Sparse Feature Coactivation Reveals Causal Semantic Modules in Large Language Models

Ruixuan Deng[1*] **Xiaoyang Hu**[2*] **Miles Gilberti**[3*] **Shane Storks**[3*]
Aman Taxali[3] **Mike Angstadt**[3] **Chandra Sripada**[3] **Joyce Chai**[3]

[1]Georgia Institute of Technology [2]Brown University [3]University of Michigan
rdeng62@gatech.edu, xiaoyang_hu@brown.edu,
{milgil, sstorks, ataxali, mangstad, sripada, chaijy}@umich.edu

Abstract

We identify semantically coherent, context-consistent network components in large language models (LLMs) using coactivation of sparse autoencoder (SAE) features collected from just a handful of prompts. Focusing on concept-relation prediction tasks, we show that ablating these components for concepts (e.g., countries and words) and relations (e.g., capital city and translation language) changes model outputs in predictable ways, while amplifying these components induces counterfactual responses. Notably, composing relation and concept components yields compound counterfactual outputs. Further analysis reveals that while most concept components emerge from the very first layer, more abstract relation components are concentrated in later layers. Lastly, we show that extracted components more comprehensively capture concepts and relations than individual features while maintaining specificity. Overall, our findings suggest a modular organization of knowledge and advance methods for efficient, targeted LLM manipulation.[1]

1 Introduction

Contemporary neural large language models (LLMs) exhibit remarkable proficiency in representing and reasoning over relational knowledge, motivating efforts toward mechanistic interpretation. Surprisingly, despite the immense complexity of LLMs, prior work has modeled this capability with simple linear transformations (Hernandez et al., 2024) and vector arithmetic (Merullo et al., 2024). However, these accounts are limited to layerwise interpretations, lacking fine-grained interpretation of specific features and their interaction.

Sparse autoencoders (SAEs; Huben et al., 2024) recently emerged as a powerful tool for extracting interpretable, often monosemantic features from the internal representations of LLMs. SAEs achieve this by learning to reconstruct dense, entangled activations at each neural network layer through sparse codes, where individual features are incentivized to activate rarely and selectively. Despite this progress, it remains unclear how features from different layers organize and coordinate to produce responses (e.g., how an LLM may access and compose its encoded knowledge to respond "*Beijing*" to a prompt for China's capital city). Cross-layer transcoders (Dunefsky et al., 2024) extend SAEs by learning to approximate MLP computations and have been used to construct computation graphs that trace information flow across LLM layers (Ameisen et al., 2025). However, these graphs can contain hundreds of densely connected nodes, making them difficult to interpret without manual grouping of features, motivating a more automated approach.

In this work, we employ SAE feature coactivation to construct inter-layer feature networks for individual prompts about various factual and linguistic relational reasoning tasks. By filtering out features that activate across diverse contexts, we identify semantically coherent connected components representing specific concepts and relations. These components are consistent across prompts, and manipulating them predictably alters model outputs. As exemplified in Figure 1, a prompt about China's capital activates components related to China and capital cities, eliciting the correct response of "*Beijing*" from the model. However, ablating features in the China component while amplifying those in the Nigeria component effectively steers its response to "*Abuja*", whereas ablating the capital component while amplifying the language component steers it to "*Chinese*." Furthermore, simultaneously ablating the China and capital components while amplifying the Nigeria and language components changes the model output to "*English*." These components also readily visualize the topol-

[*]Indicates equal contribution.
[1]Code available at https://github.com/shanestorks/SAE-Semantic-Modules.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 3419–3451

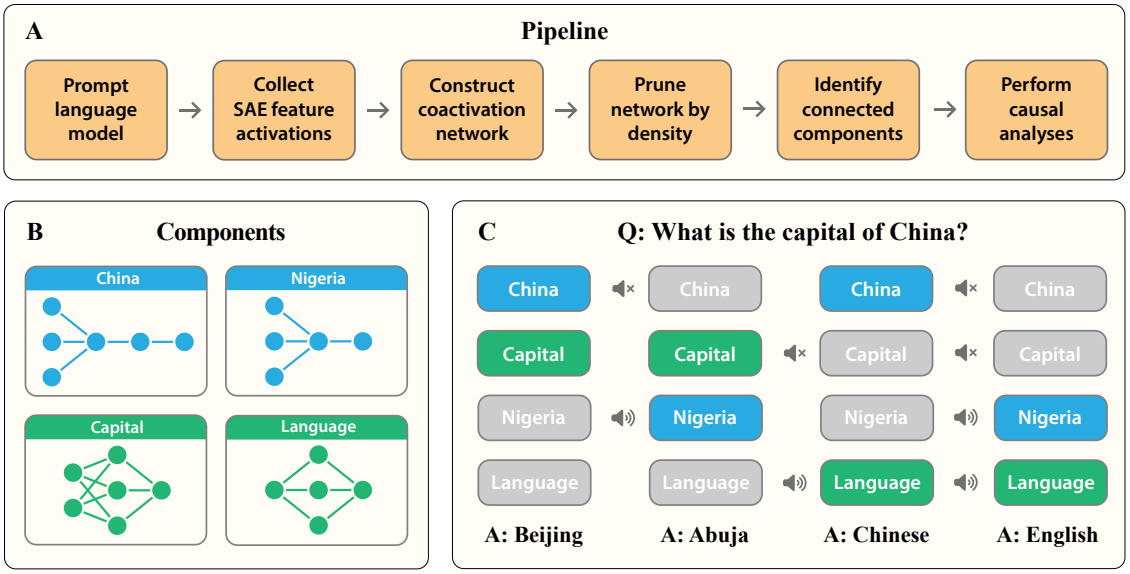


Figure 1: (A) We construct inter-layer feature networks from SAE coactivation patterns, prune high-density features, and extract task-relevant components. (B) Components are often consistent across contexts. (C) Selective ablation and amplification of components steers the model toward counterfactual outputs, overriding the original prompt.

ogy of stored concepts and relations across layers. Our experiments yield the following findings:

- Selective ablation and amplification of components consistently overrides prompted instructions, steering model outputs to predictable counterfactual answers in up to 98% of cases when steering based on individual concepts or relations, and up to 90% of cases when based on compound concept-relation pairs.
- Conceptual components representing information about specific entities and words tend to emerge from the very first network layer, while more abstract relational components (e.g., for properties of countries or translation languages) concentrate in later layers.[2]
- Our components offer a more comprehensive causal account of LLM relational reasoning than individual SAE features, and are mostly highly specific to the concepts and relations we associate them with (i.e., ablating them often preserves performance in other contexts).

Together, these results offer insights into the organization of relational knowledge in LLMs and establish sparse feature coactivation as an effective, lightweight approach for mechanistic interpretability and targeted model manipulation.

[2]This aligns with our understanding of artificial neural networks and biological systems, where higher-level abstractions emerge in later processing stages (DiCarlo et al., 2012; LeCun et al., 2015).

2 Methods

We focus our analyses on Gemma 2 2B (Team et al., 2024)[3], with some results for Gemma 2 9B in Appendix A. As shown in Figure 1, we perform several steps to construct an inter-layer feature network based on coactivation patterns, extract task-relevant components, and perform targeted causal interventions to assess their functional roles.

Activation collection. We run each input prompt through the LLM integrated with pre-trained SAEs.[4] Each SAE maps the residual stream activation at layer ℓ, $x_\ell \in \mathbb{R}^{d_{\text{model}}}$, to a sparse representation $\phi_\ell \in \mathbb{R}^{d_{\text{sae}}}$ where $d_{\text{sae}} = 16384$. This produces an activation tensor $\Phi_\ell \in \mathbb{R}^{T \times d_{\text{sae}}}$ for each layer ℓ, where T is the number of non-BOS tokens in the prompt.

Feature selection. To ensure computational tractability while preserving key information, we select a set S_ℓ of top-activated features for each layer. A feature index i is included in S_ℓ if it appears in the top $k = 5$ activations at *any* token position $t \in \{1, \ldots, T\}$:

$$S_\ell = \bigcup_{t=1}^{T} \{i \mid \Phi_\ell[t, i] \in \text{top-}k(\Phi_\ell[t, :])\}$$

[3]Accessed via `google/gemma-2-2b` in Hugging Face Hub (Wolf et al., 2020). LLM activations collected with 1 NVIDIA A100 GPU (40GB VRAM) and 12GB RAM.

[4]We use SAE Lens to apply the `width_16k/canonical` variant from the `gemma-scope-2b-pt-res-canonical` release (Lieberum et al., 2024; Bloom et al., 2024).

Graph construction. We construct a directed graph $G = (V, E)$ where each node $(\ell, i) \in V$ corresponds to a selected feature $i \in S_\ell$. Edges E connect nodes in adjacent layers according to the temporal correlation of their activation patterns across tokens in the prompt. Specifically, for feature pairs in adjacent layers $i \in S_\ell$ and $j \in S_{\ell+1}$, we compute the Pearson correlation coefficient ρ. A directed edge $e = ((\ell, i), (\ell + 1, j))$ is added to E if $\rho(\Phi_\ell[:, i], \Phi_{\ell+1}[:, j]) > \tau_{\text{corr}} = 0.9$.

Pruning and component identification. Some SAE features activate frequently across unrelated contexts, making them overly generic and hard to interpret. To reduce noise, we prune the graph using activation density scores from Neuronpedia (Lin, 2023). For each node $(\ell, i) \in V$, we retrieve its activation density $d_{\ell,i}$, i.e., the fraction of tokens in a large corpus where its corresponding feature activates. We retain only *sparse features*, i.e., those with $d_{\ell,i} \leq \tau_{\text{density}} = 0.01$,[5] creating a pruned graph G_{sparse}. We also remove any isolated nodes from G_{sparse}. We then use a straightforward BFS-based method from NetworkX[6] to identify weakly connected components within G_{sparse}.

Causal validation. To evaluate the functional significance of each component, we perform targeted interventions using `TransformerLens` (Nanda and Bloom, 2022). Specifically, we **ablate** (i.e., set to zero) the activations of SAE features in a given component during the LLM's forward pass,[7], then measure the resulting shift in the probability distribution over next-token predictions. We quantify this shift using KL divergence between the original and perturbed distributions. A component is considered causal if its ablation causes a relatively high shift to other components and leads to systematic changes in model behavior, explored in Section 3.1 through various multi-relational prediction tasks.

We further validate components in Section 3.2, where we attempt to **steer** model behavior. Specifically, we ablate components associated with prompted information, then **amplify** one or more other components unrelated to the prompt, i.e., raise the activation for each SAE feature in the component by a proportion of its maximum activation observed during activation collection. We refer to this proportion as a **steering strength** α. Steering is *successful* when it causes the model to generate a counterfactual response consistent with amplified component(s) rather than components associated with prompted information.

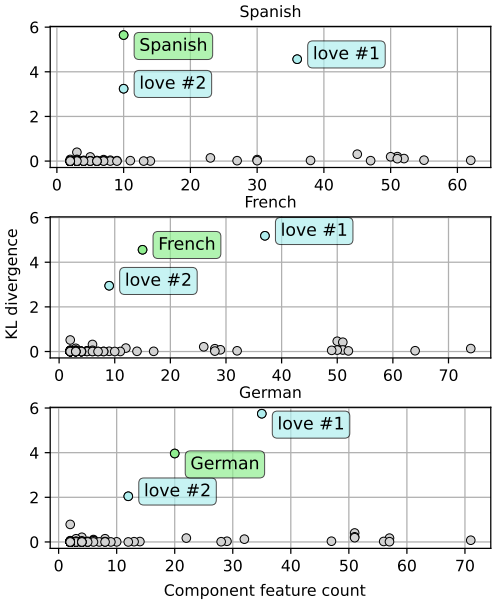


Figure 2: We extract components from LLM queries to translate *love* into Spanish, French, and German. Each component is plotted with its feature count on the x-axis and the total KL divergence between pre- and post-ablation output token distributions on the y-axis. For these and most other concept-relation pairs, only a few components exert significant causal effects. For *love*, the top three components for each language consistently correspond to either the queried word or language.

3 Experimental Results

We consider 3 multi-relational prediction tasks where the model is queried to generate a response (e.g., *Beijing*) that requires composing a given **concept** (e.g., *China*) and **relation** (e.g., *capital city*):[8]

1. **Country facts:** capital city, currency, and language (relations) of 10 countries (concepts)
2. **Word translation:** translation of 11 English words (concepts) into Spanish, French, and German (relations)
3. **Verb transformation:** synonym, antonym, past tense, capitalization, and first letter (relations) of 8 English verbs (concepts)

Gemma 2 2B achieves 100% accuracy on each task. To collect model activations, we used simple prompt templates with in-context examples, e.g.,

[5]This follows Neuronpedia's standard, which classifies features below this density as sparse and interpretable.

[6]https://networkx.org/

[7]When manipulating SAE features like this, we replace the LLM's internal activations at each layer with the corresponding SAE decoder's output after features are manipulated, then proceed with the forward pass.

[8]Latter 2 tasks adapted in part from data used in Todd et al. (2024). See Appendix B for full specifications of tasks.

for each country-capital pair, we used: "*The capital city of Peru is Lima. The capital city of South Korea is Seoul. The capital city of Saudi Arabia is Riyadh. The capital city of* {*country*} *is*". Similar templates were used across tasks (Appendix C).

The remainder of this section is organized as follows. We first identify semantic components corresponding to each concept and relation (Section 3.1). We elicit counterfactual model outputs by ablating and amplifying these components individually and in combination (Section 3.2). Lastly, we perform several analyses to examine the organization and impact of component features across model layers (Section 3.3), whether our approach enables more reliable manipulation of concepts and relations than individual SAE features (Section 3.4), and whether ablating components for one task degrades model performance on other tasks (Section 3.5).

3.1 Component Identification

For each concept-relation pair, we obtained sparse components using the methods from Section 2. As shown in Figure 2 for translation,[9] 2-3 of the components typically exerted a markedly higher causal effect on the model output than the others.

Semantic coherence. Given several impactful components, we then identify *semantically coherent* representative components for each concept and relation, i.e., components that yield predictable counterfactual outputs when manipulated (like in Figure 1). We explored a few approaches. First, we inspected their associated feature descriptions from Neuronpedia.[10] In many cases, the features within components had thematically coherent descriptions referring to common concepts or relations, e.g., most country fact task components (Appendix E).

However, these descriptions were often unreliable. For instance, none of the components obtained for prompts about Spain mentioned Spain in their feature descriptions, and some components for the word *beautiful* included features about Scotland. Additionally, some components for translation languages were described to be about programming. This suggests that even SAE features suffer from some polysemanticity. Notably, this shows that it is impossible to consistently extract comparable components by simple keyword search over SAE feature descriptions, necessitating methods like ours to group features based on task-specific activations.

To address this in the country facts task, we ablated components and observed the resulting changes in the model's top predicted tokens, selecting those that yielded the most compositional changes. As an example, the left 3 columns of Table 1 present results of ablating selected components obtained from prompts for the capital of China and Nigeria. Promisingly, when ablating country components, the model's top predicted tokens shifted to the capitals of other countries. When ablating fact components, the model assigned higher probabilities to country names.

For the remaining tasks, this approach was unsuccessful, possibly due to more complex composition operations at play than can be captured by a single component for a concept or relation. In these tasks, we instead represented each concept and relation by the union of multiple seemingly relevant causally impactful components. As shown in the other columns of Table 1, ablating these combined components for words/verbs in these tasks similarly caused the model to output other words in the prompted language of Spanish, or other capitalized words. Meanwhile, ablating relation components caused the model to repeat the input word.

Context consistency. Most concept and relation components were remarkably consistent across different contexts, e.g., the top China components obtained from capital and currency prompts were identical, and the language components from China and Nigeria prompts differed only by a few edges. Therefore, for country facts and translation, we define each concept component as the intersection of all the components for that concept across relations; similarly, each relation component is defined as the intersection of all the components for that relation across concepts. For verb transformations, while many components appeared context-insensitive, using the specific components identified for each prompted concept-relation pair maximized steering performance. Figure 3 visualizes the resulting China and language components, and more examples are provided in Appendix F.

3.2 Component Steering

Having identified distinct components for each concept and relation, we next investigate whether they can be used to steer model outputs individually

[9] Additional examples in Appendix D.

[10] These descriptions from Neuronpedia were generated by prompting GPT-4o mini (OpenAI et al., 2024) to summarize the texts that activate each feature over a large corpus.

Capital, $\frac{\text{China}}{\text{Nigeria}}$						**Spanish,** $\frac{\text{love}}{\text{red}}$						**Capitalize,** $\frac{\text{break}}{\text{like}}$					
Original		*Ctry. Abl.*		*Fact Abl.*		*Original*		*Word Abl.*		*Lang. Abl.*		*Original*		*Verb Abl.*		*Trans. Abl.*	
Beijing	97.	Madrid	39.	the	12.	Amor	41.	"	5.9	Love	42.	Break	22.	Capital	10.	break	20.
Be	.38	Warsaw	10.	China	6.7	amor	19.	El	4.9	love	14.	break	13.	The	7.6	Break	7.1
Peking	.35	Rome	9.5	Beijing	6.4	Amo	7.0	La	4.4	I	13.	BREAK	8.3		4.2		5.1
Shanghai	.23	Paris	6.1	Shanghai	5.0	amo	5.1	Malo	3.2	"	5.3	Capital	5.7	I	3.6	capital	4.3
Xi	.11	Berlin	5.0	a	2.7	"	2.9	la	2.5	LOVE	2.7	The	5.0	Yes	2.3	The	3.3
Abuja	85.	New	7.7	Lagos	17.	rojo	38.	La	3.5	Red	27.	Like	21.	Capital	5.6	like	12.
Lagos	11.	Islamabad	7.3	the	12.	Rojo	34.		3.4	red	15.	like	12.	The	4.7		6.1
...	.38	Kathmandu	6.0	Nigeria	7.1	rojo	2.1	"	3.2	"	8.1	Capital	8.6	Yes	4.5	capital	5.6
Nigeria	.29	Delhi	5.7	called	5.1	Ro	2.0	El	2.8	The	5.1	LIKE	4.3	No	4.1	the	4.6
...	.27	Tehran	4.9	Abuja	3.4	Roja	1.9	Bol	2.4	...	4.1	The	4.0		3.3	Like	3.8

Table 1: Top 5 output tokens and their likelihoods for various concept-relation pairs, before and after ablating concept or relation components. Underscore prefixes of tokens omitted for space. More examples in Appendix H.1.

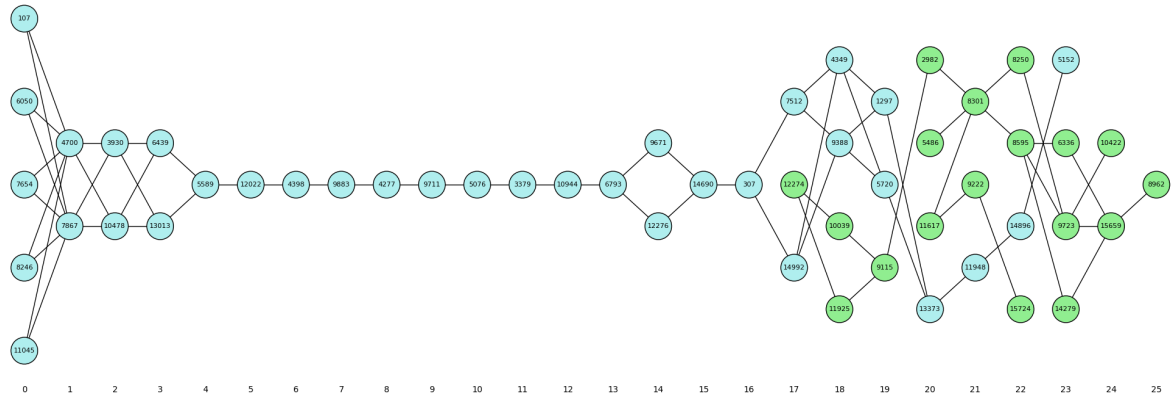


Figure 3: Components representing China (blue) and country language (green), visualized by network layer.

and in combination.[11] To test whether components generalize across different contexts, we applied a zero-shot prompt template different from the one used to collect the initial activations, e.g., for translation: *"Q: What is the* {*Spanish/French/German*} word for '{*word*}'*? Answer directly (two words max). A:"*. We used comparable prompts for other tasks, as listed in Appendix C.

Table 2 summarizes the results by task and relation, while Appendix I further breaks down performance by concepts, and Appendix J discusses common and interesting steering failure cases.

Concept steering. By ablating an **in-prompt concept** component and amplifying a **target concept** component, we successfully directed the LLM to respond to questions about relations for the in-prompt concept with *counterfactual answers* (i.e., from applying the relation to the target concept which is not actually queried in the prompt) 48-96% of the time on average (Table 2). For example, as shown in Table 3, the model consistently disregarded prompts' references to China, *love*, and *understand*, instead outputting counterfactual answers "*English*" (Nigeria's language), "*rouge*" (*red* in French), and "broke" (*break*'s past tense). Even when incorrect answers were among top tokens, they were often relevant to the in-prompt relation and target concept, e.g., the model ranked "*Yoruba*," "*Ha*[*usa*]," and "*Igbo*" (other commonly spoken languages in Nigeria) just behind the correct answer of "*English*," while it ranked a French word for *blood* just behind words for the target concept of *red*. This confirms that our identified concept components encode concept-specific information that causally determines model outputs.

We observed the most failures in the verb transformations task. Notably, some verbs had a much lower concept steering success rate than others, e.g., *possess* (6%) and *sink* (20%). These same verbs caused failures in both directions: steering to them (as target verbs) and steering away from them (as in-prompt verbs). This suggests that our method did not yield stable components for these particular verbs. Interestingly, we saw that when steering to *sink* from *break* and *hide* under the past tense

[11] See Appendix G for details on steering strength tuning.

		Country Facts				*Word Translation*				*Verb Transformation*					
C	**R**	**Cap.**	**Curr.**	**Lang.**	**Avg.**	**Spanish**	**French**	**German**	**Avg.**	**Syn.**	**Ant.**	**Past T.**	**Cap.**	**1st L.**	**Avg.**
✓	✗	.98	.93	.96	.96	.74	.74	.78	.75	.61	.45	.61	.18	.57	.48
✗	✓	1.0	.85	.95	.93	1.0	.95	1.0	.98	.00	.00	.00	.31	.81	.23
✓	✓	1.0	.77	.93	.90	.57	.69	.67	.64	.02	.00	.00	.16	.79	.19

Table 2: Steering success rates (SR) for all tasks by relation and averaged. Row 1: SR of steering from one concept to another (C✓) while preserving relation (R✗). Row 2: SR of steering to each relation from others (R✓) while preserving concept (C✗). Row 3: SR of steering to concept-relation pairs from others by target relation (C✓, R✓).

Language, CN						**French, love**						**Past Tense, understand**					
·, NG		*Capital, ·*		*Currency, ·*		*·, red*		*Spanish, ·*		*German, ·*		*·, break*		*Capitalize, ·*		*1st Letter, ·*	
_English	71.	_Beijing	96.	_Yuan	12.	_rouge	44.	_amor	41.	_Liebe	68.	_broke	28.	_Understand	6.7	_U	39.
_Yoruba	5.6	_Be	2.0	_Ren	8.4	_Rouge	30.	_Amor	13.	_die	2.4	_Broke	21.	_Know	6.4	_understood	9.3
_Ha	4.8	_Peking	.93	_China	6.1	_Sang	2.5	_el	6.9	_„	2.1	_BRO	8.5	_Past	5.9	_Understand	8.5
_Nigeria	4.4	_BE	.53	_Chinese	5.9	_Rou	1.8	_"	6.1	_lieben	1.7	_Bre	6.3	_Answer	5.7	_u	3.6
_Igbo	3.4	Beijing	.37	_RMB	5.9	_	1.2	_A	3.1	_Ich	1.6	_brake	2.6	_	4.5	_United	3.2

Table 3: Top 5 output tokens and likelihoods for prompts about China's language (columns 1–3), *love*'s French translation (columns 4–6), and *break*'s past tense (columns 7–9), after ablating an in-prompt concept or relation component (row 1 headings) and amplifying a target concept or relation component (row 2 headings). Highlighted tokens are or may begin correct answers for steered components. More examples in Appendix H.2.

relation, the model respectively generated "*sought*" and "*slid*", possibly suggesting that components for *sink* may instead represent words that begin with *s*. Meanwhile, the capitalize relation had the lowest success rate; we observed that the model usually output the target word, but not capitalized. For *hide*, the model instead output the target verb in all capital letters. This may suggest that components for specific verbs can override prompt instructions when used for steering, perhaps indicating a lower degree of composability than other tasks.

Relation steering. By ablating an *in-prompt relation* while amplifying a *target relation*, we also successfully directed the model to respond to queries about the in-prompt relation as though they concerned the target relation 23-98% of the time (Table 2). As shown in Table 3, the model disregarded prompts' references to country language, French translation, and past tense transformation, and produced correct counterfactual answers for other relations within each task, e.g., answering prompts to translate *love* into French with Spanish and German words for *love*. Even when incorrect answers were among top tokens, they were often relevant, e.g., words in the target language, or "*Know*" for the capitalized form of *understand* (a synonym for the correct answer). In some cases, correct answers competed with answers related to the prompt context, suggesting that some in-prompt information remains activated. For example, "*Chinese*" appeared among the top tokens when steering to the currency relation even after the in-prompt component for the language relation was ablated. Similarly, "*understood*" appeared among top tokens when steering to first letter after the in-prompt component for past tense was ablated.

Unsurprisingly, the most failures again occurred in this task, specifically for the synonym, antonym, and past tense transformations. For the former two, this may be expected. Unlike the capital city or currency of a country, synonym and antonym do not have objective answers, and actually depend on a variety of finer-grained dimensions, e.g., word sense and part of speech. As observed in previous visualized token distributions after ablation and steering, the country fact and word translation components have clear roles in model outputs: to promote tokens related to specific countries or words, or that are relevant to a target fact (languages, currencies, or cities) or in a target language. However, a mechanism for synonym or antonym relations should not be expected to work like this, as many tokens could be a synonym or antonym, and reasonable answers thus depend more heavily on context. While past tense forms of verbs are more consistent, this relation is much broader and less specific than relations in other tasks, as every verb has a past tense form, and thus an LLM learning a specialized module to promote past tense inflections may be

Currency, China				German, love				Antonym, break			
Capital, Nigeria		*Language, Nigeria*		*Spanish, red*		*French, red*		*Synonym, like*		*1st Letter, like*	
_Abuja	70.	_English	96.	_rojo	53.	_rouge	45.	_love	56.	_like	31.
_Lagos	25.	_Yoruba	1.1	_el	7.2	_le	6.7	_loved	8.5	_Like	20.
_	3.3	_French	.74	_roja	5.6	_Rouge	4.2	_few	3.8	_L	6.3
_Nigeria	.34	_Spanish	.60	_"	3.1	_	3.7	_loves	3.6	_ant	3.2
_...	.32	_Igbo	.54	_El	2.2	_la	3.4	_lot	3.4	_Ant	2.5

Table 4: Top 5 output tokens and their likelihoods for prompts about China's currency (columns 1–2), *love*'s German translation (columns 3–4), and *break*'s antonym (columns 5–6), after ablating both in-prompt components and amplifying components for a target concept-relation pair. More examples in Appendix H.3.

inefficient. Such relations are thus likely better captured in attention heads, which aligns with Todd et al. (2024)'s finding that antonym and past tense are modeled well by function vectors.

Composite steering. We conducted composite steering experiments where both concept and relation components were manipulated at once. By ablating both the in-prompt concept and in-prompt relation components while amplifying the target concept and target relation components, it was possible to steer the model to ignore both the in-prompt concept and in-prompt relation and answer about a different concept-relation pair 19-90% of the time (Table 2). Table 4 provides specific examples of composite steering success. As shown in the first 2 columns, when we ablated both the China and capital components while amplifying the Nigeria and currency components, the model answered "*Naira*" despite being asked about China's capital. In the middle 2 columns, we see similar success in steering the model's outputs to Spanish and French words for *red*, despite being prompted for the German translation of *love*. In the last 2 columns, when prompted for an antonym of *break*, we successfully steer the model to output a synonym of *like*, while steering it to output the first letter of *like* was less successful, with the correct answer only ranked third after "*like*" and "*Like*". We also observed interesting failure cases where in-prompt information intermingled with target information, e.g., steering from the antonym of *break* to *include* capitalized yielded "*Exclude*," a capitalized antonym of *include*. Nonetheless, these findings demonstrate significant composability of many components.

3.3 Component Organization

Having established the causal role and composability of concept and relation components, we next analyze their distribution across network layers and the relative importance of nodes within these components. To quantify the causal importance of an individual node in a concept component, we compute the average post-ablation KL divergence from the original output distribution across all relations. For nodes in relation components, we compute the same average across all concepts. Figure 4 displays results for the country facts task, with more examples in Appendix K.

Concept and relation components show distinct distribution patterns across network layers. 8 of 10 country components begin in the first layer, as do all word/verb components in other tasks. Some concept components span nearly all layers (e.g., for China and *love*), while others concentrate in early to middle layers (e.g., for Nigeria and *red*). In contrast, relation components concentrate in later layers, with most spanning only the last quarter to half of the network's layers (e.g., for country facts and target translation languages), and some spanning further across the network while having greater densities of nodes in later layers (e.g., for verb transformations). This may suggest that representations of more concrete entities are established earlier in processing, while more abstract relations emerge later. Concept component nodes exhibit various trends in KL divergence across layers. Meanwhile, relation components for country facts, past tense, and first letter have stronger causal impact in later layers, and the opposite trend holds for translation languages and capitalization.

3.4 Single Feature Steering Comparison

We hypothesize that our approach of steering based on a group of features is advantageous over steering based on an individual feature (as done in some prior works, e.g., Anthropic, 2023; Templeton et al., 2024). To validate this, we compared its efficacy against a simpler baseline: intervening on only the single most causally impactful feature. Specifically,

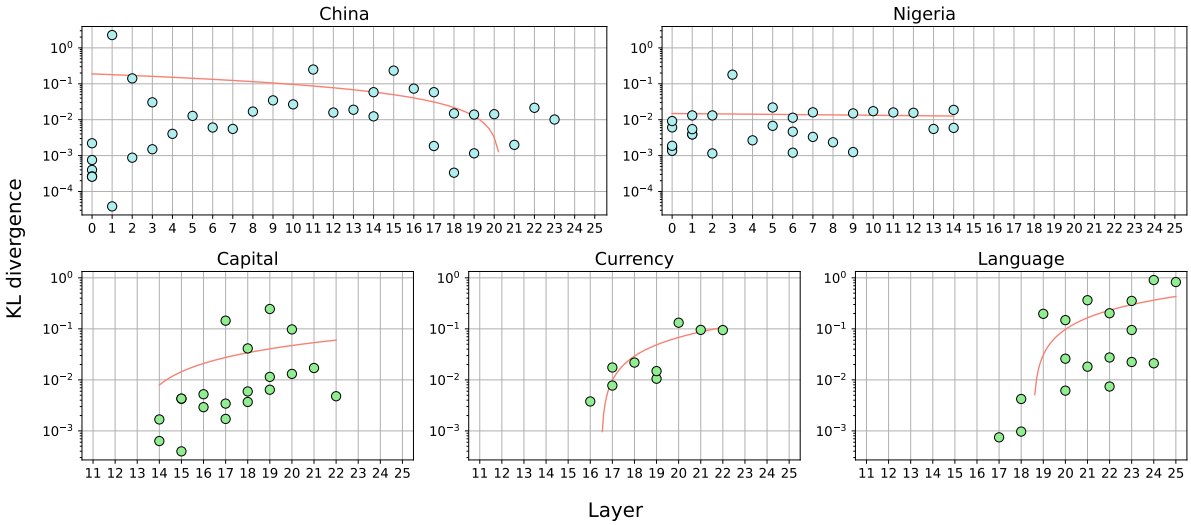


Figure 4: KL divergence between pre- and post-ablation output token distributions for each node in the China, Nigeria, and country fact components, plotted by layer. Linear regression lines plotted in red.

we identified this top feature for each concept and relation as the one whose individual ablation produced the highest KL divergence, then in line with earlier results, used it for ablation and steering.

For the country facts task, the baseline respectively achieved average success rates of 83%, 83%, and 75% for concept, relation, and composite steering. These results fall notably short of the success rates achieved using full components (respectively 96%, 93%, and 90%). This suggests that concepts are represented distributively across populations of features, and our method's success is largely attributable to identifying and manipulating these entire functional groups.

3.5 Specificity Evaluation

Lastly, we explore the degree to which ablating our identified components impacts performance on other tasks, an important question for the specificity of mechanisms and practicality of manipulating models based on them. Table 5 presents the model's average accuracy for each task when our identified components are ablated. When ablating components from the country facts task, 100% accuracy (or near it) is maintained in most cases, even when ablating components for some country-fact pairs and prompting the model for other ones. Accuracy on translation degrades slightly, possibly due to the intersection of some country- and language-specific information. In ablating word translation components, accuracy remains high on other tasks, but degrades slightly on translation itself. This may be expected, given that the relations in this task are of highly similar nature, and even overlap by a small number of features. In ablating verb transformation components, we see moderately degraded accuracy for all tasks to as low as 62-73%. This suggests that the components for this task are less specific than other tasks, in line with our relatively low steering performance with them. Nonetheless, this demonstrates that extracted components are generally fairly specific to the functions we associate them with, promising for manipulating deployed models with them.

Ablated Task	Country Facts Acc.	Word Transl. Acc.	Verb Transf. Acc.
Country Facts	1.00	0.93	1.00
Word Transl.	1.00	0.83	1.00
Verb Transf.	0.73	0.64	0.63

Table 5: Prediction accuracy over concept-relation pairs in each task when components from the same or another task are ablated. If the same task, cases where prompted concept-relation pair overlaps with ablated concept-relation pair are omitted from evaluation.

4 Related Work

Circuits and sparse feature dictionaries. Prior work has applied various methods to interpret large neural networks as compositions of subnetworks that perform specific functions (Olah et al., 2020; Elhage et al., 2021; Sharkey et al., 2025). Specialized circuits have been discovered for various functions of LLMs, including in-context learning (Olsson et al., 2022), numerical comparison (Hanna et al., 2023), indirect object recognition (Wang

et al., 2023), and broader natural language fluency (AlKhamissi et al., 2025). More recent work has developed automated circuit discovery methods to improve scalability (Goldowsky-Dill et al., 2023; Conmy et al., 2023; Hsu et al., 2025; Bhaskar et al., 2024). However, these approaches can be computationally expensive and yield circuits that are difficult to interpret. As large neural networks like LLMs are thought to encode sparse features across polysemantic neurons (Elhage et al., 2022), dictionary learning techniques like SAEs promise greater interpretability by extracting monosemantic feature directions from them (Bricken et al., 2023; Huben et al., 2024). This has allowed automated circuit discovery approaches to operate on these features instead of neurons (Marks et al., 2025), but high computational costs remain a barrier. Like our work, Li et al. (2025) used SAE feature coactivation patterns to analyze their geometry, finding spatial clustering of related concepts. Standard SAEs can also suffer from inconsistent feature quality, poor reconstruction, and weak functional alignment, prompting various architectural and training improvements (Braun et al., 2024; Rajamanoharan et al., 2024a,b). Transcoders (Dunefsky et al., 2024) extend SAEs by learning to approximate layerwise computations. Cross-layer transcoders comprise the Anthropic circuit tracing tool (Ameisen et al., 2025), which visualizes LLM responses with sparse but still difficult-to-interpret computation graphs. Building on these efforts, we follow Li et al. to leverage SAE feature coactivation, applying node pruning based on activation density to automatically discover semantically coherent, context-consistent components that can influence LLMs' relational reasoning outputs individually and in combination. Our approach offers an efficient framework for analyzing and controlling LLM behavior without exhaustive circuit tracing.

Knowledge organization in LLMs. Orthogonally, prior work has used various methods to study the organization of lexical and factual knowledge in LLMs. Factual knowledge is believed to reside in the feedforward layers of transformer-based LLMs. Geva et al. (2021) characterized these layers as key-value memories mapping textual patterns to vocabulary distributions. Dai et al. (2022) identified "knowledge neurons" whose activations correlate with specific facts. These insights informed approaches for editing factual knowledge stored in LLMs (De Cao et al., 2021; Mitchell et al., 2022; Meng et al., 2022). Geva et al. (2023) described the recall of factual associations as a three-step process of subject enrichment, relation propagation, and attribute extraction, while Hernandez et al. (2024) demonstrated that relation decoding in transformers can be approximated by simple linear transformations. Recent work on "knowledge circuits" has begun tracing causal pathways underlying factual recall (Yao et al., 2024; Ou et al., 2025).

Highly relevant to this work, recent work has explored the composition of knowledge in LLMs. Merullo et al. (2024) demonstrated that LLMs implement word2vec-style (Mikolov et al., 2013) vector arithmetic through their feedforward layers to solve some relational tasks. Bayat et al. (2025) leveraged active SAE features in contrastive paired prompts to steer LLM behaviors, showing evidence of composability of behaviors. In line with our word translation results, Dumas et al. (2025) used activation patching to reveal language-agnostic concept representations in LLMs, such that the translation language and concept can be steered independently. Our work extends these efforts by identifying finer-grained modular groups of feature directions that represent human-interpretable concepts and relations and similarly compose to form responses, possibly from a single relational prompt.

5 Conclusion

By tracing sparse feature coactivation patterns across layers, we uncover semantically coherent, context-consistent components encoding both concrete entities and abstract relations. We find that ablating or amplifying these components reliably alters model behavior. Moreover, composing concept and relation components induces compound counterfactual responses. The layer-wise distribution of components reveals a hierarchical organization: entity features emerge in early layers, while relational features cluster in deeper layers. Overall, our method provides a lightweight, interpretable framework for analyzing and steering LLM behavior without full-scale circuit tracing. This work advances scalable interpretation and control of LLM behavior by leveraging pretrained SAEs to efficiently identify functional components. Given the adoption of LLMs by humans on a greater variety of tasks, such capabilities are vital for sustaining a human-machine enterprise that is not only reliable and safe, but also transparent, aligned with human goals, and resilient to misuse and misinterpretation.

Limitations

We acknowledge that our study is limited in the following ways:

Limited datasets. First, due to computational constraints, our investigation was constrained to only 3 multi-relational reasoning tasks with small numbers of examples. When it comes to the task space, we attempted to add some additional tasks around Spanish verb conjugation, factual information about Nobel Prize winners, and taxonomical information about animals,[12] but Gemma 2 2B could not perform these tasks accurately off the shelf (for the former, this is likely due to Gemma not being trained on much Spanish data), thus we would not expect to identify semantic modules for these tasks. That said, the tasks we explored in this paper are diverse, representing factual and lexicosemantic knowledge, and similar forms of these tasks are also considered in prior relevant works (Merullo et al., 2024; Todd et al., 2024; Hernandez et al., 2024). As such, we expect that our analytical framework generalizes to other domains.

When it comes to the datasets' sizes, scaling our method up to large numbers of examples causes some complications which we intentionally chose to avoid. First, increasing the dataset sizes causes a combinatorial explosion for the reporting of steering results, which currently involve steering from each concept/relation/concept-relation pair to every other concept/relation/concept-relation pair. Consequently, with a larger dataset, any steering results reported would have to be based on subsets of such combinations, e.g., specific interchange interventions as done in the RAVEL dataset (Wu et al., 2024). However, this has a tradeoff of providing a less clear view into the composition of the semantic modules identified in our work.

This is also nontrivial due to the requirement of manually selecting components based on various heuristics outlined in Section 3. While we believe it may be possible to automate component selection in future work (as described below), this would still take a significant amount of time and computational power. Nevertheless, we believe the examples in our datasets comprise sufficiently representative samples of the domains, e.g., countries around the world, animals from various classes, orders, and families, words of various syntactic categories, and verbs with both common and irregular past tense transformations.

Limited selection of interpretability methods and models. Computational constraints also restricted our analysis to standard JumpReLU sparse autoencoders (Rajamanoharan et al., 2024b), which can suffer from inconsistent feature quality, poor reconstruction, and weak functional alignment. These limitations have motivated various improved architectures (Braun et al., 2024; Rajamanoharan et al., 2024a,b; Dunefsky et al., 2024), which may yield even more refined functional module identification under our approach. As a proof-of-concept, we extended our translation analysis to Gemma 2 2B transcoders (Dunefsky et al., 2024), successfully rediscovering some causal features for translation languages that Hanna et al. manually select, and achieving about 27% steering success rate with our extracted components.[13] This provides evidence that this approach can work with broader interpretability methods than SAEs.

Related to this, most of our results were limited to one model (Gemma 2 2B), except for the supplementary results on the country facts task with Gemma 2 9B in Appendix A. Our choice of LLMs was constrained to those with pretrained SAEs available via Neuronpedia. Further, we did not use smaller models like GPT-2, as they demonstrated a weaker grasp of the factual concepts under investigation (e.g., confusing Lagos, Nigeria's largest city, with its capital) and an inability to follow in-context instructions to shorten their answers. As we continue to improve this work, we hope to expand the results to more models within the bounds of our computational constraints.

Reliance on manual intervention. While our method of extracting components based on sparse feature coactivation is fully automated, as detailed in Section 3.1, we manually selected and/or combined components to represent each concept and relation using heuristics dependent on the task. This may give an impression that significant manual intervention is required, compromising the utility of our method. However, the focus of our work was primarily to automate the former step of component extraction, and given that, we found it relatively simple to identify semantically coherent and impactful components (the latter step). As we expanded the work to more tasks, we found that

[12] More discussion about animal taxonomy results in Appendix L.

[13] See Appendix M for more details.

different choices maximized performance depending on the nature of the task, resulting in minor variations to our method.

That said, after conducting this work, we believe these heuristics can be replaced with a simple algorithm in future work. First, as our approach already does, we can apply our sparse feature coactivation method to identify several components for each concept-relation prompt, and sort them by their KL divergence of the LLM's next token probability before and after ablation. Second, we can represent each unique concept and relation by identifying the most impactful component(s) which overlap among all prompts for that concept or relation, then taking the intersection of them. Another advantage of such an approach is that it removes dependence on feature descriptions, e.g., those we used from Neuronpedia.

Focus on sparsely activating features. Lastly, our analysis focused exclusively on features with low activation density. Intriguingly, preliminary experiments revealed that ablating high-density features produces syntactically and semantically incoherent outputs, suggesting these features serve critical yet unexplored computational functions that merit further investigation.

Acknowledgments

We thank our anonymous reviewers for their thoughtful and constructive feedback. This research was supported in part through computational resources and services provided by Advanced Research Computing at the University of Michigan, Ann Arbor. Google Gemini 3 (gemini.google.com) and Anthropic Claude (claude.ai) were used for minor language edits and generation of boilerplate and visualization code.

References

Badr AlKhamissi, Greta Tuckute, Antoine Bosselut, and Martin Schrimpf. 2025. 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.

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, and 8 others. 2025. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*.

Anthropic. 2023. Decomposing language models into understandable components. Blog post, Anthropic Research.

Reza Bayat, Ali Rahimi-Kalahroudi, Mohammad Pezeshki, Sarath Chandar, and Pascal Vincent. 2025. Steering large language model activations in sparse spaces. In *Second Conference on Language Modeling*.

Adithya Bhaskar, Alexander Wettig, Dan Friedman, and Danqi Chen. 2024. Finding transformer circuits with edge pruning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

Dan Braun, Jordan Taylor, Nicholas Goldowsky-Dill, and Lee Sharkey. 2024. Identifying functionally important features with end-to-end sparse dictionary learning. *Advances in Neural Information Processing Systems*, 37:107286–107325.

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.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

James J DiCarlo, Davide Zoccolan, and Nicole C Rust. 2012. How does the brain solve visual object recognition? *Neuron*, 73(3):415–434.

Clément Dumas, Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2025. Separating tongue from thought: Activation patching reveals language-agnostic concept representations in transformers. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31822–31841, Vienna, Austria. Association for Computational Linguistics.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. 2024. Transcoders find interpretable LLM feature circuits. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2022/toy_model/index.html.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. A mathematical framework for transformer circuits. *Transformer Circuits Thread*. Https://transformer-circuits.pub/2021/framework/index.html.

Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 2023. Dissecting recall of factual associations in auto-regressive language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12216–12235.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. 2023. Localizing model behavior with path patching. *Preprint*, arXiv:2304.05969.

Michael Hanna, Ollie Liu, and Alexandre Variengien. 2023. How does gpt-2 compute greater-than?: Interpreting mathematical abilities in a pre-trained language model. *Advances in Neural Information Processing Systems*, 36:76033–76060.

Michael Hanna, Mateusz Piotrowski, Jack Lindsey, and Emmanuel Ameisen. 2025. circuit-tracer. `https://github.com/safety-research/circuit-tracer`. The first two authors contributed equally and are listed alphabetically.

Evan Hernandez, Arnab Sen Sharma, Tal Haklay, Kevin Meng, Martin Wattenberg, Jacob Andreas, Yonatan Belinkov, and David Bau. 2024. Linearity of relation decoding in transformer language models. In *The Twelfth International Conference on Learning Representations*.

Aliyah R Hsu, Georgia Zhou, Yeshwanth Cherapanamjeri, Yaxuan Huang, Anobel Odisho, Peter R Carroll, and Bin Yu. 2025. Efficient automated circuit discovery in transformers using contextual decomposition. In *The Thirteenth International Conference on Learning Representations*.

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

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature*, 521(7553):436–444.

Yuxiao Li, Eric J. Michaud, David D. Baek, Joshua Engels, Xiaoqing Sun, and Max Tegmark. 2025. The geometry of concepts: Sparse autoencoder feature structure. *Entropy*, 27(4).

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.

Johnny Lin. 2023. Neuronpedia: Interactive reference and tooling for analyzing neural networks. Software available from neuronpedia.org.

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*.

Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*.

Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. 2024. Language models implement simple Word2Vec-style vector arithmetic. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5030–5047, Mexico City, Mexico. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2022. Fast model editing at scale. In *International Conference on Learning Representations*.

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

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*. Https://distill.pub/2020/circuits/zoom-in.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, and 1 others. 2022. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*.

OpenAI, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, and 400 others. 2024. GPT-4o system card. *Preprint*, arXiv:2410.21276.

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 pretraining. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19889–19913, Vienna, Austria. Association for Computational Linguistics.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. 2024a. Improving dictionary learning with gated sparse autoencoders. *Preprint*, arXiv:2404.16014.

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

Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeffrey Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Isaac Bloom, Stella Biderman, Adrià Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Mary Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, William Saunders, and 10 others. 2025. Open problems in mechanistic interpretability. *Transactions on Machine Learning Research*. Survey Certification.

Yushi Sun, Hao Xin, Kai Sun, Yifan Ethan Xu, Xiao Yang, Xin Luna Dong, Nan Tang, and Lei Chen. 2024. Are large language models a good replacement of taxonomies? *Proc. VLDB Endow.*, 17(11):2919–2932.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, and 3 others. 2024. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html. Blog post, Transformer Circuits Thread.

Eric Todd, Millicent Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. 2024. Function vectors in large language models. In *The Twelfth International Conference on Learning Representations*.

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*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhengxuan Wu, Atticus Geiger, Aryaman Arora, Jing Huang, Zheng Wang, Noah Goodman, Christopher Manning, and Christopher Potts. 2024. pyvene: A library for understanding and improving PyTorch models via interventions. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*, pages 158–165, Mexico City, Mexico. Association for Computational Linguistics.

Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and Huajun Chen. 2024. Knowledge circuits in pretrained transformers. In *Advances in Neural Information Processing Systems*, volume 37, pages 118571–118602. Curran Associates, Inc.

A Gemma 2 9B Results

We replicated country fact task ablation and steering using Gemma 2 9B. Results closely mirrored those observed with Gemma 2 2B. The model showed context consistency, with similar country components across relations (Figure 5) and similar relation components across countries (Figure 6). High success rates were achieved for country (93%), relation (97%), and composite (92%) steering (see Tables 6–8).

B Full Task Details

In this appendix, we provide the full list of concepts included in our analysis for each task. Additionally, as our accuracy and steering success evaluations importantly accounted for multiple possible correct answers to prompts, we provide lists of correct answers for each explored task's concept-relation pairs below. During evaluation, LLMs generate an answer of no more than 5 tokens. An LLM's answer is generally counted as correct if it contains any of the correct answers in the lists below, except for the following caveats.

In the first letter transformation and translation cases where correct answers are less than 4 characters long, the LLM must generate a correct answer on its own surrounded by whitespace or non-letters. This constraint reduces the possibility of mistakenly marking an LLM output that is not a correct answer but happens to contain one as correct, e.g., the string "te" appears in many Spanish and French words, but not all such words should be taken as correct answers for translating *you*. However, it is also common in translation for an LLM to output a token pertaining to a correct answer, then follow it up with tokens that morph its meaning (e.g., "Rotwein" when steering the LLM to output the German word for *red*). To account for this, in translation, we always mark LLM outputs beginning with a correct answer as correct. Evaluation is case-insensitive, except for the capitalization relation of the verb transformations task.

The below lists of correct answers were refined through inspecting correct answers counted as incorrect in intermediate results. Importantly, the off-the-shelf tested models achieve 100% accuracy on all tasks under the few-shot prompts used to identify components, as well as the zero-shot prompts used for ablation and steering experiments.

B.1 Country Facts

For the country facts, the full list of countries and their corresponding facts is below:

- Capital:
  - *China* → *Beijing, Peking*
  - *France* → *Paris*
  - *Germany* → *Berlin*
  - *Japan* → *Tokyo*
  - *Nigeria* → *Abuja*
  - *Poland* → *Warsaw*
  - *Russia* → *Moscow,* Москва
  - *Spain* → *Madrid*
  - *UK* → *London*
  - *USA* → *Washington*
- Currency:
  - *China* → *Yuan, ¥, RMB, CNY, renmin*
  - *France* → *Euro, Franc, €*
  - *Germany* → *Euro, €, Mark*
  - *Japan* → *Yen, ¥*
  - *Nigeria* → *Naira, ₦*
  - *Poland* → *Złoty, Zloty, PLN, Zlo*
  - *Russia* → *Ruble, ₽, RUB,* руб
  - *Spain* → *Euro, €*
  - *UK* → *Pound, £*
  - *USA* → *Dollar, $*
- Language:
  - *China* → *Mandarin, Chinese*
  - *France* → *French, Français*
  - *Germany* → *German, Deutsch*
  - *Japan* → *Japanese,* 日本語
  - *Nigeria* → *English*
  - *Poland* → *Polish*
  - *Russia* → *Russian*
  - *Spain* → *Spanish*
  - *UK* → *English*
  - *USA* → *English*

B.2 Word Translation

For translation, our evaluation scheme accounts for various aspects:

- **Parts of speech**, e.g., *poisson* (fish) and *pêcher* (to fish) as correct French translations of *fish*

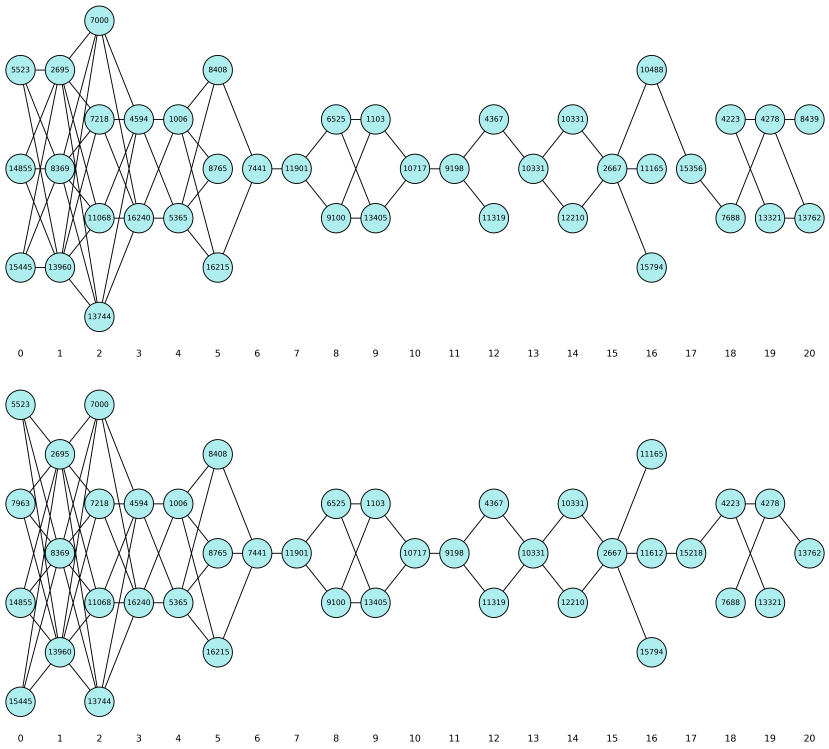


Figure 5: Gemma 2 9B China components extracted from capital and currency prompts.

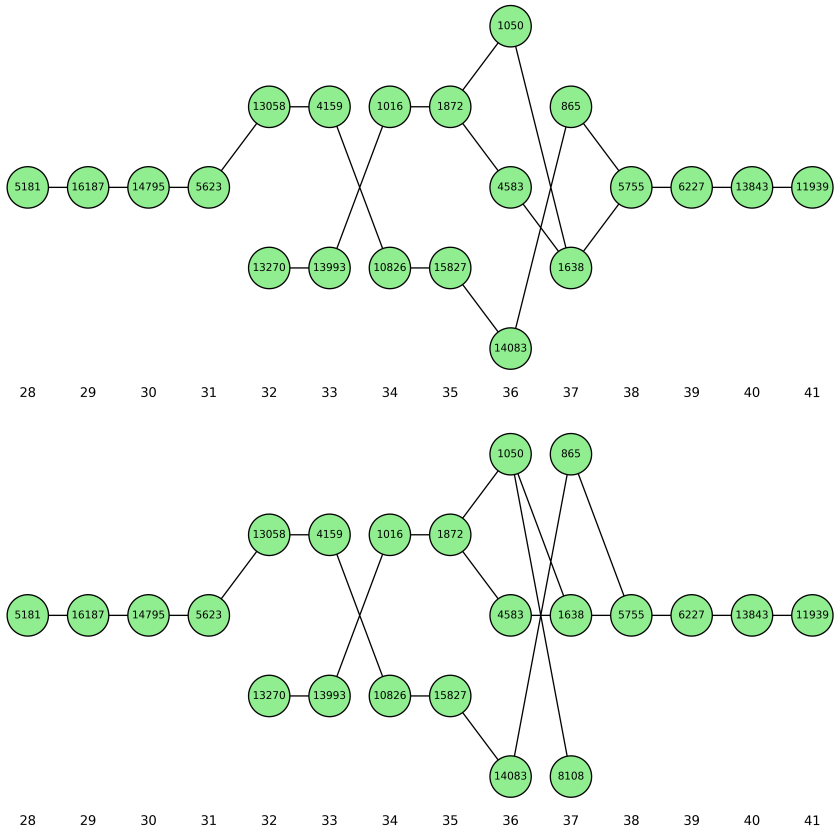


Figure 6: Gemma 2 9B language components extracted from China and Nigeria prompts.

Country	**CN**	**FR**	**DE**	**JP**	**NG**	**PL**	**RU**	**ES**	**UK**	**US**	**Average**
Success Rate	1.00	0.78	0.85	1.00	0.89	0.81	1.00	0.96	1.00	0.96	0.93

Table 6: Gemma 2 9B steering success rates for each target country across all 27 prompts that do not already query that country.

Relation	**Capital**	**Currency**	**Language**	**Average**
Success Rate	0.90	1.00	1.00	0.97

Table 7: Gemma 2 9B steering success rates for each target country fact across all 20 prompts that do not already query that country fact.

- **Verb conjugations**, e.g., *amo* and *amas* as correct Spanish translations of *love* (despite this verb taking the same form in English regardless of grammatical first or second person)
- **Semantic relevance**, e.g., *coureur* (runner) as a valid French translation of *run* despite being syntactically incorrect

These choices allow us to separate syntactic errors from semantic errors (the latter of which we focus on). The full list of correct answers is below:

- Spanish:
  - *beautiful → hermosa, hermoso, bello, bella, bonito, bonita, lindo, linda, belleza, bellisima*
  - *cat → gato, gata, gatos, gatas, felino*
  - *dog → perro, perra, perros, perras*
  - *fish → pez, pescar, pesco, pescas, pescan, pescado, pesca, faenar*
  - *good → buen, bueno, buena, bien, buenos, buenas*
  - *here → aquí, acá, aqui*
  - *learn → aprender, aprendo, aprendes, aprenden, aprende, Saber*
  - *love → amar, amor, amo, amas, aman, encanta*
  - *red → rojo, roja*
  - *run → correr, corro, corres, corren, corrida, corre, carrera, corredor*
  - *you → tú, eres, usted, le, te, ustedes, vosotros*
- French:
  - *beautiful → belle, beau, bel, beauté*
  - *cat → chat, chatte, chats, chattes, félin*
  - *dog → chien, chienne, chiens, chiennes*
  - *fish → poisson, pêcher, pêche*
  - *good → bien, bon, grand*
  - *here → ici*
  - *learn → apprendre, apprends, apprenant, savoir*
  - *love → amour, aimer, aime, aiment, aimes, aimez, aim*
  - *red → rouge*
  - *run → courir, cours, course, couler, coureur*
  - *you → tu, toi, vous, on, te*
- German:
  - *beautiful → schön, schöne, schönen*
  - *cat → Katze, Katzen*
  - *dog → Hund, Hunde*
  - *fish → Fisch*
  - *good → gut, gute, guten*
  - *here → hier*
  - *learn → lernen, lerne, lernst, lernt, lern*
  - *love → lieben, liebe, liebst, liebt, lieb*
  - *red → rot, rote, roten*
  - *run → laufen, laufe, läufst, läuft, lauf, rennen, renne, rennst, rennt, renn, Flucht*
  - *you → du, Sie, duzen*

B.3 Verb Transformation

For verb transformation, the full list of verbs and their corresponding transformations is below:

- Synonym:
  - *break → shatter, destroy, split, pause, end, fracture, stop*
  - *focus → concentrate, center, attention, attend, sharpen*
  - *hide → conceal, camouflage, bury, stash, secret*
  - *include → incorporate, contain, comprise, cover, involve, add, insert, append, package*

Ctry. / Rel.	**CN**	**FR**	**DE**	**JP**	**NG**	**PL**	**RU**	**ES**	**UK**	**US**	**Avg.**
Capital	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Currency	1.00	1.00	1.00	0.72	1.00	0.50	0.11	0.94	1.00	1.00	0.83
Language	0.50	1.00	1.00	1.00	1.00	0.78	1.00	1.00	1.00	1.00	0.93
Average	0.83	1.00	1.00	0.91	1.00	0.76	0.70	0.98	1.00	1.00	0.92

Table 8: Gemma 2 9B composite steering success rates for each target country-fact pair across all 18 prompts that do not already query the target country or fact.

  - *like → enjoy, similar, alike, love, prefer*
  - *possess → have, hold, own, control, dominate, influence, keep, succeed*
  - *sink → submerge, dip, drop, descend, plunge, drain, basin, fall, destination*
  - *understand → realize, realise, grasp, know, comprehend, get*

- Antonym:
  - *break → fix, mend, repair, continue, resume, make, whole, hold*
  - *focus → distract, unfocus, ignore, blur, lose, disperse, diversion*
  - *hide → expose, reveal, flaunt, show*
  - *include → exclude, omit, remove, delete*
  - *like → dislike, hate, avoid, unlike*
  - *possess → dispossess, lose, want, need, lack, abandon, miss, disarray, fail, not have*
  - *sink → float, rise, ascend, stand*
  - *understand → misunderstand, misinterpret, ignore, not understand*

- Past Tense:
  - *break → broke, broken*
  - *focus → focused*
  - *hide → hid, hidden*
  - *include → included*
  - *like → liked*
  - *possess → possessed*
  - *sink → sank, sunk*
  - *understand → understood*

- Capitalize:
  - *break → Break*
  - *focus → Focus*
  - *hide → Hide*
  - *include → Include*
  - *like → Like*
  - *possess → Possess*
  - *sink → Sink*
  - *understand → Understand*

- First Letter:
  - *break → B*
  - *focus → F*
  - *hide → H*
  - *include → I*
  - *like → L*
  - *possess → P*
  - *sink → S*
  - *understand → U*

Like translation, the synonym and antonym relations similarly account for a variety of interpretations of the input word, as well as some answers which have a different part of speech than the input word.

C Prompt Templates

In this appendix, we list component selection and steering evaluation prompt templates for all tasks' relations.

C.1 Country Facts

Capital city. For component selection, we use:

> *The capital city of Peru is Lima. The capital city of South Korea is Seoul. The capital city of Saudi Arabia is Riyadh. The capital city of* `{in-prompt country}` *is*

For steering evaluation, we use:

> *Q: What is the capital city of* `{in-prompt country}`*? Answer directly (two words max).*
> *A:*

Currency. For component selection, we use:

> *The currency of Peru is the Sol. The currency of South Korea is the Won. The currency of Saudi Arabia is the Riyal. The currency of* `{in-prompt country}` *is the*

For steering evaluation, we use:

> *Q: What is the currency of* `{in-prompt country}`*? Answer directly (two words max).*
> *A:*

Language. For component selection, we use:

> *The main language in Peru is Spanish. The main language in South Korea is Korean. The main language in Saudi Arabia is Arabic. The main language in* `{in-prompt country}` *is*

For steering evaluation, we use:

> *Q: What is the main language in* `{in-prompt country}`*? Answer directly (two words max).*
> *A:*

C.2 Word Translation

Spanish. For component selection, we use:

> *The Spanish word for "sing" is cantar. The Spanish word for "he" is él. The Spanish word for "bird" is pájaro. The Spanish word for "*`{in-prompt word}`*" is*

For steering evaluation, we use:

> *Q: What is the Spanish word for "*`{in-prompt word}`*"? Answer directly (two words max).*
> *A:*

French. For component selection, we use:

> *The French word for "sing" is chanter. The French word for "he" is il. The French word for "bird" is oiseau. The French word for "*`{in-prompt word}`*" is*

For steering evaluation, we use:

> *Q: What is the French word for "*`{in-prompt word}`*"? Answer directly (two words max).*
> *A:*

German. For component selection, we use:

> *The German word for "sing" is singen. The German word for "he" is er. The German word for "bird" is Vogel. The German word for "*`{in-prompt word}`*" is*

For steering evaluation, we use:

> *Q: What is the German word for "*`{in-prompt word}`*"? Answer directly (two words max).*
> *A:*

C.3 Verb Transformation

Synonym. For component selection, we use:

> *A synonym of throw is toss. A synonym of go is move. A synonym of find is discover. A synonym of* `{in-prompt verb}` *is*

For steering evaluation, we use:

> *Q: What is a synonym of* `{in-prompt verb}`*? Answer directly (one word max).*
> *A:*

Antonym. For component selection, we use:

> *An antonym of throw is catch. An antonym of go is stop. An antonym of find is lose. An antonym of* `{in-prompt verb}` *is*

For steering evaluation, we use:

> *Q: What is an antonym of* `{in-prompt verb}`*? Answer directly (one word max).*
> *A:*

Past tense. For component selection, we use:

> *The past tense of throw is threw. The past tense of go is went. The past tense of find is found. The past tense of* `{in-prompt verb}` *is*

For steering evaluation, we use:

> *Q: What is the past tense of* `{in-prompt verb}`*? Answer directly (one word max).*
> *A:*

Capitalize. For component selection, we use:

> *The word throw with the first letter capitalized is Throw. The word go with the first letter capitalized is Go. The word find with the first letter capitalized is Find. The word* `{in-prompt verb}` *with the first letter capitalized is*

For steering evaluation, we use:

> *Q: Capitalize the word* `{in-prompt verb}`*. Answer directly (one word max).*
> *A:*

First letter. For component selection, we use:

> *The first letter of throw is T. The first letter of go is G. The first letter of find is F. The first letter of* `{in-prompt verb}` *is*

For steering evaluation, we use:

> *Q: What is the first letter of the word* `{in-prompt verb}`*? Answer directly (one word max).*
> *A:*

D Additional Component KL Divergence Plots

In Figure 2, we plotted all sparse components for *love* and the three languages in the word translation task. We provide some additional examples in this appendix. Specifically, Figure 7 visualizes the component KL divergences for China and the capital city, currency, and language relations in the country facts task. Further, Figure 8 visualizes the component KL divergences for *hide* and the capitalize, first letter, and past tense relations in the verb transformation task.

E Country Facts Component Feature Description Visualizations

Figure 9 visualizes the feature descriptions for selected components in the country fact task.

F Additional Concept and Relation Component Visualizations

In this appendix, we include some additional visualizations of components extracted from studied tasks. Figures 10 and 11 visualize selected components from the word translation task, while Figure 12 visualizes selected components from the verb transformation task.

G Steering Details and Hyperparameters

For individual concept and relation steering, we selected steering strengths α_c, α_r from $\{k \cdot 0.05 : k \in \mathbb{Z}\} \cap (0, 1]$ that achieved the highest respective success rates. For composite steering, we selected the (α_c', α_r') pair from $\{\alpha_c - 0.05, \alpha_c, \alpha_c + 0.05\} \times \{\alpha_r - 0.05, \alpha_r, \alpha_r + 0.05\}$ that achieved the highest success rate. This procedure yielded the following parameters for each task:

- Country facts: $\alpha_c = 0.1$, $\alpha_r = 0.4$, $\alpha_c' = 0.15$, and $\alpha_r' = 0.45$
- Word translation: $\alpha_c = 0.05$, $\alpha_r = 0.4$, $\alpha_c' = 0.05$, and $\alpha_r' = 0.35$
- Verb transformation: $\alpha_c = 0.05$, $\alpha_r = 0.65$, $\alpha_c' = 0.05$, and $\alpha_r' = 0.7$

For the verb transformation task, we additionally tried tuning a separate steering strength for each (target) relation. For individual concept steering, this made no difference in performance. For relation steering, past tense achieved up to 3.1%, capitalization 50.0%, and first letter 87.5%. However, since antonym and synonym remained at 0% steering success rate, choosing a single α_r in facilitating composite steering was not possible. As such, we chose to omit these results to avoid unnecessarily complicating our experimental paradigm. However, for future use of this method, it may be advantageous to explore more fine-grained selection of steering strengths for heterogeneous tasks like this one.

To provide more information about how performance varies with respect to steering strength, Tables 9, 10, and 11 list the individual concept/relation and composite steering accuracies across all searched steering strengths and tasks.

H Supplementary Ablation and Steering Next Token Distributions

In this appendix, we provide additional detailed examples of how the next token distributions change under ablating and steering concept and relation components.

H.1 Individual Concept and Relation Ablation

Extending Table 1, we provide three comprehensive sets of examples for ablating individual concepts and relations: Table 12 (for country facts), Table 13 (for word translation), and Table 14 (for verb

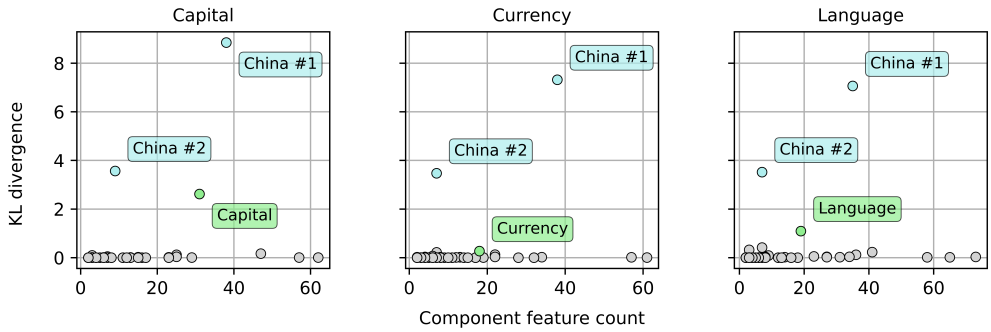


Figure 7: Plot of component feature counts versus KL divergence between pre- and post-ablation output token distributions for sparse components extracted for the capital city, currency, and language of China.

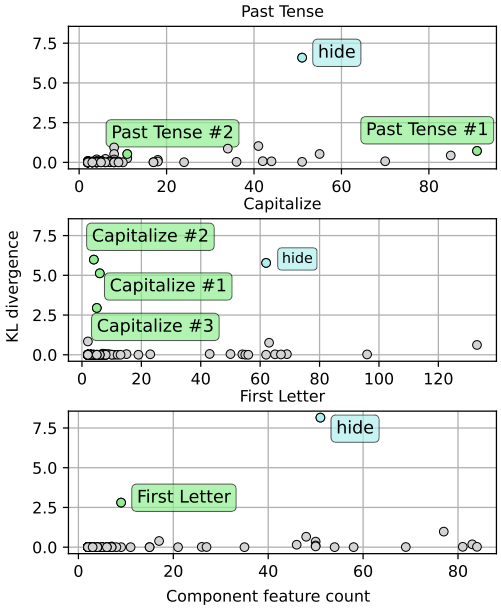
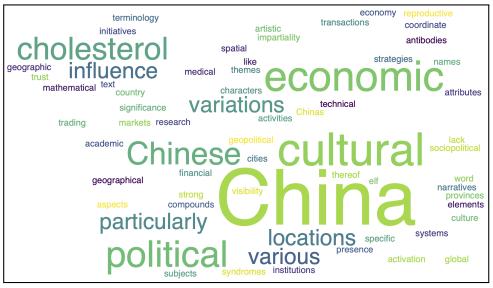


Figure 8: Plot of component feature counts versus KL divergence between pre- and post-ablation output token distributions for sparse components extracted for the capitalization, first letter, and past tense of *hide*.

Steering Strength	Country Facts	Word Translation	Verb Transformation
0.05	0.87	**0.75**	**0.48**
0.10	**0.96**	0.51	0.30
0.15	0.84	0.18	0.11
0.20	0.54	0.09	0.06
0.25	0.22	0.03	0.06
0.30	0.17	0.00	0.10
0.35	0.15	0.00	0.06
0.40	0.15	0.00	0.08
0.45	0.16	0.00	0.08
0.50	0.16	0.00	0.08
0.55	0.16	0.00	0.08
0.60	0.15	0.00	0.07
0.65	0.15	0.00	0.08
0.70	0.16	0.00	0.08
0.75	0.17	0.00	0.10
0.80	0.20	0.00	0.09
0.85	0.20	0.03	0.06
0.90	0.20	0.03	0.06
0.95	0.19	0.03	0.06
1.00	0.18	0.03	0.06

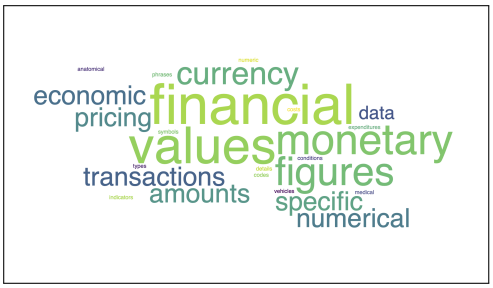
Table 9: Overall task accuracy for individual concept steering across candidate steering strengths. Accuracy under selected steering strength for each task in bold.

transformations). While the results largely mirror those highlighted in Table 1, we observe that when ablating the first letter transformation, the model still ranks tokens representing single letters highly, and many of those letters are other letters besides the first in the prompted verb (e.g., "*K*", "*E*", and "*A*" in the context of *break*). This suggests that the first letter component specifically captures first letters, and not other letters in prompted words; as such, ablating it still leaves a signal to the model to extract (non-first) letters. Further, it is possible that letter information is captured in a more general abstraction within the model which is activated by the first letter relation, but is not specific to it.

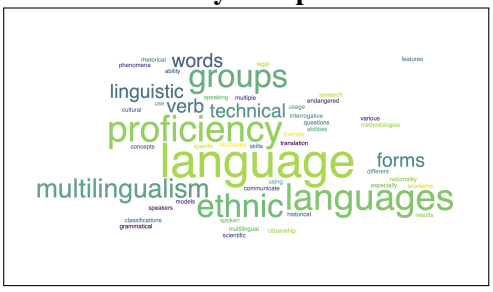


China Component

Capital Component



Currency Component



Language Component

Figure 9: Word clouds for LLM-generated descriptions of SAE features within the China, capital, currency, and language components.

Steering Strength	Country Facts	Word Translation	Verb Transformation
0.05	0.17	0.03	0.10
0.10	0.60	0.35	0.09
0.15	0.68	0.73	0.11
0.20	0.67	0.94	0.11
0.25	0.70	0.98	0.11
0.30	0.77	0.98	0.10
0.35	0.85	0.98	0.14
0.40	**0.93**	**0.98**	0.16
0.45	0.92	0.91	0.19
0.50	0.92	0.82	0.19
0.55	0.88	0.65	0.21
0.60	0.83	0.48	0.22
0.65	0.78	0.35	**0.22**
0.70	0.78	0.20	0.21
0.75	0.77	0.17	0.19
0.80	0.75	0.14	0.19
0.85	0.73	0.09	0.19
0.90	0.70	0.09	0.17
0.95	0.67	0.06	0.17
1.00	0.63	0.02	0.17

Table 10: Overall task accuracy for individual relation steering across candidate steering strengths. Accuracy under selected steering strength for each task in bold.

H.2 Individual Concept and Relation Steering

Extending Table 3, we provide three comprehensive sets of examples for steering individual concepts and relations: Table 15 (for country facts), Table 16 (for word translation), and Table 14 (for verb transformations). Observations from these extended examples largely resemble what we observed in Table 3.

H.3 Composite Concept and Relation Steering

Extending Table 4, we provide two full sets of examples for steering concepts and relations in composition: Table 18 (for country facts) and Table 19 (for word translation). Observations from these extended examples largely resemble what we observed in Table 4. An unexpected exception is that the top next token when steering from the language of China to the capital of Nigeria is "*Lagos*" (another major city in Nigeria) rather than *Abuja*. We note that the full answer from the LLM contains the names of both cities, suggesting competition between these related tokens.

I Steering Results by Concept

Table 2 summarized steering success rates by relation in each task. Here, Tables 20-22 break down concept steering success rate by concepts, while Tables 23-25 break down composite steering success rates by concept-relation pairs.

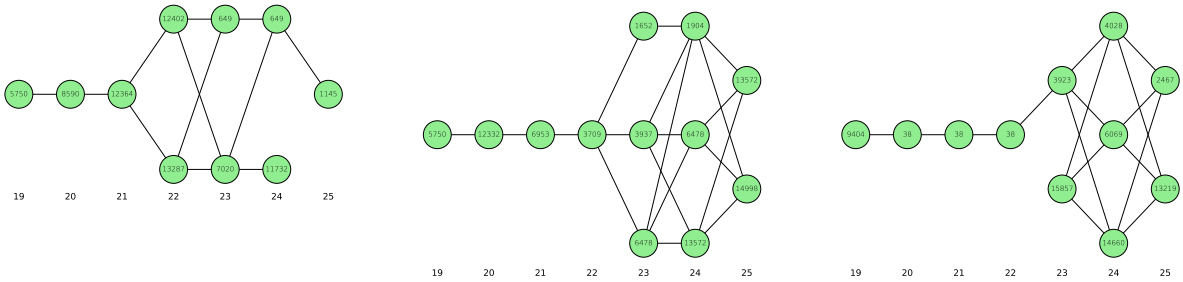


Figure 10: Components representing Spanish, French, and German translation (from left to right).

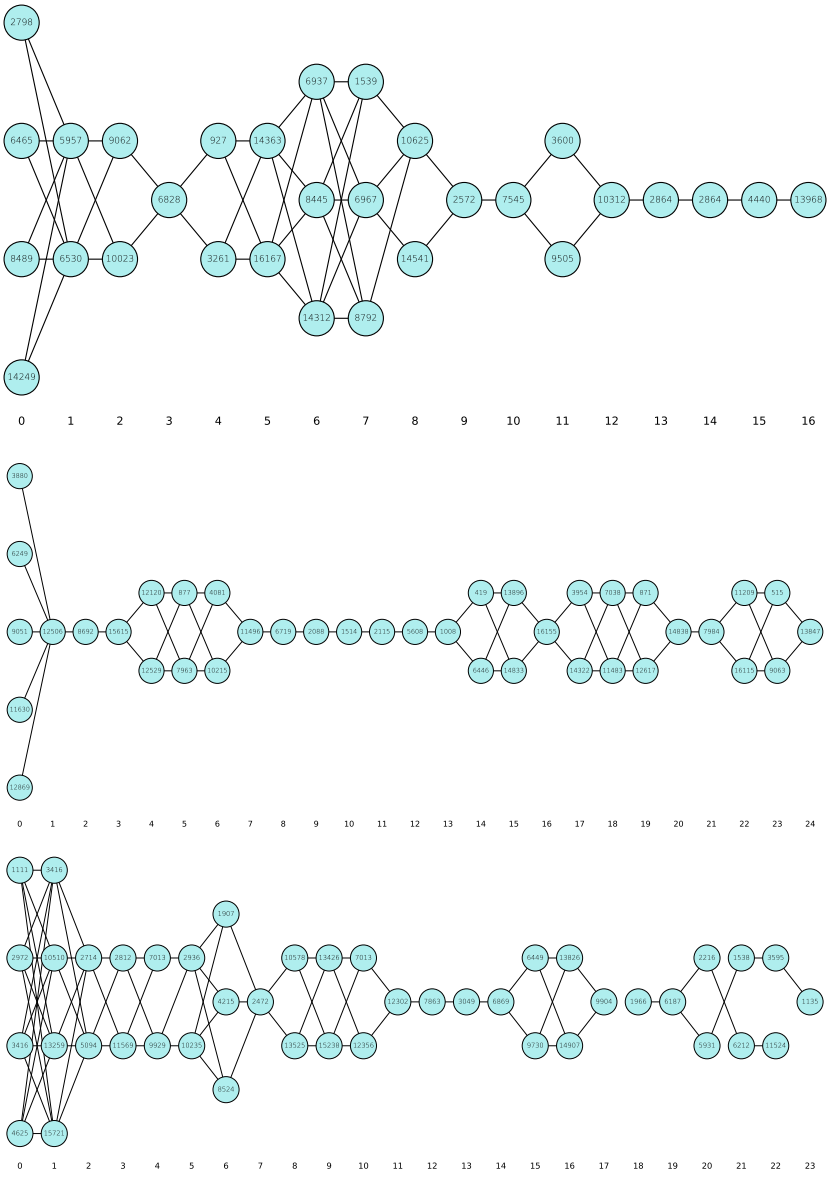


Figure 11: Components representing the translated words *beautiful*, *dog*, and *love* (from left to right).

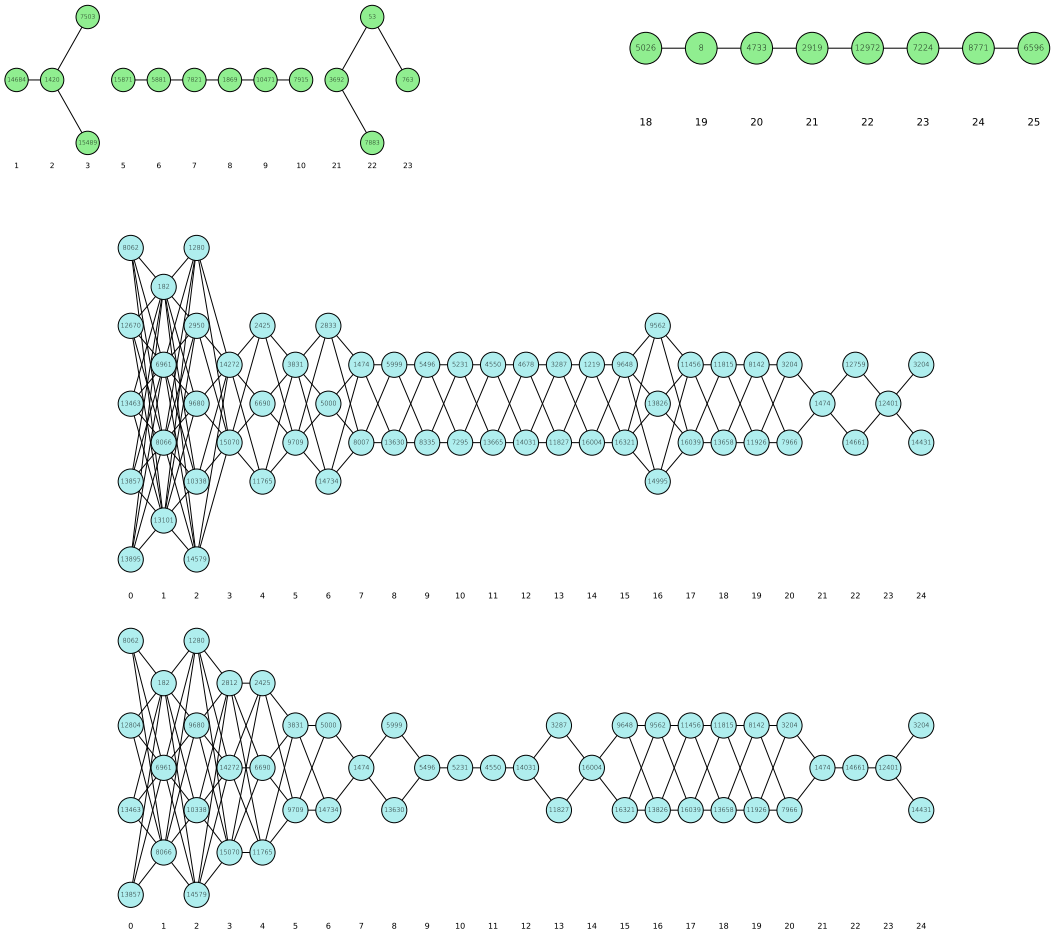


Figure 12: Components representing the capitalize (upper left) and first letter (upper right) transformations in the context of the verb *like*, and the components representing the verb *like* in the context of the capitalize (middle row) and first letter (bottom row) transformations.

Steering Strengths	Country Facts	Word Translation	Verb Transformation
0.00, 0.35	–	0.01	–
0.00, 0.40	–	0.01	–
0.00, 0.45	–	0.01	–
0.00, 0.60	–	–	0.15
0.00, 0.65	–	–	0.16
0.00, 0.70	–	–	0.16
0.05, 0.35	0.74	**0.64**	–
0.05, 0.40	0.78	0.59	–
0.05, 0.45	0.77	0.48	–
0.05, 0.60	–	–	0.17
0.05, 0.65	–	–	0.18
0.05, 0.70	–	–	**0.19**
0.10, 0.35	0.85	0.53	–
0.10, 0.40	0.85	0.47	–
0.10, 0.45	0.86	0.36	–
0.10, 0.60	–	–	0.16
0.10, 0.65	–	–	0.18
0.10, 0.70	–	–	0.18
0.15, 0.35	0.86	–	–
0.15, 0.40	0.88	–	–
0.15, 0.45	**0.90**	–	–

Table 11: Overall task accuracy for composite concept-relation steering across candidate steering strength pairs. Accuracy under selected steering strength pair for each task in bold.

J Steering Failure Analysis

In cases where steering fails, it is practically useful to understand the space of behaviors that may arise in LLMs. Figure 13 visualizes a categorization of steering failures quantified in Section 3.2.

Common failure cases. In the country facts task, most failure cases were still partly successful, reinforcing the accuracy of our component selection in this task. Specifically, most incorrect counterfactual responses were correct answers for the target fact but not country (e.g., a capital city for the wrong country, often the prompted country). Additionally, many were at least relevant to target concepts or relations, e.g., names and non-capital cities of target countries. In word translation, most failures were similarly relevant to the target language and/or word. Usually, incorrect outputs were one or more non-target words in the target language. However, in rare cases the generated words were also nearly correct, e.g., steering from the French translation of *love* to *run* yielded "à pied" (meaning "on foot"). This suggests that for these two tasks, the concept and relation components we selected were generally fairly relevant to their corresponding contexts, even if they yielded some incorrect answers.

In verb transformation, however, most relation and composite steering failures invoked unintelligible or prompt-irrelevant model outputs. This again suggests that the components for this task are less reliable or specific than others, thus manipulating them can drastically impact general model performance. Notably, when steering from other relations to capitalization, the model often generated (incorrect) sequences of capitalized tokens, e.g., "The Word, The Word." This suggests that the capitalization component primarily represents stylization of generated outputs rather than word-specific application. In concept steering, which was more effective, most failures were partial successes, suggesting that the components for verbs are more viable, and still compose predictably with more complex transformation operations. Many cases yielded relevant responses for target relations and concepts, such as the target verb in the wrong form (e.g., not capitalized or in past tense), other letters besides the first letter of the target verb, and capitalized sequences of irrelevant tokens.

Nontrivial composition. We additionally observed an interesting nontrivial composition of in-prompt and target concepts and relations in the word translation and verb transformation tasks.

In word translation, when steering from translating the word *fish* to other words across languages, the model often output translations of words related to water (e.g., *ocean*, *sea*, *river*, *boat*). This may suggest that the word *fish* is captured not only in our *fish* component, but also by other activated aquatic knowledge. When the *fish* component is ablated, this other knowledge may then integrate with target word knowledge in complex ways to yield responses that bridge the gap. For example, when steering from translating *fish* to *here* in both Spanish and German, the model generated Spanish/German words for *sea*, perhaps connecting *fish* to the location aspect of *here*. Meanwhile, when steering from translating *fish* to *run* in Spanish, the model generated a Spanish word for *river* (capable of "running"). While only anecdotal evidence, future work may consider further exploring how lexical knowledge is distributed across LLMs and synthesized during text generation.

In verb transformation, we again see nontrivial compositions of in-prompt and target relations, specifically when steering from other relations to capitalization. Here, steered outputs often consisted of capitalized tokens referring to the prompted relation, e.g., the model generated

Capital, $\frac{\text{China}}{\text{Nigeria}}$						**Currency,** $\frac{\text{China}}{\text{Nigeria}}$						**Language,** $\frac{\text{China}}{\text{Nigeria}}$					
Original		*Ctry. Abl.*		*Fact Abl.*		*Original*		*Ctry. Abl.*		*Fact Abl.*		*Original*		*Ctry. Abl.*		*Fact Abl.*	
Beijing	97.	Madrid	39.	the	12.	Yuan	80.	Euro	64.	Yuan	59.	Mandarin	59.	Spanish	49.	Chinese	24.
Be	.38	Warsaw	10.	China	6.7	Ren	14.	Lira	20.	Ren	14.	Chinese	37.	English	22.	China	18.
Peking	.35	Rome	9.5	Beijing	6.4	RMB	1.8	Krone	3.1	Yen	1.9	English	.69	French	6.7	Mandarin	11.
Shanghai	.23	Paris	6.1	Shanghai	5.0	Yen	.98	Franc	2.4	P	1.7	also	.43	German	4.0	also	3.5
Xi	.11	Berlin	5.0	a	2.7	yuan	.53	Peso	1.2	Ba	1.5	Put	.22	Italian	2.7	a	2.9
Abuja	85.	New	7.7	Lagos	17.	Naira	93.	Franc	16.	Naira	62.	English	72.	English	50.	Nigeria	40.
Lagos	11.	Islamabad	7.3	the	12.	N	2.9	Euro	9.1	Dollar	6.0	Ha	11.	French	31.	English	14.
...	.38	Kathmandu	6.0	Nigeria	7.1	Nai	2.5	D	7.4	Niger	4.3	Yoruba	4.7	Spanish	2.1	Nigerian	9.8
Nigeria	.29	Delhi	5.7	called	5.1	naira	.36	Pound	5.1	Currency	2.4	Igbo	2.4	Arabic	2.0	...	2.9
...	.27	Tehran	4.9	Abuja	3.4	K	.23	Krone	4.8	Nai	1.4	Nigerian	1.6	Dutch	1.1	also	2.3

Table 12: Top 5 output tokens and their likelihoods for China (top) and Nigeria (bottom) across all three relations, before and after ablating the relevant country or relation component in Gemma 2 2B. Underscore prefixes of tokens are omitted for space.

Spanish, $\frac{\text{love}}{\text{red}}$						**French,** $\frac{\text{love}}{\text{red}}$						**German,** $\frac{\text{love}}{\text{red}}$					
Original		*Word Abl.*		*Lang. Abl.*		*Original*		*Word Abl.*		*Lang. Abl.*		*Original*		*Word Abl.*		*Lang. Abl.*	
Amor	41.	"	5.9	Love	42.	Amour	35.	Le	15.	Love	42.	Liebe	66.	Du	9.0	love	16.
amor	19.	El	4.9	love	14.	amour	26.	La	6.4	I	11.	Lie	4.8	"	6.6	Liebe	13.
Amo	7.0	La	4.4	I	13.	A	5.7	"	6.1	love	10.	Ich	4.4	Die	4.4	Love	9.6
amo	5.1	Malo	3.2	"	5.3	J	3.8	le	3.7	"	5.8	"	3.4	das	4.1	I	8.7
"	2.9	la	2.5	LOVE	2.7	"	2.7	la	3.3	LOVE	2.5	Herz	1.7	Der	3.9	"	6.8
rojo	38.	La	3.5	Red	27.	Rouge	43.	La	4.2	Red	36.	Rot	51.	"	4.6	red	30.
Rojo	34.		3.4	red	15.	rouge	31.	"	3.4	red	19.	rot	28.	Sch	2.8	Red	17.
rojo	2.1	"	3.2	"	8.1	Rou	4.7		3.2	The	5.9	rote	5.8	ge	2.5	rojo	6.1
Ro	2.0	El	2.8	The	5.1	ROU	2.6	Le	2.7	Rouge	5.6	Rote	3.5		2.2	Rojo	4.3
Roja	1.9	Bol	2.4	...	4.1	Le	2.2	En	2.6	"	5.5	ro	2.0	Gel	2.2	rouge	3.7

Table 13: Top 5 output tokens and their likelihoods for *love* (top) and *red* (bottom) across all three languages, before and after ablating the relevant word or language component. Underscore prefixes of tokens are omitted for space.

Past Tense, $\frac{\text{break}}{\text{like}}$						**Capitalize,** $\frac{\text{break}}{\text{like}}$						**1st Letter,** $\frac{\text{break}}{\text{like}}$					
Original		*Verb Abl.*		*Trans. Abl.*		*Original*		*Verb Abl.*		*Trans. Abl.*		*Original*		*Verb Abl.*		*Trans. Abl.*	
Broke	37.	-	9.9	Past	8.7	Break	22.	Capital	10.	break	20.	B	62.	T	40.	K	30.
broke	27.	T	5.0	T	7.7	break	13.	The	7.6	Break	7.1	b	17.	C	11.	E	14.
Broken	7.6	Was	4.6	(	7.4	BREAK	8.3		4.2		5.1	A	1.9	A	10.	k	13.
BRO	6.2	To	4.5	t	5.6	Capital	5.7	I	3.6	capital	4.3	E	1.5	O	9.9	A	5.7
Bre	2.8	O	4.3	E	5.1	The	5.0	Yes	2.3	The	3.3	Break	1.3	t	6.7	e	5.1
liked	48.	was	8.7	like	8.4	Like	21.	Capital	5.6	like	12.	L	38.	E	48.	A	13.
Liked	34.	have	7.6	Past	8.3	like	12.	The	4.7		6.1	l	8.7	e	9.8	I	6.8
-	1.1	Have	7.0	T	8.2	Capital	8.6	Yes	4.5	capital	5.6	A	6.2	H	9.5	a	6.7
Loved	1.0	had	5.9	t	7.2	LIKE	4.3	No	4.1	the	4.6	I	4.1	F	6.6	E	4.6
Liked	1.0	-	5.9	Like	7.2	The	4.0		3.3	Like	3.8	"	3.2	T	3.8	"	4.2

Table 14: Top 5 output tokens and their likelihoods for *break* (top) and *like* (bottom) across three verb transformations, before and after ablating the relevant verb or transformation component. Underscore prefixes of tokens are omitted for space.

Capital, CN						**Currency, CN**						**Language, CN**					
·, NG		*Currency, ·*		*Language, ·*		*·, NG*		*Capital, ·*		*Language, ·*		*·, NG*		*Capital, ·*		*Currency, ·*	
_Abuja	87.	_Yuan	13.	_Mandarin	35.	_Naira	75.	_Beijing	97.	_Chinese	38.	_English	71.	_Beijing	96.	_Yuan	12.
_Nigeria	4.6	_yuan	11.	_Chinese	28.	_naira	8.3	_Be	.94	_Mandarin	38.	_Yoruba	5.6	_Be	2.0	_Ren	8.4
_Lagos	3.9	_RMB	11.	_English	20.	_Nai	3.7	_BE	.57	_English	20.	_Ha	4.8	_Peking	.93	_China	6.1
_-	.58	_China	7.0	_mandarin	2.4	_Nigeria	3.3	_Peking	.33	_Simplified	.92	_Nigeria	4.4	_BE	.53	_Chinese	5.9
_	.57	_Ren	4.9	_Spanish	1.8	_Nigerian	2.4	Beijing	.20	_Spanish	.56	_Igbo	3.4	Beijing	.37	_RMB	5.9

Table 15: Top 5 output tokens and likelihoods for prompts about China's capital (columns 1–3), currency (columns 4–6), and language (columns 7–9), after ablating an in-prompt country or fact component (row 1 headings) and amplifying a target country (Nigeria) or fact component (row 2 headings). Highlighted tokens are or may begin correct answers for steered components.

Spanish, love						**French, love**						**German, love**					
·, red		*French, ·*		*German, ·*		*·, red*		*Spanish, ·*		*German, ·*		*·, red*		*Spanish, ·*		*French, ·*	
_rojo	43.	_amour	14.	_Liebe	71.	_rouge	44.	_amor	41.	_Liebe	68.	_rot	53.	_amor	50.	_amour	14.
_Rojo	8.7	_A	11.	_Ich	2.3	_Rouge	30.	_Amor	13.	_die	2.4	_Rot	33.	_el	8.6	_"	9.7
_Sang	5.7	_"	6.7	_die	1.5	_Sang	2.5	_el	6.9	_„	2.1	_rote	3.6	_"	7.4	_l	7.5
_roja	2.8	_l	6.5	_„	1.5	_Rou	1.8	_"	6.1	_lieben	1.7	_Rote	2.0	_Amor	3.7	_A	6.7
_rojo	1.8	_Amour	5.3	_zu	1.4	_	1.2	_A	3.1	_Ich	1.6	_Blut	1.2	_a	2.7	_	6.5

Table 16: Top 5 output tokens and likelihoods for prompts to translate the word *love* into Spanish (columns 1–3), French (columns 4–6), and German (columns 7–9), after ablating an in-prompt word or language component (row 1 headings) and amplifying a target word (*red*) or language component (row 2 headings). Highlighted tokens are or may begin correct answers for steered components.

Past Tense, understand						**Capitalize, include**						**1st Letter, break**					
·, break		*Cap., ·*		*1st Lett., ·*		*·, understand*		*Past Tense, ·*		*1st Lett., ·*		*·, focus*		*Past Tense, ·*		*Cap., ·*	
_broke	28.	_Understand	6.7	_U	39.	_Understand	14.	⊗		35.	_i	22.	_F	26.	⊗		30.	_Break	21.
_Broke	21.	_Know	6.4	_understood	9.3	_The	10.	_surla		12.	_I	21.	_C	18.	_surla		16..	_"	6.1
_BRO	8.5	_Past	5.9	_Understand	8.5	_I	6.3	AddTagHelper		13.	_include	21.	_O	15.	AddTagHelper		13.	_The	5.6
_Bre	6.3	_Answer	5.7	_u	3.6	_understand	4.3	_These		9.7	_Include	6.3	_f	4.4	_These		11.	_“	5.5
_brake	2.6	_	4.5	_United	3.2	_	4.1	_nahilalakip		6.9	_In	1.9	_o	3.3	_nahilalakip		8.6	_	3.7

Table 17: Top 5 output tokens and likelihoods for prompts to get the past tense (columns 1–3), capitalized form (columns 4–6), and first letter (columns 7–9) of various verbs, after ablating an in-prompt verb or transformation component (row 1 headings) and amplifying a target verb (*understand*) or transformation component (row 2 headings). Highlighted tokens are or may begin correct answers for steered components. "*s*" refers to Old English long S.

Capital, China				**Currency, China**				**Language, China**			
Currency, Nigeria		*Language, Nigeria*		*Capital, Nigeria*		*Language, Nigeria*		*Capital, Nigeria*		*Currency, Nigeria*	
_Nigeria	43.	_English	75.	_Abuja	70.	_English	96.	_Lagos	73.	_Naira	48.
_Naira	30.	_Yoruba	4.0	_Lagos	25.	_Yoruba	1.1	_Abuja	23.	_Nigeria	17.
_naira	8.9	_Igbo	3.6	_	3.3	_French	.74	_	2.3	_Dollar	5.8
_N	3.0	_French	3.5	_Nigeria	.34	_Spanish	.60	_Nigeria	.46	_	4.5
_	2.8	_Spanish	3.0	_...	.32	_Igbo	.54	_...	.32	_naira	4.1

Table 18: Top 5 output tokens and their likelihoods for prompts about China's capital (columns 1–2), currency (3–4), and language (5–6), after ablating both in-prompt components and amplifying a target country-fact pair. Highlighted tokens are or may begin correct answers for steered components.

Spanish, love				French, love				German, love			
French, red		*German, red*		*Spanish, red*		*German, red*		*Spanish, red*		*French, red*	
_rouge	47.	_rot	31.	_rojo	43.	_rot	28.	_rojo	53.	_rouge	45.
_Rouge	8.2	_Rot	26.	_roja	6.5	_Rot	20.	_el	7.2	_le	6.7
_Sang	3.2	_rote	11.	_el	6.2	_rote	9.0	_roja	5.6	_Rouge	4.2
_sang	1.9	_die	2.1	_El	3.2	_die	2.9	_"	3.1	_	3.7
_	1.9	_Rote	1.6	_"	2.5	_roten	2.5	_El	2.2	_la	3.4

Table 19: Top 5 output tokens and their likelihoods for prompts to translate *love* into Spanish (columns 1–2), French (3–4), and German (5–6), after ablating both in-prompt components and amplifying a target word-language pair. Highlighted tokens are or may begin correct answers for steered components.

Country	**CN**	**FR**	**DE**	**JP**	**NG**	**PL**	**RU**	**ES**	**UK**	**US**	**Average**
Success Rate	1.00	1.00	1.00	1.00	1.00	0.78	1.00	0.78	1.00	1.00	0.96

Table 20: Steering success rate for each target country across all 27 prompts for other countries.

Word	**cat**	**dog**	**fish**	**you**	**beautiful**	**good**	**red**	**here**	**learn**	**love**	**run**	**Average**
Success Rate	0.60	0.87	0.47	0.57	0.93	0.83	0.90	0.47	1.00	1.00	0.63	0.75

Table 21: Steering success rate for each target translated word across all 30 prompts for other words.

Verb	**break**	**focus**	**hide**	**include**	**like**	**possess**	**sink**	**understand**	**Average**
Success Rate	0.66	0.34	0.69	0.63	0.83	0.06	0.20	0.46	0.48

Table 22: Steering success rate for each target verb across all 35 prompts for other verbs.

Ctry. / Fact	**CN**	**FR**	**DE**	**JP**	**NG**	**PL**	**RU**	**ES**	**UK**	**US**	**Average**
Capital	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Currency	1.00	0.94	1.00	0.22	0.61	0.50	0.50	0.94	1.00	1.00	0.77
Language	0.50	1.00	1.00	1.00	1.00	0.78	1.00	1.00	1.00	1.00	0.93
Average	0.83	0.98	1.00	0.74	0.87	0.76	0.83	0.98	1.00	1.00	0.90

Table 23: Composite steering success rates for each target country-fact pair across all 18 prompts that do not already query the target country or fact.

Word / Lang.	**cat**	**dog**	**fish**	**you**	**beautiful**	**good**	**red**	**here**	**learn**	**love**	**run**	**Average**
Spanish	0.55	0.65	0.10	0.50	1.00	0.20	0.75	0.55	0.70	1.00	0.25	0.57
French	0.75	0.90	0.05	0.55	0.95	0.90	0.65	0.50	0.95	1.00	0.35	0.69
German	0.55	0.45	0.15	0.50	0.80	1.00	0.75	0.60	1.00	1.00	0.55	0.67
Average	0.62	0.67	0.10	0.52	0.92	0.70	0.72	0.55	0.88	1.00	0.38	0.64

Table 24: Composite steering success rates for each target word-language pair across all 20 prompts that do not already query the target word or language.

Verb / Trans.	**break**	**focus**	**hide**	**include**	**like**	**possess**	**sink**	**understand**	**Average**
Synonym	0.00	0.00	0.00	0.00	0.14	0.00	0.00	0.00	0.02
Antonym	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Past Tense	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Capitalize	0.00	0.00	0.93	0.14	0.18	0.00	0.00	0.04	0.16
First Letter	1.00	0.00	0.89	0.93	0.71	1.00	1.00	0.75	0.79
Average	0.20	0.00	0.36	0.21	0.21	0.20	0.20	0.16	0.19

Table 25: Composite steering success rates for each target verb-transformation pair across all 10 prompts that do not already query the target verb or transformation.

"The Past Tense Of The Word" when steering from past tense, "The Opposite of" when steering from antonym, and "The First Letter Of The" when steering from first letter. As mentioned in Section 3.2, this provides more evidence that the capitalization relation primarily represents the stylizing of generated text rather than the capability to manipulate it in a context-specific manner. Further, the intermingling of in-prompt and target information suggests that information about prompted relations is very widely distributed across many network layers and feature directions, even though the components for relations like past tense, antonym, and first letter are relatively large.

K Supplementary Results on Component Organization

Extending Figure 4 from Section 3.3, we provide additional KL divergences by node and layer in Figure 14 (for translated words), Figure 15 (for target translation languages), Figure 16 (for verbs to transform), and Figure 17 (for verb transformations).

L Supplementary Results on Taxonomical Reasoning

Beyond the tasks in the paper, we additionally attempted to apply our method to a dataset for the taxonomic class, order, and family of 26 animals. However, we excluded these results from the paper after discovering that Gemma 2 2B could not perform this task perfectly off-the-shelf, thus steering based on extracted components also performed poorly.

Interestingly, however, causally impactful components for animal species exhibited a very strong hierarchical structure across the three taxonomy tasks. Generally, the component for a given species at the class level was a subgraph of the component for the same species at the order level, and the order level component a subgraph of the family level component. (18) This can be phrased as a general observation that as we move from more general to more specific categories, components increase in size by adding features.

We were able to reconstruct a reasonably accurate taxonomic hierarchy of the animals in our dataset by applying agglomerative clustering on the Jaccard distance matrix between family components of different species. The taxonomy is very accurate at class level, with some notable mistakes at order and family level. This coheres with previous results on LLMs' general performance on taxonomy tasks (Sun et al., 2024).

In future work, we intend to explore simpler tasks based on hierarchical knowledge.

M Transcoder Analysis

Transcoders (Dunefsky et al., 2024) are a major development in interpretability research. They operate by reconstructing the multilayer perceptron layer outputs from the inputs to the MLP layer.

To investigate whether our approach could be used to identify components for transcoders, we extend our translation task to the Gemma Scope 2 2B transcoders (Lieberum et al., 2024) and focus on language components for French, German, and Spanish. The model's outputs were quite robust to ablating top-activated features, so we were only able to identify causal components for *dog*, *run*, *beautiful*, *learn*, *love*, *red*, and *good*. In initial experiments, we found that for some prompts, we could ablate essentially all of the top-activated features without changing the model's output, so we attribute this poor performance at least in part to relevant information being carried in the error term.

We found that a correlation threshold of 0.7 was more suitable for identifying causal components in transcoders, by testing decreasing values in increments of 0.5 and inspecting the results. We then ran our algorithm as described, with the exception of skipping the density pruning step. The lower threshold caused us to identify very large components with quite a bit of overlap. We assigned only the nodes unique to each language to the final components used for ablation and steering.

We conducted a relation steering experiment, ablating the component identified for the original language and steering the component identified for another. Testing integer values up to 20, we found that 13 produced the best accuracy. Our final accuracy was 27%.

This performance is not as good as our method achieves with SAEs, and comes with quite a few limitations. However, through manual inspection we found that the same causal features that Hanna et al. identify in their language intervention demo notebook are present in our identified components. While our method may need refining and adaptation to work at scale with transcoders, based on these results we feel confident that feature coacti-

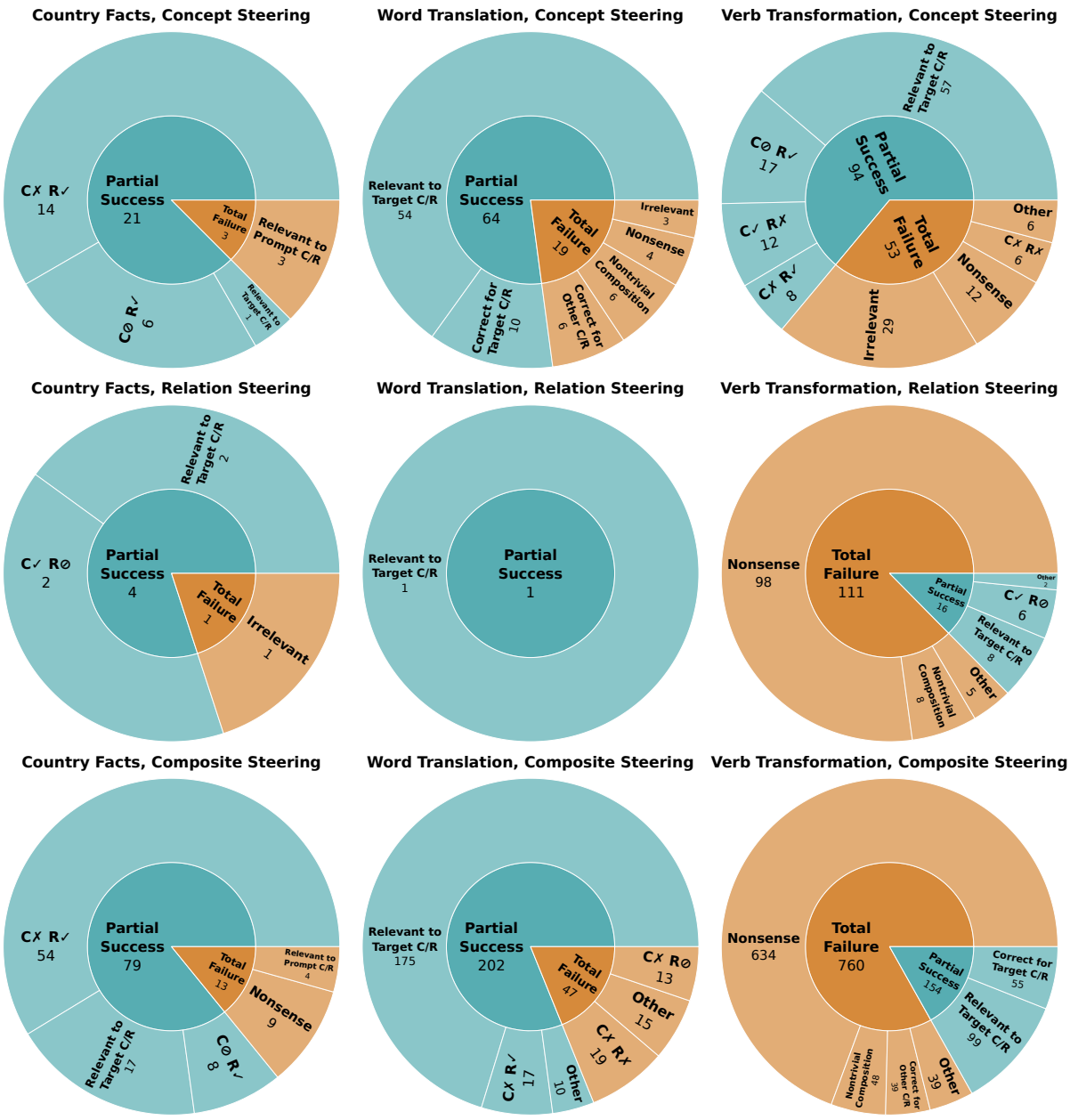


Figure 13: Coarse and fine categorization of steering failures for all tasks and steering types. For coarse categories, *Partial Success* refers to steered model outputs that were apparently related or close to the expected answer for the target concept-relation pair, while *Total Failure* describes outputs that were not, including both counterfactually incorrect and incoherent outputs. In the fine categories, *C* and *R* refer to concept and relation. ✓ indicates a steered model output that was a correct answer for the target concept or relation, ∅ for the in-prompt concept or relation, and ✗ for other concepts or relations; such cases are sparser in composite steering on the verb transformations task, and are thus condensed into broader categories. *Other* refers to fine categories that make up less than 4% of total failures, which are condensed for readability. While *Irrelevant* refers to fluent outputs that were not clear responses to their corresponding prompts, *Nonsense* refers to outputs that were not clearly fluent text, including odd tokens (e.g., "AddTagHelper", "⊗⊗⊗⊗⊗"), repeated tokens (e.g., "few few few few few"), or otherwise unintelligible sequences of tokens (e.g., "1. The").

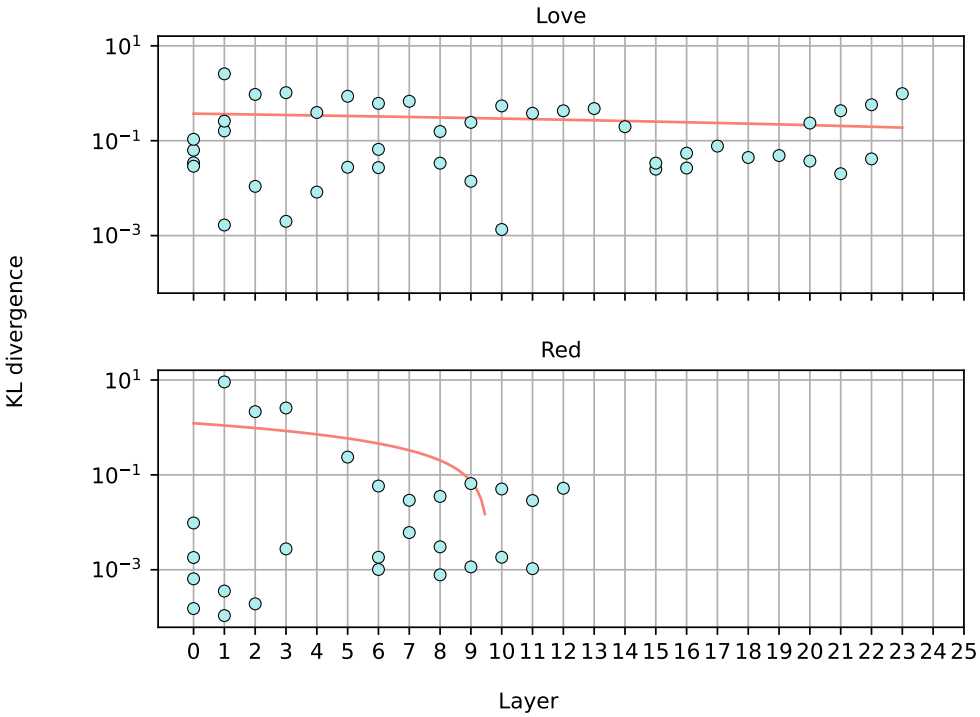


Figure 14: KL divergence between pre- and post-ablation output token distributions for each node in the *love* and *red* components from the word translation task, plotted by layer. Linear regression lines plotted in red.

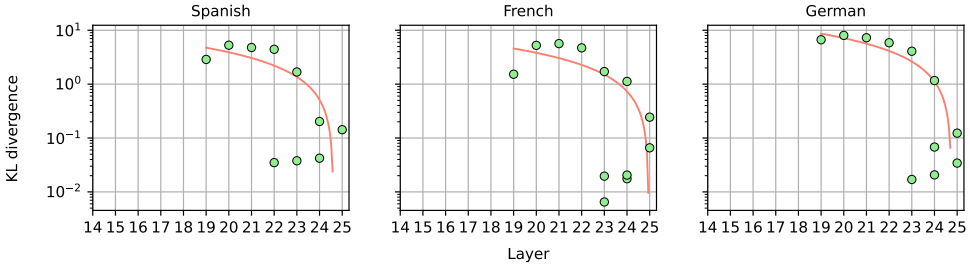


Figure 15: KL divergence between pre- and post-ablation output token distributions for each node in the translation target language components, plotted by layer. Linear regression lines plotted in red.

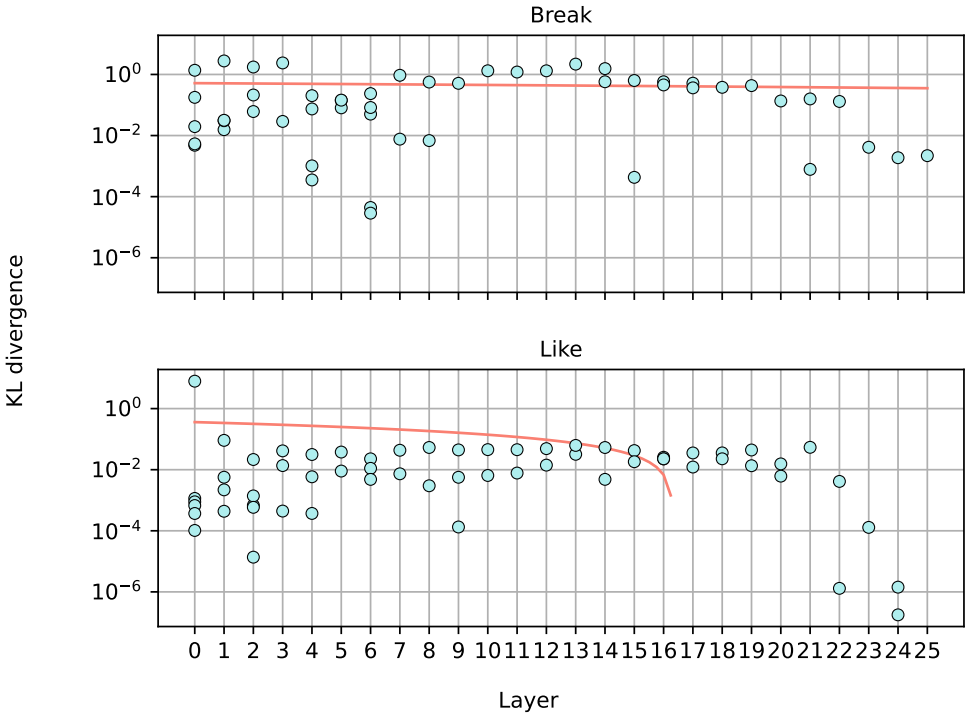


Figure 16: KL divergence between pre- and post-ablation output token distributions for each node in the *break* and *like* components from the verb transformation task, plotted by layer. Linear regression lines plotted in red.

vation can help identify relevant conceptual components in transcoders.

N License Information

We provide available licenses and terms of use for key artifacts employed in this work, including relevant links:

- **Hugging Face Transformers**
  - License: Apache 2.0 (https://github.com/huggingface/transformers/blob/main/LICENSE)
  - Terms of Service: https://huggingface.co/terms-of-service
- **Gemma 2 2B**
  - License: Apache 2.0 (https://github.com/google-deepmind/gemma/blob/main/LICENSE)
  - Terms of Use: https://ai.google.dev/gemma/terms
- **Gemma 2 9B**
  - License: Apache 2.0 (https://github.com/google-deepmind/gemma/blob/main/LICENSE)
  - Terms of Use: https://ai.google.dev/gemma/terms
- **Gemma Scope**
  - License: Apache 2.0 (https://huggingface.co/google/gemma-scope-2b-pt-res/blob/main/LICENSE)
  - Terms of Use: https://ai.google.dev/gemma/terms
- **Transformer Lens**
  - License: MIT (https://github.com/TransformerLensOrg/TransformerLens/blob/main/LICENSE)
- **SAE Lens**
  - License: MIT (https://github.com/jbloomAus/SAELens/blob/main/LICENSE)
- **NetworkX**
  - License: 3-clause BSD (https://github.com/networkx/networkx/blob/main/LICENSE.txt)
- **Neuronpedia API**

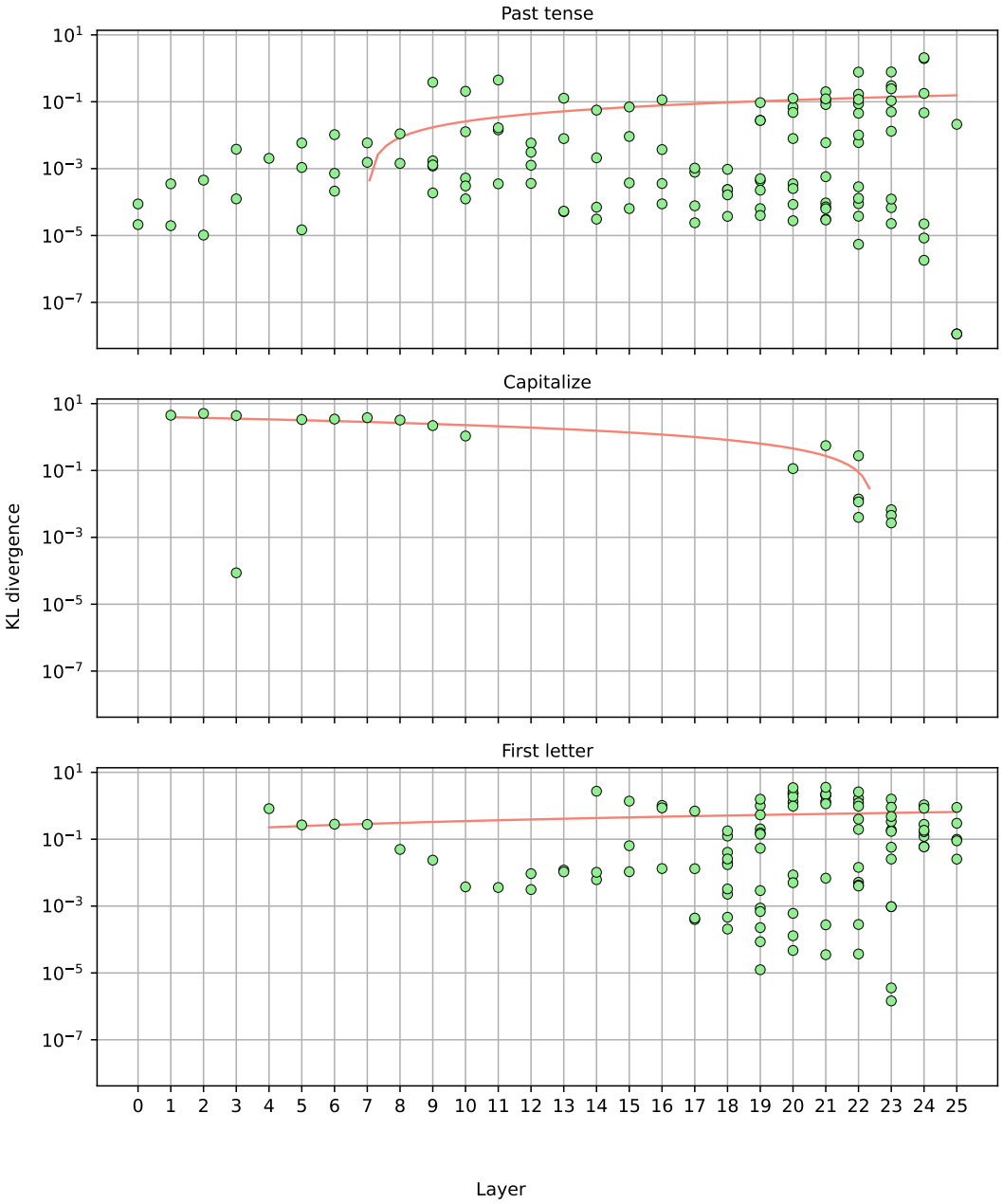


Figure 17: KL divergence between pre- and post-ablation output token distributions for each node in the verb transformation components, plotted by layer. Linear regression lines plotted in red.

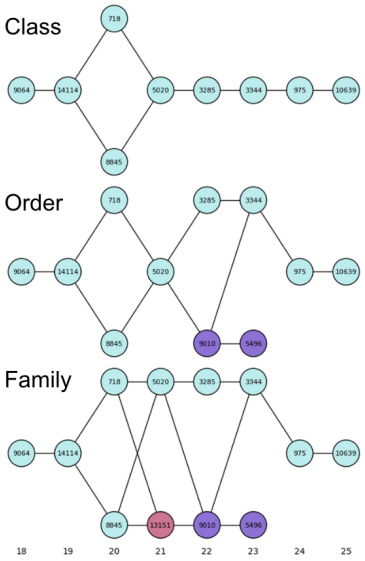


Figure 18: Species components for Angelfish as extracted from class, order, and family tasks. Nodes present in all three are shown in blue, nodes present in order and family are shown in purple, and nodes present in just family are shown in red.

- License: MIT (https://github.com/hijohnnylin/neuronpedia/blob/main/LICENSE)

We have verified that this work acts in accordance with all available licenses and terms of use.