RouterInterp: Understanding Superposed Specialisation in MoE Routing

Anonymous authors
Paper under double-blind review

Abstract

Sparse Mixture of Experts (MoE) models scale more efficiently than dense models by routing tokens to modular expert networks that are only active when relevant to the task. A leading hypothesis for the performance of MoE models is that each expert specialises in a single, coherent domain. However, interpretability efforts that assume this hypothesis have generally been unsuccessful. We propose and present evidence for an alternative account that we call the *Superposed Specialisation Hypothesis* (SSH): experts specialise in a disjoint union of fine-grained features rather than one broad domain. Leveraging the SSH, we introduce *RouterInterp*, a method for interpreting expert routing that identifies Sparse Autoencoder features most predictive of routing decisions and produces unified natural language explanations. On gpt-oss-20b, explanations from RouterInterp predict expert routing with 77% higher accuracy than prior methods. This work provides a scalable method for generating concise and more accurate explanations of expert routing and increases our understanding of a previously uninterpretable component of foundation models.

1 Introduction

Sparse Mixture-of-Experts (MoE) transformers have emerged as a promising approach for scaling frontier language models (Cai et al., 2025; Fedus et al., 2022b; Du et al., 2022). The strong performance of Sparse MoE models has often been attributed to expert specialisation (Lewis et al., 2021): if each expert learns to handle a subset of the input data distribution or perform only a subset of computations, then using only a subset of the parameters for each input can be both *effective* and *efficient*. We call this explanation for the success of MoEs the **Specialisation Hypothesis**.

Prior work has struggled to recover interpretable patterns of expert specialisation (Jiang et al., 2024; Lewis et al., 2021; Zoph et al., 2022). We formalise two distinct forms of the Specialisation Hypothesis. Under the **Domain Specialisation Hypothesis** (DSH), each expert specialises in a single coherent domain: semantically similar fine-grained categories (micro-domains) cluster within an expert. Under the **Superposed Specialisation Hypothesis** (SSH), each expert specialises in a disjoint collection of features spanning multiple unrelated micro-domains, mirroring the superposition of features in shared neurons (Elhage et al., 2022).

Inspired by SSH, we develop **RouterInterp**, a method that interprets MoE routing via sparse autoencoder (SAE) latents (Makhzani & Frey, 2013; Cunningham et al., 2024; Bricken et al., 2023): it identifies SAE latents most predictive of each expert's activation, generates natural language explanations for those latents, and aggregates them into expert-level explanations (Section 3). RouterInterp achieves 89% and 81% accuracy in predicting expert routing from SAE features on OLMoE-1B-7B and gpt-oss-20b, and attains a 0.62 explanation score on gpt-oss-20b, outperforming token-based explanations (0.35) (Section 4).

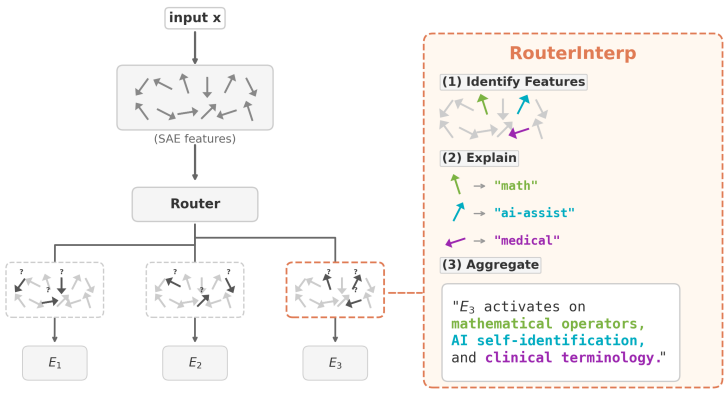

Figure 1: RouterInterp explains routing decisions as a combination of interpretable features. Sparse autoencoder (SAE) latents, sparse feature representations extracted from model activations, pick out directions in activation space that correspond to interpretable features. RouterInterp *identifies* SAE latents that are most predictive of routing, *explains* these features in concise natural language strings, and *aggregates* these explanations into an expert-level explanation. While no single monosemantic concept captures an expert's behavior, aggregating multiple features makes routing understandable.

2 Specialisation Hypotheses

Here, we formalise the two hypotheses (DSH and SSH) introduced in Section 1, state their predictions, and give theoretical motivations for why SSH might hold in practice. Suppose that we have a set of D micro-domains occurring in a corpus $\mathcal{C}$, denoted $\mathcal{D} = \{d_1, \ldots, d_D\}$. Suppose also that we have an MoE layer with E experts. Firstly, note that when $D = E$, then we should expect the experts to specialise in a single micro-domain. Secondly, when $D < E$, then we should expect multiple experts to specialise in the same micro-domain or for there to be redundant experts which are never routed to. Both of these cases are consistent with both the DSH and SSH. However, in case of $D > E$, the two hypotheses make different predictions. Under the DSH, similar micro-domains are routed to the same expert and so an expert specialises in a semantically coherent domain consisting of adjacent micro-domains - expert specialisation can be explained by a single (monosemantic) concept or domain. Under the SSH, dissimilar micro-domains are routed to the same expert and so an expert specialises in a collection of semantically disjoint domains - expert specialisation can only be explained by multiple (polysemantic) concepts or domains.

There are two core arguments for why we might expect the Superposed Specialisation Hypothesis to be a more accurate description of expert specialisation. First, we argue that the router is incentivised to assign dissimilar domains to the same expert so that the expert can perform Computation in Superposition (Hänni et al., 2024; Linsefors & Bushnaq, 2025a;b; Newgas, 2025), i.e., a single expert can have multiple disjoint transforms that it applies depending on the input domain. This mirrors Elhage et al. (2022): just as sparse features that are not frequently co-activated can share the same neurons, dissimilar domains that are not frequently co-activated can share an expert. In both cases, low co-activation reduces the *interference* cost of multiple features or domains sharing the same capacity, which is why superposition is effective for models: it allows representing more features (micro-domains) than the number of available neurons (experts). Second, MoE training commonly includes a *load-balancing loss* that encourages the router to distribute tokens evenly across experts (Shazeer et al., 2017a; Fedus et al., 2022a). With relatively small batch sizes, this can unintentionally encourage routing dissimilar inputs to the same expert: a single sequence typically contains tokens from the same macro-domain, so a small batch contains few macro-domains and the load-balancing loss distributes tokens from these few across all experts. This necessarily requires that some tokens from the same macro-domain will be routed to different experts - in opposition to the DSH, where we would expect all tokens from the same macro-domain to be routed to the same expert.

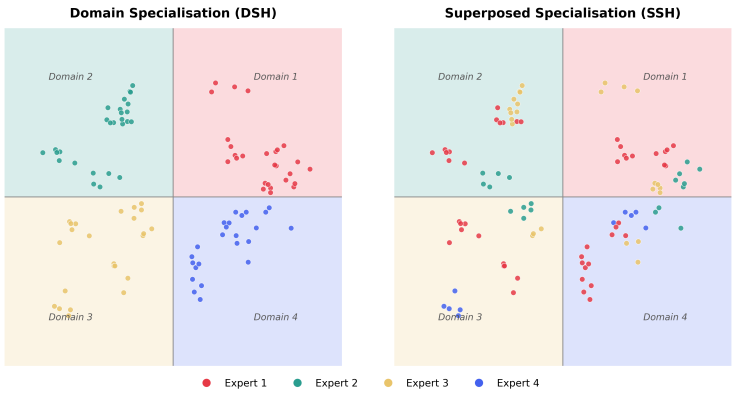

Figure 2: Two hypotheses for expert specialisation in MoE models: (a) Domain Specialisation (DSH) and (b) Superposed Specialisation (SSH). (a) Under the DSH, examples from the same semantic domain (coloured region) are routed to the same expert (same colour); experts specialise in one coherent domain. (b) Under the SSH, examples routed to the same expert (same colour) are scattered across disjoint domains (different coloured regions); clusters of points with the same expert are *micro-domains* (fine-grained features), and experts specialise in a disjoint union of micro-domains. We suggest the SSH better describes expert specialisation in practice.

3 RouterInterp

RouterInterp operates in three stages: (1) identifying SAE features most relevant to each expert's routing behaviour, (2) generating natural language explanations for each feature, and (3) aggregating these into expert-level explanations.

Feature Selection. We identify which SAE features are most predictive of each expert's activation using the 'ρ-usefulness' metric (Ilyas et al., 2019), a data-driven selection criterion that measures how discriminative a feature is for expert selection. A feature f is ρ-useful for expert E_i if the expected activation of f is higher when expert E_i is selected than when it is not:

$$\rho(f, E_i) = \mathbb{E}[f(\boldsymbol{x}) \mid E_i \in \mathcal{T}(\boldsymbol{x})] - \mathbb{E}[f(\boldsymbol{x}) \mid E_i \notin \mathcal{T}(\boldsymbol{x})] \tag{1}$$

where $\mathcal{T}(\boldsymbol{x})$ denotes the set of selected experts for input $\boldsymbol{x}$. Features with high ρ-usefulness scores are specifically predictive of a particular expert's activation. For each expert E_i, we select the top-n features ranked by ρ-usefulness score, denoting this set of features $\mathbb{F}_i$[1].

Feature Explanations. Given the selected feature set $\mathbb{F}_i$ for each expert E_i, we generate natural language explanations for each feature following the automated interpretability framework of (Paulo et al., 2025). For each feature $f \in \mathbb{F}_i$, we present a language model (the *explainer*) with *positive examples* where f activates strongly and *contrastive negatives* where f does not activate despite being semantically similar (in embedding space). The explainer infers what concept or pattern the feature detects, producing a concise natural language description.

Aggregation. To obtain an expert-level explanation, we prompt a language model with all n feature explanations for expert E_i and ask it to synthesise a single coherent paragraph describing when the expert activates. The goal is an explanation that enables predicting when the expert activates; we evaluate this using an LLM as a proxy for human judgment.

[1]We evaluate the effect of feature set size $|\mathbb{F}_i|$ on explanation quality and compare against an alternative selection method based on cosine similarity between router weight vectors and SAE decoder directions in Section F.

	Layer 4	Layer 8	Layer 12	Layer 16	Layer 20
RouterInterp	**0.60**	**0.64**	**0.61**	**0.62**	**0.60**
Bigram Baseline	0.23	0.47	0.42	0.35	0.40

Table 1: RouterInterp's natural language explanations predict expert routing with much higher explanation scores than explanations generated from high-frequency token statistics. Explanation scores measure how accurately a language model can predict expert activation given only the explanation. Across different depths of gpt-oss-20b, RouterInterp outperforms the bigram based explanations by 36–161%. Each expert explanation aggregates the 10 most ρ-useful SAE features.

Evaluation. To evaluate the quality of the explanations, we adapt the AutoInterp scoring framework (Paulo et al., 2025). First, we construct a balanced evaluation set for each expert containing *positive examples* (contexts where a token was routed to the expert) and *hard negatives* (semantically similar contexts with no tokens routing to that expert). Then, an LLM (the *scorer*) receives the expert explanation and the example from the evaluation set and predicts whether the expert would be activated. This yields a binary prediction per example, which we compare against the ground truth routing decision. Final explanation score is reported as F1 score of this binary classification task. We compare RouterInterp's explanations against an n-gram baseline: we take token n-grams (unigrams or bigrams) that most frequently co-occur with each expert and prompt an LLM to summarise them into a natural language description.

4 Results

Table 1 shows results for gpt-oss-20b, where each expert's explanation aggregates explanations of the 10 SAE features with highest ρ-usefulness scores for that expert. RouterInterp achieves 0.60–0.64 across layers, substantially outperforming the explanations generated from bigram-expert co-occurrence frequencies (bigram baseline: 0.23–0.47). We illustrate examples of RouterInterp's explanations in Section G. To test the SSH, we also evaluate whether SAE latents can predict expert routing well. We extract model activations and encode them with an SAE encoder to produce sparse feature representations. We train a linear classifier on these sparse representations to predict whether a given expert is activated for some context. On OLMoE-1B-7B and gpt-oss-20b, the SAE-based predictor achieves 0.89 and 0.81 Recall respectively, substantially outperforming unigram and bigram baselines (0.52–0.74). These results support our hypothesis that routing is determined by features rather than surface-level token statistics. Setup, metric definition, and results across layers are in Section C.

5 Conclusion

We introduced *RouterInterp*, a method that interprets MoE routing decisions by identifying sparse autoencoder features most predictive of expert selection and aggregating their explanations into unified natural language descriptions. RouterInterp's explanations achieve 0.62 explanation score compared to 0.35 for token-expert co-occurrence methods. RouterInterp emerged from formalizing two alternative specialisation hypotheses: the *Domain Specialisation Hypothesis* (DSH), which posits that experts specialize in semantically coherent domains, and the *Superposed Specialisation Hypothesis* (SSH), which models experts as responding to a superposition of distinct features spanning multiple unrelated micro-domains. Our experiments provide initial evidence for the SSH: High quality (81%/89% recall) of SAE-based routing prediction shows that experts are highly polysemantic, with the set of features most aligned with any single expert often spanning disjoint concepts. We acknowledge the following limitations of our method: (1) RouterInterp is downstream of SAE quality (whether features are monosemantic) and of the quality of feature-level explanations, and (2) textual summaries struggle to be both concise and complete as more features are included, so interactive presentations may better serve expert interpretation. We are excited to improve upon these issues in future work.

References

Samira Abnar, Harshay Shah, Dan Busbridge, Alaaeldin Mohamed Elnouby Ali, Josh Susskind, and Vimal Thilak. Parameters vs flops: Scaling laws for optimal sparsity for mixture-of-experts language models. In *ICLR Workshop, ICML*, 2025. URL https://arxiv.org/abs/2501.12370.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Anthropic. System card: Claude haiku 4.5, October 2025. URL https://www.anthropic.com/claude-haiku-4-5-system-card.

Kola Ayonrinde. Awesome adaptive computations, 2023. URL https://github.com/koayon/awesome-adaptive-computation/.

Kola Ayonrinde, Michael T Pearce, and Lee Sharkey. Interpretability as compression: Reconsidering SAE explanations of neural activations. In *NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning*, 2024. URL https://openreview.net/forum?id=hAqeEZRVSD.

Andrea Banino, Jan Balaguer, and Charles Blundell. Pondernet: Learning to ponder. *arXiv preprint arXiv:2107.05407*, 2021.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Leonard Bereska and Stratis Gavves. Mechanistic interpretability for AI safety - a review. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=ePUVetPKu6. Survey Certification, Expert Certification.

Dan Braun, Lucius Bushnaq, and Lee Sharkey. Stochastic parameter decomposition. In *Submitted to ILIAD 2: ODYSSEY*, 2025. URL https://openreview.net/forum?id=dEdS9ao8gN. under review.

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.

Lucius Bushnaq, Stefan Heimersheim, Nicholas Goldowsky-Dill, Dan Braun, Jake Mendel, Kaarel Hänni, Avery Griffin, Jörn Stöhler, Magdalena Wache, and Marius Hobbhahn. The local interaction basis: Identifying computationally-relevant and sparsely interacting features in neural networks. *arXiv preprint arXiv:2405.10928*, 2024.

Weilin Cai, Juyong Jiang, Fan Wang, Jing Tang, Sunghun Kim, and Jiayi Huang. A survey on mixture of experts in large language models. *IEEE Transactions on Knowledge and Data Engineering*, 2025.

Marmik Chaudhari, Jeremi Nuer, and Rome Thorstenson. Superposition in mixture of experts. In *Mechanistic Interpretability Workshop at NeurIPS 2025*, 2025. URL https://openreview.net/forum?id=bZqopmfZDE.

Jeff Clune, Jean-Baptiste Mouret, and Hod Lipson. The evolutionary origins of modularity. *Proceedings of the Royal Society b: Biological sciences*, 280(1755):20122863, 2013.

Valérie Costa, Thomas Fel, Ekdeep Singh Lubana, Bahareh Tolooshams, and Demba Ba. From flat to hierarchical: Extracting sparse representations with matching pursuit. *arXiv preprint arXiv:2506.03093*, 2025.

Róbert Csordás, Christopher Potts, Christopher D Manning, and Atticus Geiger. Recurrent neural networks learn to store and generate sequences using non-linear representations. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen (eds.), *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 248–262, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi:10.18653/v1/2024.blackboxnlp-1.17. URL https://aclanthology.org/2024.blackboxnlp-1.17/.

Hoagy Cunningham, Aidan Ewart, Logan Riggs Smith, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=F76bwRSLeK.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. Glam: Efficient scaling of language models with mixture-of-experts. In *International conference on machine learning*, pp. 5547–5569. PMLR, 2022.

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.

Joshua Engels, Eric J Michaud, Isaac Liao, Wes Gurnee, and Max Tegmark. Not all language model features are one-dimensionally linear. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=d63a4AM4hb.

William Fedus, Jeff Dean, and Barret Zoph. A review of sparse expert models in deep learning. *arXiv preprint arXiv:2209.01667*, 2022a.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022b.

Daniel Filan, Stephen Casper, Shlomi Hod, Cody Wild, Andrew Critch, and Stuart Russell. Clusterability in neural networks. *arXiv preprint arXiv:2103.03386*, 2021.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling, 2020. URL https://arxiv.org/abs/2101.00027.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tcsZt9ZNKD.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(11):7436–7456, 2021.

Kaarel Hänni, Jake Mendel, Dmitry Vaintrob, and Lawrence Chan. Mathematical models of computation in superposition. In *ICML 2024 Workshop on Mechanistic Interpretability*, 2024.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/e2c420d928d4bf8ce0ff2ec19b371514-Paper.pdf.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991. doi:10.1162/neco.1991.3.1.79.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

Jakub Krajewski, Jan Ludziejewski, Kamil Adamczewski, Maciej Pióro, Michal Krutul, Szymon Antoniak, Kamil Ciebiera, Krystian Król, Tomasz Odrzygózdz, Piotr Sankowski, Marek Cygan, and Sebastian Jaszczur. Scaling laws for fine-grained mixture of experts. *CoRR*, abs/2402.07871, 2024. URL https://doi.org/10.48550/arXiv.2402.07871.

Edmund Lau, Zach Furman, George Wang, Daniel Murfet, and Susan Wei. The local learning coefficient: A singularity-aware complexity measure. In Yingzhen Li, Stephan Mandt, Shipra Agrawal, and Emtiyaz Khan (eds.), *Proceedings of The 28th International Conference on Artificial Intelligence and Statistics*, volume 258 of *Proceedings of Machine Learning Research*, pp. 244–252. PMLR, 03–05 May 2025. URL https://proceedings.mlr.press/v258/lau25a.html.

Mike Lewis, Shruti Bhosale, Tim Dettmers, Naman Goyal, and Luke Zettlemoyer. Base layers: Simplifying training of large, sparse models. In *International Conference on Machine Learning*, pp. 6265–6274. PMLR, 2021.

Johnny Lin. Llama 3.3 70b, temporal feature analysis, featured research, gpt-oss-20b, and *a lot more*, November 2025. URL https://www.neuronpedia.org/blog/fall-update.

Linda Linsefors and Lucius Bushnaq. Circuits in superposition 2: Now with less wrong math, 2025a. URL https://www.lesswrong.com/posts/FWkZYQceEzL84tNej/circuits-in-superposition-2-now-with-less-wrong-math.

Linda Linsefors and Lucius Bushnaq. Rotations in superposition, 2025b. URL https://www.lesswrong.com/posts/LZ7YMPJueB6qjL24n/rotations-in-superposition.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Ekdeep Singh Lubana, Can Rager, Sai Sumedh R Hindupur, Valerie Costa, Greta Tuckute, Oam Patel, Sonia Krishna Murthy, Thomas Fel, Daniel Wurgaft, Eric J Bigelow, et al. Priors in time: Missing inductive biases for language model interpretability. *arXiv preprint arXiv:2511.01836*, 2025.

Alireza Makhzani and Brendan Frey. K-sparse autoencoders. *arXiv preprint arXiv:1312.5663*, 2013.

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. URL https://openreview.net/forum?id=I4e82CIDxv.

Callum McDougall, Avery, and Lucius Bushnaq. Theories of modularity in the biological literature, 2022. URL https://www.lesswrong.com/posts/JzTfKrgC7Lfz3zcwM/theories-of-modularity-in-the-biological-literature.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Evan Pete Walsh, Oyvind Tafjord, Nathan Lambert, Yuling Gu, Shane Arora, Akshita Bhagia, Dustin Schwenk, David Wadden, Alexander Wettig, Binyuan Hui, Tim Dettmers, Douwe Kiela, Ali Farhadi, Noah A. Smith, Pang Wei Koh, Amanpreet Singh, and Hannaneh Hajishirzi. OLMoe: Open mixture-of-experts language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=xXTkbTBmqq.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 16–30, 2023.

Adam Newgas. Compressed computation: Dense circuits in a toy model of the universal-AND problem. In *Mechanistic Interpretability Workshop at NeurIPS 2025*, 2025. URL https://openreview.net/forum?id=J3Kds2Rxov.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi:10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

Kiho Park, Yo Joong Choe, and Victor Veitch. The linear representation hypothesis and the geometry of large language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=UGpGkLzwpP.

Gonçalo Santos Paulo, Alex Troy Mallen, Caden Juang, and Nora Belrose. Automatically interpreting millions of features in large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=EemtbhJOXc.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL https://openreview.net/forum?id=n6SCkn2QaG.

Jonas Pfeiffer, Sebastian Ruder, Ivan Vulić, and Edoardo Maria Ponti. Modular deep learning. *arXiv preprint arXiv:2302.11529*, 2023.

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

L Sharkey. Sparsify: A mechanistic interpretability research agenda. In *AI Alignment Forum*, 2024.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*, 2017a.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017b. URL https://openreview.net/forum?id=B1ckMDqlg.

Lewis Smith. The 'strong'feature hypothesis could be wrong. In *AI Alignment Forum*, 2024.

Curt Tigges. Thread on 2025 goodfire moe interpretability hackathon. https://x.com/CurtTigges/status/1953877787552755890, Aug 2025. X (formerly Twitter) thread by @CurtTigges; accessed 2025-12-17.

George Wang, Jesse Hoogland, Stan van Wingerden, Zach Furman, and Daniel Murfet. Differentiation and specialization of attention heads via the refined local learning coefficient. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=SUc1UOWndp.

Martin Wattenberg and Fernanda Viégas. Relational composition in neural networks: A survey and call to action. In *ICML 2024 Workshop on Mechanistic Interpretability*, 2024. URL https://openreview.net/forum?id=zzCEiUIPk9.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a. URL https://arxiv.org/abs/2505.09388.

Liu Yang, Kangwook Lee, Robert D Nowak, and Dimitris Papailiopoulos. Looped transformers are better at learning learning algorithms. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=HHbRxoDTxE.

Xingyi Yang, Constantin Venhoff, Ashkan Khakzar, Christian Schroeder de Witt, Puneet K. Dokania, Adel Bibi, and Philip Torr. Mixture of experts made intrinsically interpretable. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=6QERrXMLP2.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*, 2022.

A Background

A.1 Sparse Autoencoders (SAE)

A fundamental challenge in interpreting LLM activations is *superposition*: models encode more features than the number of available dimensions, resulting in *polysemantic neurons* that activate for multiple unrelated concepts (Elhage et al., 2022). Sparse Autoencoders (SAEs) address this by mapping an activation vector $\boldsymbol{x} \in \mathbb{R}^N$ to a corresponding sparse latent representation $\boldsymbol{z} \in \mathbb{R}^F$.[2] Ideally, $\boldsymbol{z}$ should represent an "unpacking" of the compressed representations into monosemantic, single-concept latents referred to as *features* (Bricken et al., 2023; Cunningham et al., 2024).

The SAE architecture consists of an encoder that projects activations into a sparse latent space, and a decoder that reconstructs the original activation from the sparse latents:

$$\boldsymbol{z} = \sigma_s(\boldsymbol{W}_{\text{enc}}(\boldsymbol{x} - \boldsymbol{b}_{\text{pre}}) + \boldsymbol{b}_{\text{enc}}) \tag{2}$$

$$\hat{\boldsymbol{x}} = \boldsymbol{W}_{\text{dec}}\boldsymbol{z} + \boldsymbol{b}_{\text{pre}} \tag{3}$$

where $\boldsymbol{W}_{\text{enc}} \in \mathbb{R}^{N \times F}$, $\boldsymbol{W}_{\text{dec}} \in \mathbb{R}^{F \times N}$, and σ_s is a sparsifying activation function. We follow Gao et al. (2025)'s **Top-K SAE** approach, where $\sigma_s = \text{TopK}(\cdot, k)$ retains only the k largest activations per input and zeros out the rest, ensuring that the latent is sparse. The SAE is trained to optimise reconstruction fidelity $\mathcal{L} = \|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2^2$, ensuring that the original activation can be accurately reconstructed from the sparse latent representation $\boldsymbol{z}$.

A.2 Mixture of Experts (MoE)

In a standard MoE layer, the dense feed-forward network (FFN) is replaced by E parallel expert networks $\{\boldsymbol{E}_i\}_{i=1}^E$, each typically an FFN itself. A learned router network ($\boldsymbol{W}_r$) determines which experts process each token. Given an input token representation $\boldsymbol{x}$, the router computes routing logits $h(\boldsymbol{x}) = \boldsymbol{W}_r \cdot \boldsymbol{x}$ and converts them to a probability distribution over experts via softmax:

$$p_i(\boldsymbol{x}) = \frac{\exp(h(\boldsymbol{x})_i)}{\sum_{j=1}^E \exp(h(\boldsymbol{x})_j)} \tag{4}$$

To enforce sparsity, only the top-k experts with the highest probabilities are selected. Letting $\mathcal{T}$ denote the set of selected expert indices, the layer output is computed as the weighted sum of expert outputs:

$$\boldsymbol{y} = \sum_{i \in \mathcal{T}} p_i(\boldsymbol{x}) \cdot \boldsymbol{E}_i(\boldsymbol{x}) \tag{5}$$

A key challenge in MoE training is *load balancing*: without intervention, models tend to collapse to using only a few experts, leaving others undertrained (Shazeer et al., 2017b). This is addressed through auxiliary losses that encourage uniform expert utilization.

B Related Work

B.1 Sparse Feature Decompositions for Model Interpretability

SAEs rely on the Linear Representation Hypothesis (Mikolov et al., 2013; Bricken et al., 2023; Olah et al., 2020; Nanda et al., 2023; Park et al., 2024): the hypothesis that features in LMs are represented linearly (Bereska & Gavves, 2024). The LRH is often a controversial assumption within interpretability (Smith, 2024; Csordás et al., 2024). However, in our case of interpreting expert routing, routing is usually performed by a linear map and so only linearly accessible information can be important for routing. This makes SAEs uniquely well suited to our problem[3].

[2] Typically $F > N$, allowing the SAE to represent more features than the dimensionality of the original activation vector $\boldsymbol{x}$.

[3] Also note that the Tuned Lens (Belrose et al., 2023) also validates that linear probes on intermediate residual stream states capture meaningful structure.

SAEs have drawbacks, however. SAEs can struggle to capture logical hierarchical structure (Costa et al., 2025); they do not efficiently capture structures of multi-dimensional features (Engels et al., 2025); they do not generally capture relations across multiple tokens (Lubana et al., 2025); and the structure of Relational Composition (Wattenberg & Viégas, 2024) remains difficult for SAEs.

Most work interpreting representations to date has focused on the impact of intermediate representations on the output logits of a forward pass (Bricken et al., 2023) or on representations in a subsequent layer (Marks et al., 2025). We instead focus on how representations impact the behaviour of a subsequent routing layer.

Inspired by the apparent success of SAE-based explanations, Sharkey (2024) describes a framework for the continual improvement of MI explanations. The framework's three stages are: (1) *mathematical description* (breaking down the neural network into functional parts [4]), (2) *semantic description* (labelling each functional part in a way that is understandable to humans [5]) and (3) *validation* (using the semantic description to make predictions about model behaviour and evaluating these predictions). Here our combination of the trained SAEs and our Ansatz (conjecture) that router weights are linear combinations of weights provide our mathematical description of routing. We leverage AutoInterp (Paulo et al., 2025) to aid with semantic description and evaluation of the functional parts and the empirical success of this approach provides us with evidence for the Ansatz.

B.2 Mixture of Experts Interpretability

Shazeer et al. (2017b) designed the modern Sparse MoE layer to improve the efficiency and scalability of very large neural networks. Other researchers, however, hoped that the specialisation routing provided might also extend some interpretability benefits (Jacobs et al., 1991; Fedus et al., 2022a). However, Jiang et al. (2024) find it difficult to obtain clean interpretability results with their MoE model Mixtral, remarking "Surprisingly, we do not observe obvious patterns in the assignment of experts based on the topic [of the input text]." Lewis et al. (2021) and Zoph et al. (2022) analyse the unigram patterns of routing and find that the observed level of specialisation varies dramatically making interpretability difficult. Tigges (2025) report some success with unigram analysis of a few experts such as a 'business' expert. We show, however, that by using SAE features as a basis for explanation rather than tokens, we are able to achieve more accurate and concise explanations.

Yang et al. (2025b) suggest that with very large numbers of experts, routing can be made somewhat interpretable. However, their setting requires more experts than is compute optimal given scaling laws for sparse models (Abnar et al., 2025; Krajewski et al., 2024) or more experts than is typical in high performing open source MoE models (Liu et al., 2024; Yang et al., 2025a; Agarwal et al., 2025; Muennighoff et al., 2025).

Chaudhari et al. (2025) provide an alternative model for understanding the phenomena of superposition (Elhage et al., 2022) in MoE models. Chaudhari et al. (2025) find that individual experts exhibit greater monosemanticity than equivalent dense models (in a toy setting). Crucially, their analysis measures monosemanticity at the level of *expert weight matrices*, not at the level of *routing decisions*, which we focus on. We argue that even if individual experts represent their assigned features monosemantically, the routing decision itself may be polysemantic. The router partitions the input space such that features assigned to the same expert only compete with each other for representational capacity (Chaudhari et al., 2025). This creates an incentive to route *dissimilar* domains to the same expert: micro-domains that rarely co-occur can share an expert with little interference, enabling the expert to effectively perform Computation in Superposition (Hänni et al., 2024).

[4] where here the parts could be in terms of representations (Bricken et al., 2023) or computations (Braun et al., 2025)

[5] note that implicitly this formulation requires the Independent Additivity Principle (Ayonrinde et al., 2024) - if multiple parts are relevant at one time, then it must be the case that we can understand the whole simply from understanding the constituent parts

B.3 Adaptive Computation

Sparse MoE models are a type of Adaptive Computation model (Graves, 2016; Ayonrinde, 2023)[6]. Adaptive Computation models are typically either more FLOP or parameter efficient than dense models as they either use a subset of their parameters (for example sparse MoE models or Early Exit models (Banino et al., 2021)) or reuse parameters (for example looping models (Dehghani et al., 2018; Yang et al., 2024)) respectively. There has been little work on the interpretability of Adaptive Computation models and while our work focuses on interpreting routing in the sparse MoE layer, we would be excited about work generalising this approach to routing in early-exit models like Mixture of Depths (Raposo et al., 2024).

B.4 Modularity in Neural Networks

Many brain-inspired neural network architectures employ modularity (the organisation of a system into functional, sparsely connected subunits) as a core component (Pfeiffer et al., 2023), as modularity is believed to be one of the key properties that makes brains efficient and effective (Clune et al., 2013).

We might hypothesise that because neural networks generalise so well and have much lower effective dimensionality than the model dimension would suggest (Lau et al., 2025), we should expect the models to contain intrinsic modularity. That is to say that even though NNs look fully connected and highly entangled, perhaps there is some way of viewing computation such that the computation is in fact highly modularised (Bushnaq et al., 2024). However, efforts to find such modularity have proven difficult (Filan et al., 2021), partially because it is not clear in what form we should expect such modularity to appear.

One way to understand the modularity that neural networks naturally and implicitly form, is to enforce some modular computation and see how the neural networks react. Sparse MoE models have this kind of enforced modularity and we hope that in studying these models we can develop better tools for uncovering possible latent modularity in dense neural networks. In particular, one promising path for understanding modularity may be through the optimisation pressures that resulted in such modularity. For example specialisation and reducing connection costs are two evolutionary pressures for modularity to develop in biological models (Clune et al., 2013; McDougall et al., 2022). Analogously, we might expect that specialisation pressures in the training processes for AI systems also results in the development of modularity as in Wang et al. (2025).

C SAE features predict MoE routing

Beside generating explanations, we also evaluate whether SAE features can linearly predict routing patterns. We extract activations $\boldsymbol{x}$ from the residual stream before the routing function and use an SAE encoder to produce sparse feature representations $\boldsymbol{z}$. We then train a logistic regression classifier on these sparse feature representations, one per expert E_i, to predict whether a given expert is activated for some context. We compare to n-gram baselines: for each token in the vocabulary, we count how often it co-occurs with each expert's activation on a training set. At inference, we predict that a token routes to the experts it most frequently co-occurred with.

We report *Recall* as our main evaluation metric, measuring the proportion of experts that are correctly predicted as being routed to. [7] For a token t with actual expert set $E_{\text{actual}}(t)$ and predicted expert set $E_{\text{pred}}(t)$, both of size k:

$$\text{Recall} = \frac{1}{N}\sum_{t=1}^{N}\frac{|E_{\text{pred}}(t) \cap E_{\text{actual}}(t)|}{|E_{\text{actual}}(t)|} \tag{6}$$

where N is the total number of tokens.

[6] Also known as Conditional or Dynamic Computation models (Han et al., 2021)

[7] Since both the predicted set $E_{\text{pred}}(t)$ and actual set $E_{\text{actual}}(t)$ have cardinality k by construction (due to top-k routing), Recall equals Precision and F1 in this setting (equal set sizes imply FP = $k - |E_{\text{pred}} \cap E_{\text{actual}}|$ = FN).

	OLMoE-1B-7B	gpt-oss-20b
Unigram Baseline	0.70	0.52
Bigram Baseline	0.74	0.55
SAE Predictor (Ours)	**0.89**	**0.81**

Table 2: Token-level statistics are insufficient to predict MoE routing: richer representations are required. N-gram baselines, which predict routing from token co-occurrence frequencies, achieve only 0.52–0.74 Recall, while our SAE-based predictor achieves 0.81–0.89 by leveraging learned feature representations. The performance gap widens on the larger model (gpt-oss-20b), suggesting routing becomes increasingly context-dependent at scale. This motivates using SAE features, rather than tokens, as the basis for interpreting expert routing. Results reported as Recall for layer 11 of OLMoE-1B-7B and layer 16 of gpt-oss-20b.

	Layer 4	Layer 8	Layer 12	Layer 16	Layer 20
Unigram Baseline	0.57	0.60	0.52	0.57	0.55
Bigram Baseline	0.63	0.64	0.56	0.59	0.59
SAE Predictor (Ours)	**0.82**	**0.85**	**0.79**	**0.82**	**0.83**

Table 3: The advantage of SAE-based prediction over token co-occurrence frequencies holds across all network depths. SAE-based prediction achieves 0.79–0.85 Recall across layers 4–20 of gpt-oss-20b, while n-gram baselines plateau at 0.52–0.64. The narrow range of SAE predictor performance suggests routing complexity is similar across layers.

We evaluate whether SAE latents can predict MoE routing decisions across different model architectures. Table 2 compares our SAE-based predictor against n-gram baselines on both OLMoE-1B-7B and gpt-oss-20b. The SAE predictor achieves 0.89 and 0.81 Recall respectively, substantially outperforming both baselines across both architectures. N-gram baselines perform markedly worse on the larger model (0.74 for OLMoE vs 0.55 for gpt-oss), suggesting that routing decisions in larger models are more context-dependent and cannot be predicted from surface-level token statistics alone. In contrast, the SAE predictor maintains strong performance across both architectures, demonstrating that routing information is consistently encoded in SAE features regardless of model size, architecture, or training distribution.

Table 3 shows that the advantage of SAE-based prediction holds across all layers of gpt-oss-20b. The consistent advantage at both early and late layers indicates that routing is better explained by SAE latents than by token statistics, supporting our hypothesis that routing operates at the feature level. We provide a detailed ablation on the number of active SAE latents m needed to outperform n-gram baselines in Section D.

D Routing Depends on Multiple Features

In Section C, we reported routing prediction results using $m = 32$ active features. Here we ablate the effect of the number of active features m on routing prediction Recall. Table 4 shows Recall for varying $m \in \{1, 2, 4, 8, 16, 32, 64\}$ across five layers of gpt-oss-20b.

The results reveal two key patterns. First, Recall continues to improve with more features across all layers, indicating that routing decisions depend on multiple distinct concepts —consistent with polysemantic expert behaviour. Second, even a single feature ($m = 1$) outperforms n-gram baselines at most layers (underlined), demonstrating that individual SAE features carry substantial routing information. This aligns with our finding that ρ-useful features achieve strong explanation performance with just 1–2 features (Appendix F).

	Layer 4	Layer 8	Layer 12	Layer 16	Layer 20
Unigram Baseline	0.570	0.602	0.523	0.573	0.553
Bigram Baseline	0.631	0.639	0.560	0.593	0.589
SAE Predictor ($m = 1$)	<u>0.690</u>	0.462	<u>0.597</u>	<u>0.687</u>	<u>0.748</u>
SAE Predictor ($m = 4$)	0.790	<u>0.809</u>	0.754	0.811	0.847
SAE Predictor ($m = 8$)	0.803	0.824	**0.811**	0.800	0.844
SAE Predictor ($m = 16$)	0.817	0.830	0.810	**0.830**	0.861
SAE Predictor ($m = 32$)	**0.824**	**0.865**	0.801	0.820	0.866
SAE Predictor ($m = 64$)	0.822	0.858	0.793	0.825	**0.873**

Table 4: Routing prediction improves with more SAE features, consistent with polysemantic expert behaviour predicted by the SSH. <u>Underlined</u> values outperform n-gram baselines; **bold** marks best Recall with fewest features. Notably, even a single feature ($m = 1$) outperforms n-gram baselines at 4 of 5 layers, while performance continues improving up to $m = 32$–64, indicating that routing depends on multiple distinct concepts rather than a single domain.

E Experimental setup

Models We evaluate on two MoE architectures: OLMoE-1B-7B (Muennighoff et al., 2025) and gpt-oss-20b (Agarwal et al., 2025).

Datasets. For experiments with OLMoE, we use OLMoE-mix-0924 (Muennighoff et al., 2025) for SAE training and routing prediction experiments. For gpt-oss, we use a following mixture: 75% gpt-oss-generated continuations from The Pile (Gao et al., 2020) prompts[8] and 25% FineWeb (Penedo et al., 2024). We use The Pile (Gao et al., 2020) as a dataset for explanation scoring.

SAEs. For OLMoE, we train Top-K SAEs for 100M tokens on activations sampled from layers 3, 7, 11, and 15 (out of OLMoE's 16 layers). Our SAEs have $k = 32$ and 32,768 features (an expansion of 16x compared to the residual stream size). For gpt-oss (24 layers), we use trained BatchTopK SAEs[9] (Lin, 2025) on layers 4, 8, 12, 16, and 20, with 131,072 features and $k = 64$.

Routing Prediction. Linear classifiers are trained on SAE features from 1M tokens. We vary the number of active features m to assess how routing information distributes across features. Specifically, for OLMoE we use $m \in \{1, 2, 4, 8, 16, 32\}$, while for gpt-oss we use $m \in \{1, 2, 4, 8, 16, 32, 64\}$.

We compare our SAE-based classifiers to unigram and bigram models that predict routing based on n-gram frequencies in training data. Concretely, for each n-gram (unigram or bigram) in the train set, we collect the number of times it activates each expert. We then predict the activated experts by taking the top-k experts with the highest activation frequency for a given n-gram.

Expert Interpretation. For gpt-oss, we generate explanations across layers using the top-10 ρ-useful features per expert. We use the Delphi library (Paulo et al., 2025) with Claude Haiku 4.5 (Anthropic, 2025) for explanation generation, summarisation, and calculating explanation scores. We use the all-MiniLM-L6-v2 model from Sentence-Transformers library (Reimers & Gurevych, 2019), for calcuting embeddings for samples in the evaluation set for explanation scoring,

[8] `https://huggingface.co/datasets/andyrdt/gpt-oss-20b-rollouts`
[9] `https://huggingface.co/andyrdt/saes-gpt-oss-20b`

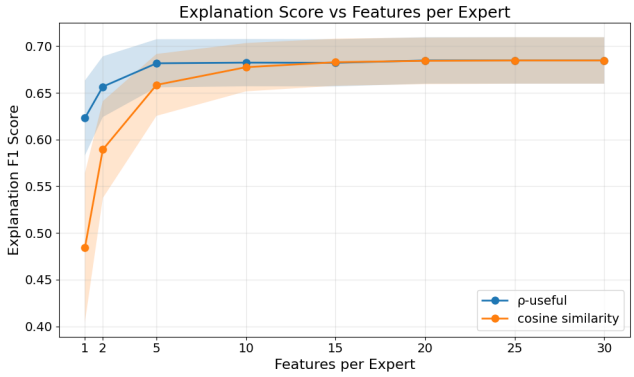

Figure 3: The most useful features (ranked by the ρ-usefulness metric) are individually *more* predictive of expert routing than features with highest cosine similarity with router weights. Despite selecting largely non-overlapping feature sets, both methods converge to similar performance at 10 features, suggesting routing information is redundantly encoded across many SAE features. Results are reported for layer 16 of gpt-oss-20b. Shaded regions show standard deviation across experts.

F Feature Selection Method and Set Size

In Section 3, we describe selecting the top-n features by ρ-usefulness for each expert. Here we justify this choice by evaluating (1) how the feature set size n affects explanation quality, and (2) how ρ-usefulness compares to an alternative selection method based on cosine similarity.

As an alternative to ρ-usefulness, we consider selecting features based on the geometric alignment between SAE decoder directions $\boldsymbol{d}_f \in W_{dec}$ and router weight vectors $\boldsymbol{w}_k \in \boldsymbol{W}_r$. For each expert E_k, we rank features f by $\cos(\boldsymbol{d}_f, \boldsymbol{w}_k)$ and select the top-n.

To isolate the effect of feature selection, we replace the LLM summarisation with a simpler aggregation: we evaluate each feature's explanation independently and label an example as positive if *at least one* feature's explanation predicts expert activation. This setup directly measures whether adding more features increases predictive coverage: as the feature set grows, we track whether additional features contribute new predictive information.

Figure 3 shows that individual ρ-useful features carry strong predictive signal— even a single feature achieves F1 ≈ 0.62. In contrast, features with high cosine similarity to router weights are only predictive in aggregate (F1 ≈ 0.49 at $n = 1$), suggesting that geometric alignment with router weights does not guarantee a feature fires when that expert is selected. Both methods converge to approximately the same explanation score (≈ 0.68) at 10 features.

The convergence at 10 features is interesting given that the two criteria select largely non-overlapping feature sets. This suggests that routing-relevant information is encoded across many SAE features, and with enough features, both methods eventually capture sufficient signal to predict routing.

Based on these results, we use $n = 10$ features per expert throughout our experiments.

G Example Expert Explanations

Table 5 shows different levels of RouterInterp explanations for a randomly selected expert (Expert 8) from layer 16 of gpt-oss-20b. It illustrates how expert-level explanations aggregate multiple feature-level explanations, which are in turn acquired by explaining examples where the feature activates.

Level	Explanation
Expert 8	This expert activates across five distinct contexts: (1) Structural and grammatical elements—function words, articles, prepositions, conjunctions, and punctuation that organize text and parse grammatical relationships. (2) Informational and technical content—medical/clinical terminology, dates, measurements, quantitative data, multilingual technical terms, and metadata markers, with stronger activation in content-dense, semantically specific text. (3) Logical and reasoning anchors—punctuation marking boundaries, conjunctions signaling relationships, mathematical operators, and discourse markers in technical or argumentative contexts. (4) Self-identification phrase—high activation on tokens within "a large language model trained by OpenAI" in system messages. (5) Mathematical and technical notation—numerical values, operators, variables, LaTeX notation, and domain-specific keywords encoding mathematical constraints and relationships.
Feature 46328	*Self-identification phrase—tokens within "a large language model trained by OpenAI" in system messages, particularly " large," " model," " trained," and "AI."* **Activating Examples:** [1] Chat GPT , a large language model trained by Open AI .
Feature 70455	*Logical and reasoning anchors—punctuation marking boundaries, conjunctions signaling relationships, mathematical operators, and discourse markers in technical or argumentative contexts.* **Activating Examples:** [1] a GUI (like Chess Base , Arena , Stockfish 's own GUI , or
Feature 73639	*Structural and grammatical elements—articles, prepositions, conjunctions, punctuation, and mathematical delimiters that organize and parse text.* **Activating Examples:** [1] and $ y $. Let $ n $ be the number of possible values of [2] s 393 56 ash . Maybe it's best to ignore
Feature 81948	*Informational and technical content—medical/clinical terms, dates, measurements, quantitative data, multilingual technical terms, and metadata markers. Higher activation in content-dense, semantically specific text.* **Activating Examples:** [1] though he termed it " spec ulative ." He acknowledged [2] I 've never been through anything like this before .
Feature 91358	*Mathematical and technical notation—numerical values, operators, variables, LaTeX notation, and domain-specific keywords encoding mathematical constraints and relationships.* **Activating Examples:** [1] But it's given that he secured 110 marks . So 170 - T [2] 2 - y _ 1 } + \{} frac { y _ 2 ^ 2 }{

Table 5: RouterInterp first explains features by looking at examples where these features activate strongly, and then aggregates these explanations into an expert-level explanation. For example, Expert 8 from gpt-oss-20b layer 16 specialises in AI self-identification (*Feature 46328*), medical terms (*Feature 81948*), mathematical notation (*Feature 91358* and *Feature 73639*), and reasoning anchors (*Feature 70455*). Here we show explanations for the top 5 ρ-useful SAE features (rather than 10) for brevity.