	Detects newlines in different languages	arb,fra,hin,por,spa
1.00	7806	-	-
1.00	13276	-	-
1.00	13404	Sentence-boundary detector through punctuation and other delimiters	arb,fra,por,spa,zh

Table 9: **2-way L1-Sparsity Crosscoder CLAMS French/English Annotation for BLOOM-1B.** Each block is one pairwise comparison. RELIE is sorted from 0.00 to 1.00, where < 0.3 gets attributed to the first checkpoint; > 0.7 to second; shared otherwise). Interpreted function gives a description if a linguistic role was detected, "–" otherwise. Languages lists which languages the feature highly activates on, * means that the activation was relatively less common. While earlier checkpoints (*e.g.*, 550M) capture language specific low-level function words, later checkpoints (*e.g.*, 55B and 341B) increasingly share such features across languages.

RelIE	FeatID	Interpreted Function
1B-4B shared		
[0.53, 0.33, 0.15]	1067	Detects subtoken *-ans* typically in names
[0.41, 0.39, 0.20]	941	Detects plural nouns that are art-related professions
[0.45, 0.41, 0.14]	4897	Detects plural nouns that end with *-ists*, (*e.g.*, *protagonist*, *capitalist*, *pharmacist*)
[0.32, 0.43, 0.25]	15204	Detects singular nouns found in technical discourse (*e.g.*, *method*, *function*, *guide*, *recipe*) preceeded by the word "This"
1B-286B shared		
[0.55, 0.10, 0.34]	7489	Detects singular *woman* noun
[0.52, 0.03, 0.45]	1641	Detects newlines
[0.52, 0.01, 0.46]	3852	Detects singular *man* noun
4B specific		
[0.00, 1.00, 0.00]	15556	Detects a full stop and promotes connection words or newlines
[0.00, 1.00, 0.00]	11274	Multi-word noun or compound noun detector
[0.00, 0.99, 0.01]	8318	Detects regular plural nouns
[0.00, 0.96, 0.04]	10020	-
[0.11, 0.69, 0.20]	15950	Detects regular plural nouns
[0.02, 0.68, 0.30]	10523	Detects plural nouns mostly depicting humans (*e.g.*, *people*, *students*, *bloggers*)
[0.11, 0.62, 0.26]	15118	-
4B-286B shared		
[0.01, 0.30, 0.69]	11987	-
286B specific		
[0.00, 0.00, 1.00]	15323	Detects plural nouns found in technical discourse
[0.08, 0.15, 0.77]	14228	Multi word named entity detector (proper nouns, locations etc.)
[0.00, 0.00, 1.00]	15027	Activates on last token of capitalized names (person, location etc.)
[0.00, 0.18, 0.82]	6746	Detects deverbal nouns / nominalizations, abstract/eventive nouns formed from verbs
[0.00, 0.10, 0.90]	5317	-
[0.01, 0.23, 0.76]	14629	Newline detector
[0.10, 0.08, 0.82]	13117	Newline detector
[0.00, 0.00, 1.00]	15129	First name detector
[0.00, 0.01, 0.99]	14623	Detects prepositions

Table 10: **3-way L1-Sparsity Crosscoder Annotation for Pythia-1B | Comparison 1B ↔ 4B ↔ 286B.** RELIE shows 3-way one-versus-all RELIE vector; Interpreted Function provides a description if a linguistic role was detected, and "–" otherwise. Rows are grouped by checkpoint specificity according to RELIE: features dominated by one checkpoint (1B, 4B, 286B specific); pairwise shared features (1B–4B, 1B–286B, 4B–286B shared); and shared across all (1B–4B–286B shared). A missing group means no such features found in the top-10 IE features of all checkpoints. RELIE-based triplet comparisons reveal that earlier checkpoints (*e.g.*, 1B and 4B) primarily detect low-level lexical and morphological patterns such as suffixes and irregular plurals, whereas later checkpoints (*e.g.*, 286B) increasingly specialize in higher-level syntactic and semantic functions, including named entity, nominalization, and technical discourse related noun detection.

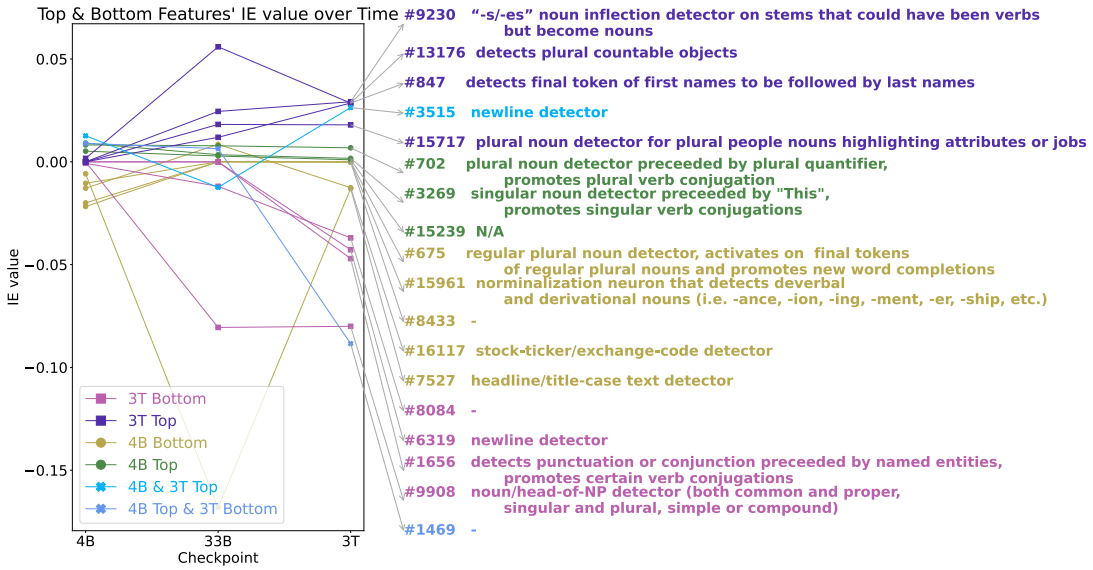


Figure 6: **IE evolution of Top-5 & Bottom-5 Features for OLMo-1B checkpoints 4B & 3T.** IEs are calculated using BLiMP subject–verb agreement tasks. "–" means the feature was not interpretable. In some cases, a feature can belong to multiple categories at once. Some low-level features, such as newline detectors, persist across training, whereas the usage of simpler lexical detectors fade as more abstract grammatical pattern detectors emerge.

RelIE	FeatID	Interpreted Function
4B specific		
[1.00, 0.00, 0.00]	675	Regular plural noun detector, activates on final tokens of regular plural nouns and promotes new word completions
[1.00, 0.00, 0.00]	10707	-
[0.99, 0.00, 0.01]	8433	-
[1.00, 0.00, 0.00]	15961	Nominalization feature that detects deverbal and derivational nouns (*i.e.*, *-ance*, *-ion*, *-ing* etc.)
[0.64, 0.24, 0.12]	3269	Singular noun detector preceeded by *This*, promotes singular verb conjugations
4B-33B shared		
[0.36, 0.34, 0.30]	702	Plural noun detector preceeded by plural quantifier (*e.g. most*, *some*), promotes plural verb conjugation
4B-3048B shared		
[0.38, 0.25, 0.37]	16117	Stock-ticker/exchange-code detector
33B specific		
[0.00, 1.00, 0.00]	5966	Detects commas followed by parenthetical clauses
[0.03, 0.90, 0.07]	7527	Headline/title-case text detector
[0.00, 0.80, 0.19]	10924	Detects first names that aren't followed up by last names
[0.00, 1.00, 0.00]	10692	Regular plural noun detector
33B-3048B shared		
[0.00, 0.50, 0.50]	9908	Noun/head-of-NP detector (both common and proper, singular and plural, simple or compound)
[0.00, 0.50, 0.50]	15717	Plural noun detector for plural people nouns highlighting attributes or jobs
[0.00, 0.52, 0.48]	14569	Detects last token of multi-token first names followed by last names
[0.00, 0.46, 0.54]	9230	*-s/-es* noun inflection detector on stems that could have been verbs but become nouns
[0.02, 0.65, 0.33]	847	Detects final token of first names to be followed by last names
3048B specific		
[0.25, 0.24, 0.51]	3515	Newline detector
[0.00, 0.29, 0.71]	13176	Detects plural countable objects
[0.01, 0.24, 0.75]	8084	-
[0.09, 0.06, 0.85]	1469	-
[0.00, 0.00, 1.00]	1656	Detects punctuation or conjunction preceeded by named entities, promotes certain verb conjugations
[0.00, 0.00, 1.00]	6319	Newline detector
[0.00, 0.00, 1.00]	5550	Newline detector that promotes certain sentence beginnings

Table 11: **3-way L1-Sparsity Crosscoder Annotation for OLMo-1B | Comparison 4B ↔ 33B ↔ 3048B.** Similar to Pythia, OLMo progresses from detecting lower-level lexical and morphological patterns in early checkpoints to more abstract grammatical and noun-phrase features later on, but OLMo may be retaining a stronger persistence of surface-level detectors (*e.g.*, newlines, suffixes) compared to Pythia's sharper shift.

RelIE	FeatID	Interpreted Function	Languages
6B specific			
[1.00, 0.00, 0.00]	3672	Detects ellipsis and question/exclamation marks	arb,eng,fra,hin,por,spa
[0.78, 0.14, 0.07]	7122	Main-verb head detector	eng
[0.83, 0.14, 0.03]	10388	Plural noun detector for several languages (*e.g.*, *players*, *usários* - users, *al naas* - the people, महिलाएं - ladies)	arb,eng,fra,hin,por,spa
[0.72, 0.23, 0.06]	9163	Noun-phrase head detector of multi-word noun chunk, activates on the key content (noun or adjective) that carries the meaning	eng,fra,por,spa
[0.62, 0.17, 0.20]	15288	Detects *Ev*, *ev*, *év* subtokens in different languages	eng,fra,por,spa
[0.59, 0.17, 0.24]	5704	-	-
6B-55B shared			
[0.35, 0.41, 0.24]	15248	Detects the token *that* only in English	eng
[0.36, 0.36, 0.28]	12525	Boss concept detector (*e.g.*, *chief*, *jefe*, *chefs*, *chefe*, मुख्य - boss)	eng,fra,hin,por,spa
[0.56, 0.27, 0.17]	15758	Detects head of multi-token or compound nouns	eng
55B specific			
[0.15, 0.63, 0.22]	10862	Predicts tokens related to the concept or form of *consult* (*e.g.*, 咨询, *fasr* - explain)	eng,fra,por,spa
55B-341B shared			
[0.05, 0.31, 0.64]	6997	Proper-noun/ID detector that activates on named-entity heads	arb,eng,fra,hin,por,spa
341B specific			
[0.00, 0.00, 1.00]	15193	Detects punctuation and parenthesis	arb,eng,fra,hin,spa,zh
[0.00, 0.00, 1.00]	2598	-	-
[0.00, 0.00, 1.00]	12151	-	-
[0.00, 0.00, 1.00]	9110	-	-
[0.00, 0.00, 1.00]	7066	-	-
[0.00, 0.00, 1.00]	6461	-	-
6B-55B-341B shared			
[0.35, 0.32, 0.32]	12140	Multilingual relative pronoun detector (*e.g.*, *que*, *that*, *who*, *aladhi*)	arb,eng,fra,por,spa
[0.32, 0.31, 0.37]	4610	Phrasal-verb/PP-complement detector that fires on the first token of a verb-plus-particle or adjective-plus-preposition pattern	eng,fra,por,spa
[0.39, 0.26, 0.35]	5819	Activates most on new beginning of clauses right after a punctuation and wanes until a new clause	arb,eng,fra,spa

Table 12: **3-way L1-Sparsity Crosscoder CLAMS French/English Annotation for BLOOM-1B | 6B ↔ 55B ↔ 341B.** The languages column depicts which languages appeared to also use this feature when observing the feature's top-activating sentences. Early checkpoints often rely on language-specific patterns (*e.g.*, English token detectors, suffix patterns), but later checkpoints increasingly learn cross-lingual and language-shared features, such as multilingual pronoun, noun-phrase, and clause detectors.

RelIE	FeatID	Interpreted Function	Languages
6B specific			
[0.94, 0.02, 0.04]	3672	Detects ellipsis and question/exclamation marks	arb,eng,fra,hin,por,spa
[1.00, 0.00, 0.00]	5273	Detects commas and full stop	eng,fra,hin,por,spa
[0.45, 0.28, 0.26]	849	Plural nouns / noun compounds that typically are the subject at BOS, promotes plural verb completion	eng
[0.50, 0.24, 0.26]	10974	3rd person plural pronoun *they* detector	eng
[0.59, 0.24, 0.17]	9163	Noun-phrase head detector, activates on the key content word (noun or adjective) that carries the meaning of multi-word noun chunk	eng,fra,por,spa
[0.70, 0.23, 0.07]	15758	Detects head of multi token or compound nouns	eng
6B-55B shared			
[0.44, 0.31, 0.25]	10235	Adverbial connectives (*e.g.*, *then*, *tambien*, *also*, *tambem*)	eng,por,spa
[0.40, 0.35, 0.26]	4332	Detects regular plural nouns	eng
[0.49, 0.37, 0.15]	5819	Activates most on new beginning of clauses right after a punctuation and wanes until a new clause	arb,eng,fra,spa
6B-341B shared			
[0.40, 0.21, 0.39]	5704	-	-
[0.45, 0.23, 0.32]	7007	People-related regular plural nouns (*e.g.*, *workers*, *investors*, *experts*)	eng
55B specific			
[0.19, 0.52, 0.29]	14073	Detects noun beginnings related to academic write-ups (*e.g.*, *dissertation*, *thesis*, निबंध - Hindi, *essai* - French)	eng,fra,hin
[0.00, 1.00, 0.00]	8086	Detects repeated interpuncts, most likely a training data artifact	-
341B specific			
[0.00, 0.00, 1.00]	7694	-	-
[0.00, 0.00, 1.00]	9110	-	-
[0.00, 0.00, 1.00]	15193	Detects punctuation and parenthesis	arb,eng,fra,hin,spa,zh
[0.10, 0.30, 0.60]	11280	-	-
[0.00, 0.00, 1.00]	6461	-	-
[0.01, 0.05, 0.93]	14020	-	-
[0.00, 0.04, 0.96]	2598	-	-
[0.00, 0.00, 1.00]	1638	Detects brackets/parenthesis/punctuation	eng,fra,por,spa
[0.00, 0.00, 1.00]	7066	-	-
[0.00, 0.00, 1.00]	12151	-	-

Table 13: **3-way L1-Sparsity Crosscoder MultiBLiMP English Annotation for BLOOM-1B | 6B $\leftrightarrow$ 55B $\leftrightarrow$ 341B.** While this annotation focuses on finding features with just English examples, many features in BLOOM still activate across multiple languages due to cross-lingual representations, showing that the model often leverages multilingual patterns such as shared connectives, pronouns, and beginning of clause detectors.

RelIE	FeatID	Interpreted Function	Languages
6B specific			
[1.00, 0.00, 0.00]	16271	Detects comma, punctuation possibly to mark new clauses	eng,fra,hin,por,spa,zh
[0.91, 0.01, 0.08]	3672	Detects ellipsis and question/exclamation marks	arb,eng,fra,hin,por,spa
[1.00, 0.00, 0.00]	5273	Detects commas and full stop	eng,fra,hin,por,spa
[1.00, 0.00, 0.00]	7529	Detects comma, punctuation possibly to mark new clauses	eng,fra,hin,por,spa,zh
[1.00, 0.00, 0.00]	11289	Detects comma, punctuation possibly to mark new clauses	eng,fra,hin,por,spa,zh
[0.51, 0.23, 0.26]	13809	1st person singular pronoun detector in diff surface forms (*e.g.*, *j'*, *Je*, *je*, *yo*)	fra,spa
[0.63, 0.24, 0.12]	8643	Plural first person pronoun detector, multilingual	arb,fra,por,spa
6B-55B shared			
[0.63, 0.26, 0.12]	14472	Plural second person pronoun detector	fra
6B-341B shared			
[0.33, 0.25, 0.42]	5819	Activates most on new beginning of clauses right after a punctuation and wanes until a new clause	arb,eng,fra,spa
[0.45, 0.24, 0.31]	12522	capitalized *Je* surface form detector, 1st person singular in French but also activates on names and *Jeux* game and *Yo*	eng,fra,spa
55B specific			
[0.00, 1.00, 0.00]	14378	-	-
[0.00, 1.00, 0.00]	6506	-	-
[0.00, 1.00, 0.00]	13738	-	-
[0.29, 0.61, 0.10]	16196	-	-
[0.20, 0.60, 0.21]	2471	Detects plural second person pronouns, multilingual	arb,eng,fra,hin
341B specific			
[0.00, 0.00, 1.00]	15193	Detects punctuation and parenthesis	arb,eng,fra,hin,spa,zh
[0.09, 0.01, 0.90]	14020	-	-
[0.02, 0.00, 0.98]	12151	-	-
[0.00, 0.00, 1.00]	1638	Detects brackets/parenthesis/punctuation	eng,fra,por,spa
[0.03, 0.00, 0.97]	7694	-	-
[0.02, 0.01, 0.97]	7066	-	-
[0.02, 0.00, 0.98]	6461	-	-
[0.00, 0.00, 1.00]	2598	-	-
[0.03, 0.00, 0.97]	9110	-	-

Table 14: **3-way L1-Sparsity Crosscoder MultiBLiMP French Annotation for BLOOM-1B | 6B $\leftrightarrow$ 55B $\leftrightarrow$ 341B.** Similar to the English examples only analysis, we find that many features found with a French dataset activate across multiple languages, reflecting BLOOM's shared cross-lingual representations. Some early detectors remain French-focused (*e.g.*, surface-form capitalization and plural pronouns), while later features, like punctuation and pronoun detectors, consolidate to be multilingual.

RelIE	FeatID	Interpreted Function	Languages
6B specific			
[0.79, 0.00, 0.21]	14483	Locative/in-marker में	hin
[0.67, 0.18, 0.16]	4192	Present-habitual marker using जाना / होना for singular-masculine subjects	hin
[0.67, 0.22, 0.11]	1174	The -ने participle ending marking habitual aspect, here with a masculine-plural (respectful) participle	hin
[0.66, 0.22, 0.13]	8471	Subordinating conjunction कि (“that”) introducing subordinate clauses	hin
[0.51, 0.24, 0.25]	2539	Perfective participle plural (and oblique-singular) ending used adjectivally	hin
6B-55B shared			
[0.45, 0.36, 0.20]	6215	Detects abstract singular nouns (chance, permission, responsibility, order, signal, shelter, danger, etc.)	hin
[0.40, 0.54, 0.07]	4338	Light-verb root कर used as “do” auxiliary in compound verbs (root + another light-verb + conjugation)	hin
6B-341B shared			
[0.64, 0.09, 0.27]	3969	Feminine possessive marker की (“of”/’s)	hin
[0.68, 0.05, 0.27]	4361	Masculine possessive marker का (“of”/’s)	hin
55B specific			
[0.19, 0.61, 0.20]	11884	Marker detecting the second noun or second element in a compound	hin
55B-341B shared			
[0.26, 0.31, 0.42]	4579	Nominalizer of “to be,” functioning as a gerund (“its being X,” “because of X being”)	hin
[0.05, 0.43, 0.52]	643	Detects first token of verbs / verb roots that appear before subject number conjugation	hin
[0.04, 0.37, 0.58]	2526	Perfective aspect marker in compounds like किया गया (“did/gave” in the perfective)	hin
[0.02, 0.54, 0.44]	1082	Inflection of the verb “to be” (हो) in the subjunctive/continuous mood (*e.g.*, हो सकता: हो चुका) indicating possibility or completed action	hin
341B specific			
[0.20, 0.21, 0.59]	11856	Negation marker नहीं placed before verbs	hin
[0.00, 0.00, 1.00]	2598	-	-
[0.00, 0.00, 1.00]	14020	-	-
6B-55B-341B shared			
[0.48, 0.26, 0.26]	2563	Plural pronoun marker for “you/you all” or “us/we”	hin

Table 15: **3-way L1-Sparsity Crosscoder MultiBLiMP Hindi Annotation for BLOOM-1B | 6B ↔ 55B ↔ 341B.** When using just Hindi subject–verb agreement examples to find highest IE features, most features remain strongly language-specific, focusing on tense, aspect, case, and possessive markers, with fewer cross-lingual activations compared to English and French. This may be due to Hindi’s lower representation in the BLOOM training data (2%) relative to French (15%), which limits the emergence of more shared, language-agnostic detectors.

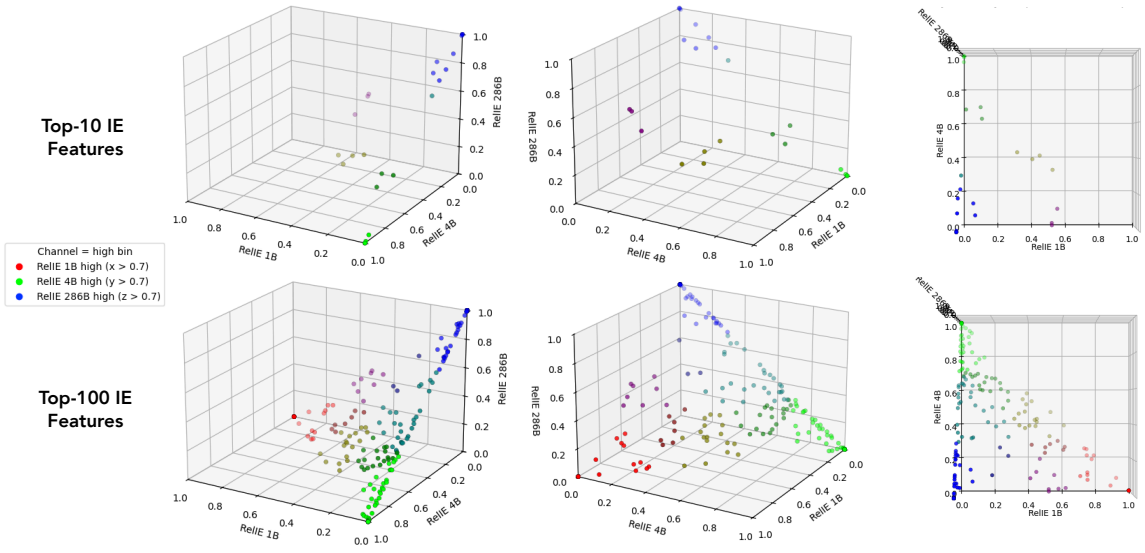
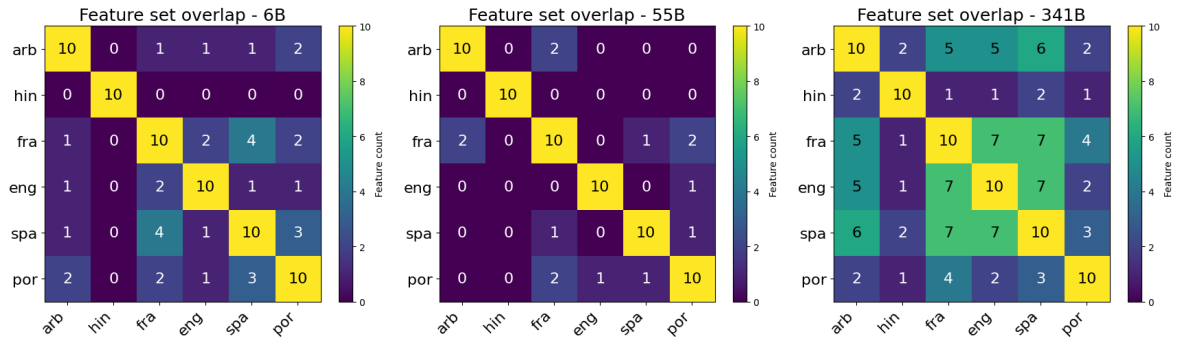
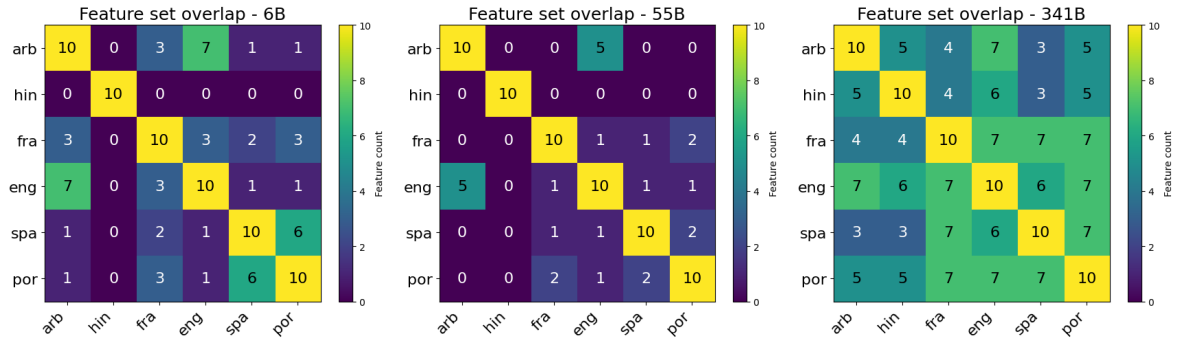


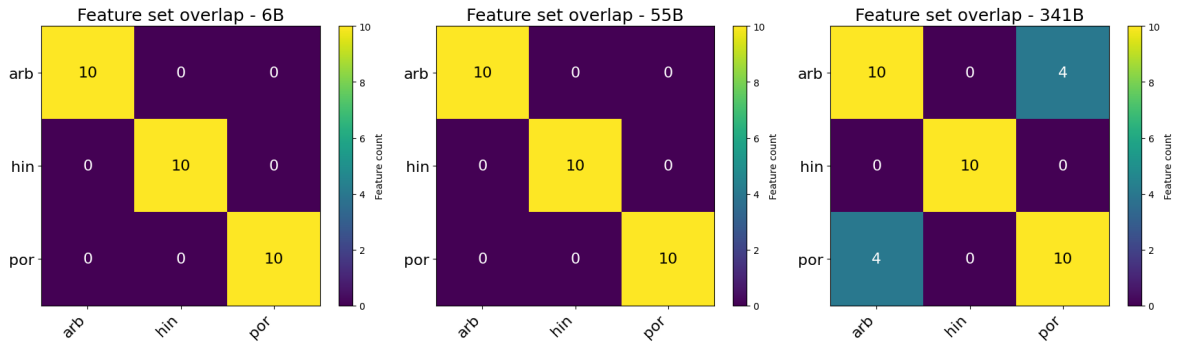
Figure 7: **RELIE of Top-10 and 100 IE features on BLiMP for Pythia-1B checkpoints {1B, 4B, 286B}.** Distinct clusters near each corner indicate checkpoint-specific features. In the Top-10 row, the 4B-286B pair and features shared across all three checkpoints dominate, whereas the 1B-286B pair have relatively fewer shared features. Additionally, the checkpoint-specific regions for 4B and 286B are noticeably denser, suggesting a richer set of unique features in these models that are trained on more data than 1B.



(a) Feature set overlap — SV-# (6B, 55B, 341B)

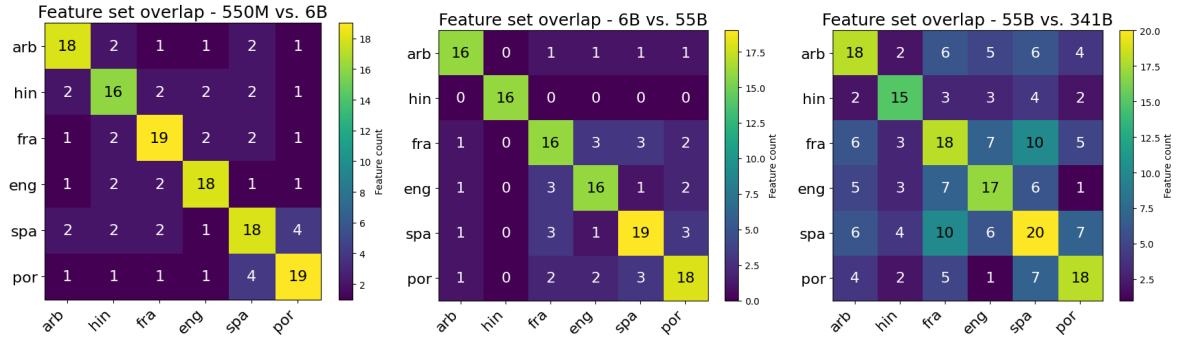


(b) Feature set overlap — SV-P (6B, 55B, 341B)

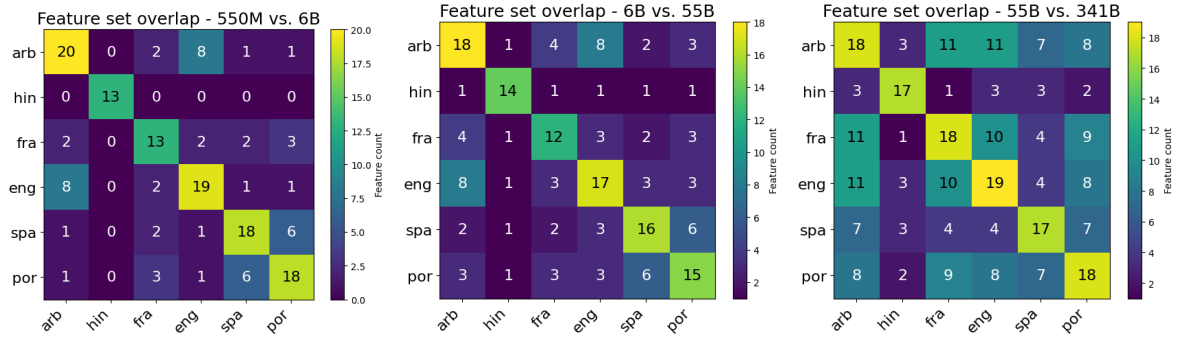


(c) Feature set overlap — SV-G (6B, 55B, 341B)

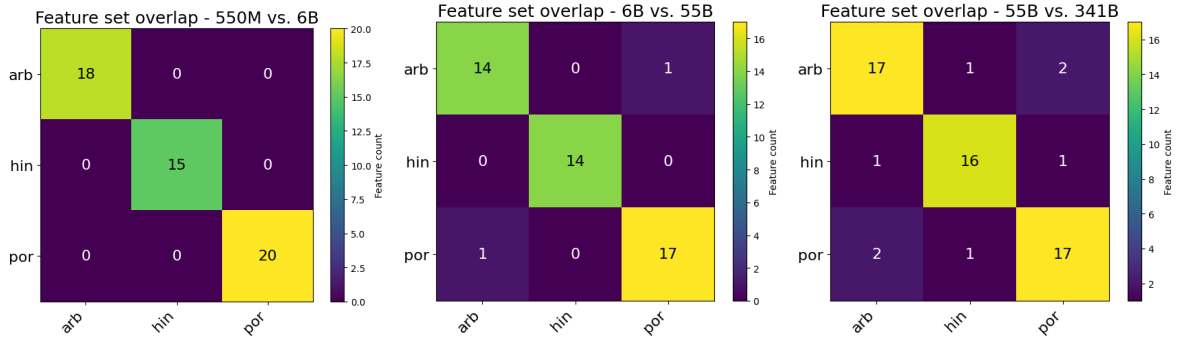
Figure 8: **Top-10 IE feature overlap in BLOOM-1B across languages and subtasks with the 3-way comparison.** Across BLOOM-1B checkpoints, feature overlap is generally higher among script-sharing languages (*e.g.*, English, French, Spanish, Portuguese) and increases across pretraining, while languages like Arabic and Hindi, which are less frequent in the training data and use different scripts, show relatively less cross-lingual feature sharing.



(a) Feature set overlap — SV-# (550M vs. 6B, 6B vs. 55B, 55B vs. 341B)



(b) Feature set overlap — SV-P (550M vs. 6B, 6B vs. 55B, 55B vs. 341B)



(c) Feature set overlap — SV-G (550M vs. 6B, 6B vs. 55B, 55B vs. 341B)

Figure 9: **Top-10 IE feature overlap (per checkpoint) in BLOOM-1B across languages and subtasks with the 2-way comparison.** The 2-way comparison shows a similar pattern to the 3-way analysis, with high feature overlap among related languages (*e.g.*, English, French, Spanish, Portuguese). Notably, in the 55B vs. 341B comparison, Arabic—despite not being Indo-European—shares more features than Hindi, suggesting better cross-lingual generalization for Arabic at later checkpoints. One explanation can be that Arabic is more prevalent in the training data (5%) than Hindi (2%).