Sparse Feature Coactivation Reveals Composable 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 country-relation tasks, we show that ablating semantic components for countries and relations changes model outputs in predictable ways, while amplifying these components induces counterfactual responses. Notably, composing relation and country components yields compound counterfactual outputs (Figure 1). We find that, whereas most country components emerge from the very first layer, the more abstract relation components are concentrated in later layers; within relation components themselves, nodes from later layers tend to have a stronger causal impact on model outputs. Overall, these findings suggest a modular organization of knowledge within LLMs and advance methods for efficient, targeted model manipulation.

1 Introduction

Sparse autoencoders (SAEs) have emerged as a powerful tool for extracting interpretable features from large language models (LLMs), but it remains unclear how these features integrate across layers to produce coherent responses. In this work, we uncover modular semantic structures in the form of networks of coactivating SAE features. Compared to Ameisen et al. [1], our approach does not require manual grouping of features. We also provide a more granular, feature-level view of the mechanism identified by Merullo et al. [25]. For a full discussion of related work, see Appendix A.

We focus our analyses on Gemma 2 2B [34], with additional results for Gemma 2 9B in Appendix D. For each prompt, we collect activations from pre-trained SAEs [21, 3] and select the set of features that appear in the top-5 activations at any token position for each layer. Second, we construct a directed graph where nodes are the selected features and an edge connects features in adjacent layers if their activation patterns have a Pearson correlation over 0.9. Third, to filter out overly generic features, we prune this graph by removing any feature with an activation density greater than 0.01 according to Neuronpedia [22]. Finally, we identify the resulting weakly connected components using a standard BFS algorithm [15] and validate their causal role by ablating or amplifying their activations [27] and measuring the model's output distribution shift.

*Equal contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

2 Methods

We focus our analyses on Gemma 2 2B[2] [34], with additional results for Gemma 2 9B included in Appendix D. For each prompt, we collect SAE feature activations across layers, construct an inter-layer feature network based on coactivation patterns, prune high-density features, extract task-relevant connected components, and perform targeted causal interventions to assess their functional roles (Figure 1). The detailed steps are as follows:

Activation collection: We run input prompt through the model integrated with pre-trained SAEs from the `gemma-scope-2b-pt-res-canonical` release (`width_16k/canonical` variant, loaded via SAE Lens) [21, 3]. 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 $t \in \{1, \ldots, T\}$:

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

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 features $i \in S_\ell$ and $j \in S_{\ell+1}$, we compute the Pearson correlation coefficient:

$$\rho\left(\Phi_\ell[:, i], \Phi_{\ell+1}[:, j]\right) = \frac{\text{cov}(\Phi_\ell[:, i], \Phi_{\ell+1}[:, j])}{\sigma(\Phi_\ell[:, i])\sigma(\Phi_{\ell+1}[:, j])}$$

A directed edge $e = ((\ell, i), (\ell + 1, j))$ is added to E if $\rho(\Phi_\ell[:, i], \Phi_{\ell+1}[:, j]) > \tau_{\text{corr}} = 0.9$. Edge weights are assigned as $w(e) = \min(1.0, \rho)$.

Density-based pruning: Some SAE features activate frequently across unrelated contexts, making them overly generic and hard to interpret. To eliminate such noise, we prune the graph using activation density scores from Neuronpedia [22].[3] For each node $(\ell, i) \in V$, we retrieve its activation density $d_{\ell,i}$, defined as the fraction of tokens in a large corpus where the feature activates. We retain only *sparse features*, those with $d_{\ell,i} \leq \tau_{\text{density}} = 0.01$, creating a pruned graph G_{sparse}. This threshold follows Neuronpedia's standard, which classifies features below this density as sparse and interpretable. We also remove any isolated nodes from G_{sparse}.

Component identification: We use a straightforward BFS-based method implemented by NetworkX[4] to identify weakly connected components within G_{sparse}.

Causal validation: To evaluate the functional significance of each component, we perform targeted interventions using `TransformerLens` [27]. Specifically, we ablate or amplify the activations of SAE features in a given component during the model's forward pass and 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 manipulation leads to systematic and interpretable changes in model behavior. Additional details and results are provided in Section 3.

3 Experimental Results

We focused on country-capital, country-currency, and country-language tasks for China, France, Germany, Japan, Nigeria, Poland, Russia, Spain, the United Kingdom, and the United States. To

[2] Accessed via `google/gemma-2-2b` through Hugging Face Transformers [36]. LLM activations collected with a single NVIDIA A100 GPU (40GB VRAM) and 12GB RAM.

[3] `https://neuronpedia.org`

[4] `https://networkx.org/documentation/stable/_modules/networkx/algorithms/components/connected.html`

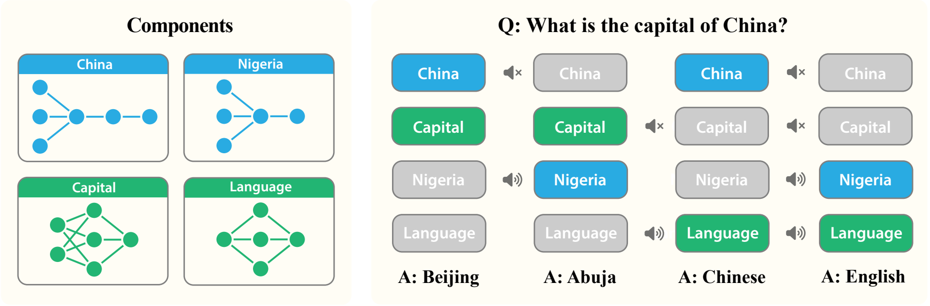


Figure 1: Selective component ablation and amplification steers model toward counterfactual outputs.

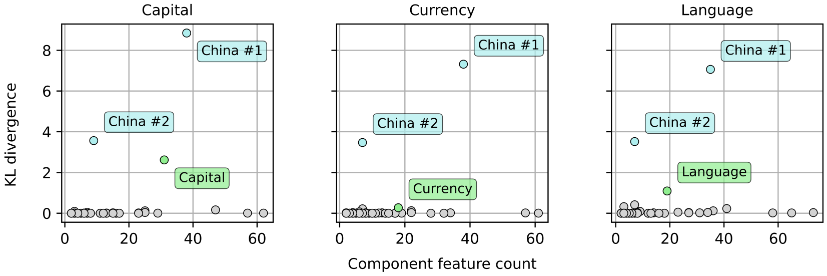


Figure 2: Component feature count (x-axis) vs. causal effect (KL divergence) when ablated (y-axis).

collect model activations for each country-capital pair, we used the following prompt template with in-context examples: "*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 for country-currency and country-language pairs.

3.1 Component Identification

For each country-relation pair, we obtained around 70 connected components using the methods outlined in Section 1. Typically, two to three of the extracted components exert a markedly higher causal effect on the model output than the others (Figure 2).

Semantic coherence. To evaluate the roles of these top components, we began by inspecting their associated feature descriptions from Neuronpedia. In most cases, the features within a given component had thematically coherent descriptions, often referring to a common country or relation (Appendix C). However, there are exceptions—for instance, none of the high-impact components obtained from Spain-related prompts explicitly mention Spain in their feature descriptions. Given that feature descriptions are not always reliable, we performed component ablations and observed the resulting changes in the model's top predicted tokens. Table 1 presents results from ablation experiments using components obtained from China and Nigeria-related prompts. Promisingly, when country components were ablated, the model's top predicted tokens shifted predictably to the capitals, currencies, or languages of other countries. When relation components were ablated, the model assigned higher probabilities to country names.

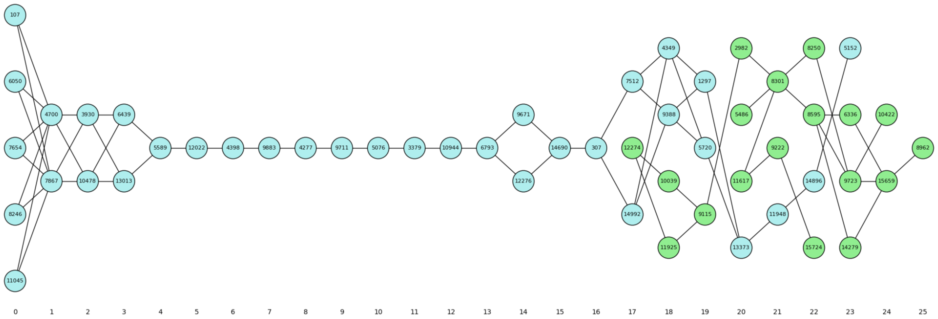


Figure 3: China (blue) and language (green) components.

Capital						**Currency**						**Language**					
Original		*Ctry. Abl.*		*Rel. Abl.*		*Original*		*Ctry. Abl.*		*Rel. Abl.*		*Original*		*Ctry. Abl.*		*Rel. 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 1: Top five 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.

Capital, China						**Currency, China**						**Language, China**					
·, Nigeria		*Currency, ·*		*Language, ·*		*·, Nigeria*		*Capital, ·*		*Language, ·*		*·, Nigeria*		*Capital, ·*		*Currency, ·*	
_Abuja	87.	_Yuan	15.	_Mandarin	32.	_Naira	75.	_Beijing	98.	_Chinese	.35	_English	71.	_Beijing	96.	_Yuan	15.
_Nigeria	4.6	_yuan	12.	_Chinese	27.	_naira	8.3	_Be	.75	_Mandarin	.33	*_Yoruba*	5.6	_Be	1.5	_Ren	10.
_Lagos	3.9	_RMB	11.	*_English*	25.	_Nai	3.7	_BE	.34	*_English*	.28	*_Ha*	4.8	*_Peking*	.83	_RMB	6.8
_-	.58	_Ren	5.6	*_Spanish*	2.2	*_Nigeria*	3.3	*_Peking*	.27	*_Simplified*	.82	*_Nigeria*	4.4	Beijing	.42	_yuan	6.3
_	.57	*_China*	5.4	_mandarin	2.1	*_Nigerian*	2.4	Beijing	.25	*_Spanish*	.79	*_Igbo*	3.4	_BE	.30	*_The*	5.3

Table 2: Top five output tokens and likelihoods for prompts about China after ablating an in-prompt component and amplifying a target country or relation.

Context consistency. We found that country and relation components are remarkably consistent across different contexts. Therefore, in subsequent experiments, we define each country component as the intersection of all the components for that country across relations; similarly, each relation component is defined as the intersection of all the components for that relation across countries. Figure 3 shows the resulting China and Language components.

3.2 Component Steering

Having identified distinct graph components for each country and relation, we next investigate whether these components can be used to steer model outputs individually and in combination. To test whether components generalize across different contexts, we applied a test prompt template different from the one used to collect the initial activations: *"Q: What is the* {*capital city of / currency of / main language in*} {*country*}*? Answer directly (two words max). A:"*.

Country steering. By ablating an *in-prompt country* component and amplifying a *target country* component, we successfully directed the LLM to respond to questions about the capital, currency, and language of the in-prompt country with *counterfactual answers*, i.e., the capital, currency, or language for the target country, which is not actually queried in the prompt. As shown in Table 2, when we ablated the China component and amplified the Nigeria component, the model consistently responded with the desired counterfactual answers "*Abuja*", "*Naira*", and "*English*" for capital, currency, and language questions, respectively—disregarding the prompt's reference to China. Overall, country steering successfully produced the desired counterfactual answers 96% of the time (Table 3). This confirms that our identified country components encode country-specific information that causally determines model outputs.

Relation steering. Similarly, by ablating an *in-prompt relation* while amplifying a *target relation*, we successfully directed the model to respond to queries about the *in-prompt relation* as though they concerned the *target relation*. As shown in Table 2, with the capital component ablated, the model answered a question about China's capital with "*Yuan*" and "*Mandarin*" when currency and language

CN	FR	DE	JP	NG	PL	RU	ES	UK	US	Avg.	Cap.	Curr.	Lang.	Avg.
1.00	1.00	1.00	1.00	1.00	0.78	1.00	0.78	1.00	1.00	0.96	0.95	0.90	0.90	0.92

Table 3: Steering success rates for each target country and relation.

Capital, China				**Currency, China**				**Language, China**			
Currency, Nigeria		*Language, Nigeria*		*Capital, Nigeria*		*Language, Nigeria*		*Capital, Nigeria*		*Currency, Nigeria*	
_Naira	34.	_English	71.	_Abuja	70.	_English	93.	_Abuja	49.	_Naira	40.
_Nigeria	28.	*_French*	5.1	*_Lagos*	26.	*_French*	1.7	*_Lagos*	46.	*_Nigeria*	18.
_naira	13.	*_Yoruba*	4.1	_	2.3	*_Yoruba*	1.5	_	1.7	*_Dollar*	6.2
_N	4.1	*_Spanish*	3.8	*_Nigeria*	.41	*_Spanish*	1.1	*_Nigeria*	.78	_naira	4.7
_	3.2	*_Igbo*	3.0	_...	.30	*_Igbo*	.55	_...	.36	_	3.7

Table 4: Top five output tokens and likelihoods for prompts about China after ablating both in-prompt components (row 1) and amplifying a target country-relation pair (row 2).

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	0.94	1.00	0.72	0.61	0.50	0.11	0.94	1.00	1.00	0.78
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	0.91	0.87	0.76	0.70	0.98	1.00	1.00	0.90

Table 5: Composite steering success rates for each target country-relation pair.

components were respectively amplified. We observed similar results for currency and language prompts. The average success rate for relation steering is 92% (Table 3).

Composite steering. We conducted composite steering experiments where both country and relation components were manipulated at once. By ablating both the *in-prompt country* and *in-prompt relation* components while amplifying the *target country* and *target relation* components, it is indeed possible to steer the model to ignore both the *in-prompt country* and *in-prompt relation* and answer about a different country-relation pair. Table 4 provides specific examples of composite steering success. As shown in the first column, when we ablated both the China and capital components while amplifying the Nigeria and currency components, the model correctly answered "*Naira*" despite being asked about China's capital. The average steering success rate for composite steering is 90% (Table 5).

3.3 Component Organization

Having established the causal role and composability of country and relation components, we next analyze their distribution across network layers and the relative importance of individual nodes within these components. To quantify the causal importance of an individual country node, we compute the average post-ablation KL divergence from the original output distribution across all relations. For a relation node, we compute the same average across all countries. Country and relation components show distinct distribution patterns across model layers. Eight out of the ten country components tested begin in the first layer of the network; some (e.g., China) span nearly all layers, while others (e.g., Nigeria) concentrate in early to middle layers. In contrast, all three relation components appear only in later layers of the network (Figure 4). This suggests that representations of concrete entities are established earlier in processing, while those of abstract relational concepts emerge later. Not only do all three relation components concentrate in later layers, but we find that, even within each relation component, nodes from later layers tend to have a stronger causal impact on model outputs. This is not the case for country nodes, which exhibit variable relationships between layer depth and KL divergence ranging from positive to negative.

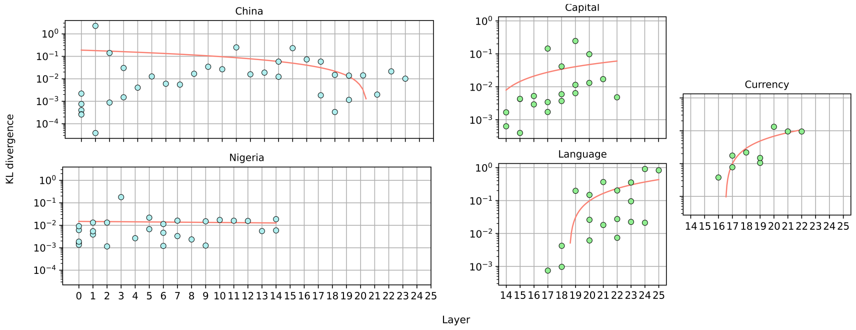


Figure 4: Node-wise KL divergence between pre- and post-ablation output token distributions.

References

[1] 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, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 2025.

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

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

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

[5] 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, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

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

[7] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 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, 2022.

[8] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

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

[10] 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. Toy models of superposition. *Transformer Circuits Thread*, 2022. https://transformer-circuits.pub/2022/toy_model/index.html.

[11] 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, Dario Amodei, Tom Brown, Jack Clark, Jared Kaplan, Sam McCandlish, and Chris Olah. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 2021. https://transformer-circuits.pub/2021/framework/index.html.

[12] Mor Geva, Jasmijn Bastings, Katja Filippova, and Amir Globerson. 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, 2023.

[13] Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5484–5495, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[14] Nicholas Goldowsky-Dill, Chris MacLeod, Lucas Sato, and Aryaman Arora. Localizing model behavior with path patching, 2023.

[15] Aric Hagberg, Daniel Schult, and Pieter Swart. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[16] Michael Hanna, Ollie Liu, and Alexandre Variengien. 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, 2023.

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

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

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

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

[21] Tom Lieberum, Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Nicolas Sonnerat, Vikrant Varma, János Kramár, Anca Dragan, Rohin Shah, and Neel Nanda. Gemma scope: Open sparse autoencoders everywhere all at once on gemma 2, 2024.

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

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

[24] Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[25] Jack Merullo, Carsten Eickhoff, and Ellie Pavlick. Language models implement simple Word2Vec-style vector arithmetic. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *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, June 2024. Association for Computational Linguistics.

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

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

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

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

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

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

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

[33] Lee Sharkey, Bilal Chughtai, Joshua Batson, Jack Lindsey, Jeff Wu, Lucius Bushnaq, Nicholas Goldowsky-Dill, Stefan Heimersheim, Alejandro Ortega, Joseph Bloom, Stella Biderman, Adria Garriga-Alonso, Arthur Conmy, Neel Nanda, Jessica Rumbelow, Martin Wattenberg, Nandi Schoots, Joseph Miller, Eric J. Michaud, Stephen Casper, Max Tegmark, William Saunders, David Bau, Eric Todd, Atticus Geiger, Mor Geva, Jesse Hoogland, Daniel Murfet, and Tom McGrath. Open problems in mechanistic interpretability, 2025.

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

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

[36] 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, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.

[37] Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and Huajun Chen. Knowledge circuits in pretrained transformers. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 118571–118602. Curran Associates, Inc., 2024.

A Related Work

Mechanistic interpretability aims to reverse-engineer neural networks into human-interpretable algorithms. Central to this field is the hypothesis that large networks can be understood as compositions of subnetworks, known as circuits, that perform specific functions [11, 28, 33]. Early efforts involved manual identification of circuits for specific tasks such as numerical comparison [16] and indirect object recognition [35]. Work on in-context learning revealed specialized mechanisms like induction heads, which detect repeated subsequences and predict their completion [29]. To improve scalability, automated circuit discovery methods were developed, including path patching [14], ACDC [6], CD-T [18], and edge pruning [2]. However, these approaches are often computationally expensive and can yield circuits that are difficult for humans to interpret.

The theory of superposition proposes that networks represent more features than dimensions by encoding sparse features across polysemantic neurons [10]. Dictionary learning techniques such as SAEs promise greater interpretability by extracting monosemantic features from polysemantic neurons [5, 19]. However, standard SAEs face challenges such as inconsistent feature quality, poor reconstruction, and weak functional alignment—issues that recent work has sought to address through architectural and training improvements [4, 31, 32]. SAEs have also allowed circuit discovery to operate on interpretable features instead of neurons [23]. Circuit tracing efforts by Anthropic [1] utilized transcoders [9], an alternative approach for extracting human-interpretable features from LLMs. Despite these advances, high computational costs remain a significant barrier. More recently, coactivation patterns have been explored for understanding SAE feature organization. Li et al. [20] analyzed the geometry of SAE features, finding spatial clustering of related concepts. Building on this approach, we construct directed graphs based on feature coactivation and introduce node pruning based on activation density. This allows us to discover semantically coherent, context-consistent connected components that can influence model outputs individually or in combination. Our approach offers a computationally efficient framework for analyzing and controlling LLM behavior without exhaustive circuit tracing.

Factual knowledge is believed to reside in the feedforward layers of transformer-based LLMs. Geva et al. [13] characterized these layers as key-value memories mapping textual patterns to vocabulary distributions. Dai et al. [7] identified "knowledge neurons" whose activations correlate with specific facts. These insights have informed approaches for editing factual knowledge stored in LLMs [8, 26, 24]. Geva et al. [12] described the recall of factual associations as a three-step process involving subject enrichment, relation propagation, and attribute extraction. Hernandez et al. [17] demonstrated that relation decoding in transformers can be approximated by simple linear transformations. Merullo et al. [25] showed that LLMs implement Word2Vec-style vector arithmetic to solve some relational tasks. Recent work on "knowledge circuits" has begun tracing causal pathways underlying factual recall [37, 30]. Leveraging SAEs, our method offers a more human-interpretable analysis of LLMs' knowledge organization by identifying emergent connected components that correspond to task-related concepts.

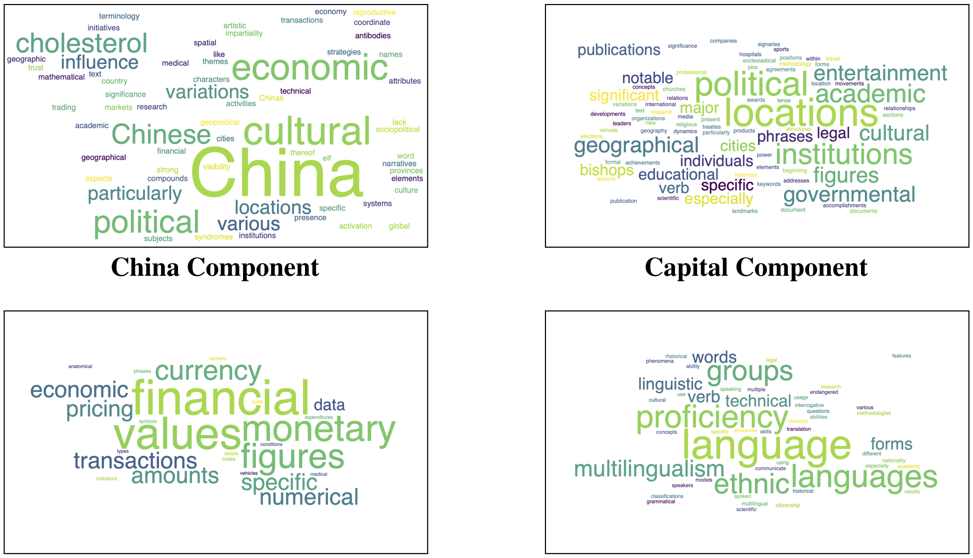
B Model and Hyperparameter Selection

Our choice of LLMs was constrained to those with pretrained SAEs available via Neuronpedia. 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.

For individual country 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 parameters $\alpha_c = 0.1$, $\alpha_r = 0.45$, $\alpha'_c = 0.15$, and $\alpha'_r = 0.45$.

C Component Feature Visualizations

Figure 5 shows word clouds for LLM-generated descriptions [22] of SAE features within the China, capital, currency, and language components.



China Component **Capital Component**

Currency Component **Language Component**

Figure 5: Component word clouds.

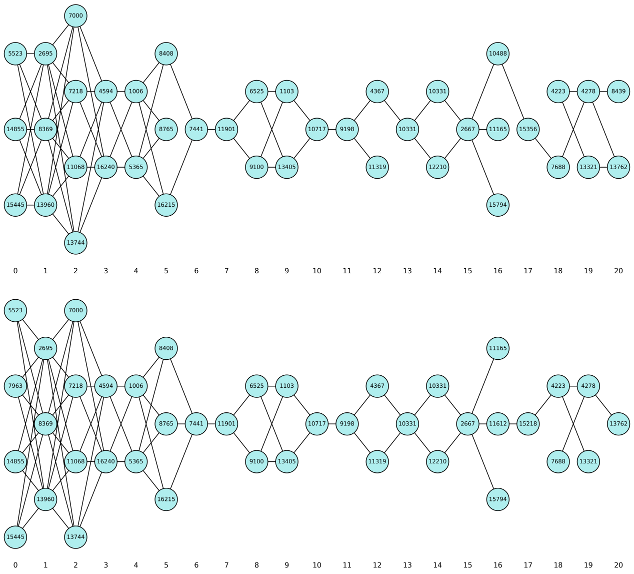


Figure 6: China components extracted from capital and currency prompts (Gemma 2 9B).

D Gemma 2 9B Results

We replicated all experiments 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 6) and similar relation components across countries (Figure 7). High success rates were achieved for country (93%), relation (97%), and composite (92%) steering (see Tables 6–7).

CN	FR	DE	JP	NG	PL	RU	ES	UK	US	Avg.	Cap.	Curr.	Lang.	Avg.
1.00	0.78	0.85	1.00	0.89	0.81	1.00	0.96	1.00	0.96	0.93	0.90	1.00	1.00	0.97

Table 6: Steering success rates for each target country and relation (Gemma 2 9B).

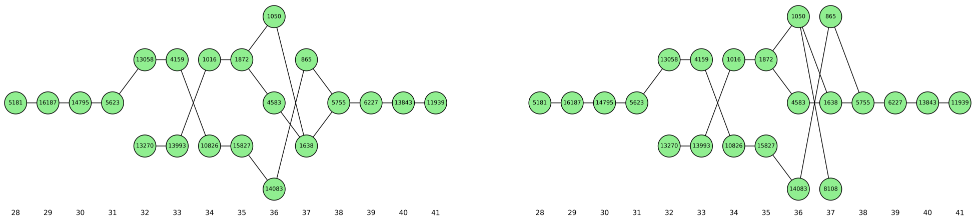


Figure 7: Language components extracted from China and Nigeria prompts (Gemma 2 9B).

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 7: Composite steering success rates for each target country-relation pair (Gemma 2 9B).

E 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**
  - 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.