POLYSEMANTIC EXPERTS, MONOSEMANTIC PATHS: ROUTING AS CONTROL IN MOES

Charles Ye[*]
Independent
dogdynamics@proton.me

Bo Yuan
Georgia Institute of Technology
byuan48@gatech.edu

Lee Sharkey
Goodfire
lee@goodfire.ai

ABSTRACT

An LLM's residual stream is both **state** and **instruction**: it encodes the current context and determines the next transformation. We introduce a parameter-free decomposition for Mixture-of-Experts models that splits each layer's hidden state into a **control** signal that causally drives routing and an orthogonal **content** channel invisible to the router. Across six MoE architectures, we find that models preserve surface-level features (language, token identity, position) in the content channel, while the control signal encodes an abstract function that rotates from layer to layer. Because each routing decision is low-bandwidth, this rotation forces compositional specialization across layers. While individual experts remain polysemantic, **expert paths become monosemantic**, clustering tokens by semantic function across languages and surface forms. The same token (e.g., ":") follows distinct trajectories depending on whether it serves as a type annotation, an introductory colon, or a time separator. Our decomposition identifies the source of this structure: clusters in the control subspace are substantially more monosemantic than those in the full representation. As a result, the natural unit of interpretability in MoEs is not the expert but the trajectory.

1 INTRODUCTION

Mixture-of-Experts (MoE) architectures have become the dominant paradigm for frontier language models[1]. While the mechanism is architecturally simple, how MoEs organize computation remains poorly understood.

The intuitive model of a "committee of specialists" where each expert masters a distinct domain has been largely invalidated (Jiang et al., 2024; Xue et al., 2024). Instead, experts are *polysemantic*, activating for unrelated concepts. Prior work finds varied and often weak correlations between experts and interpretable categories: semantic domains (Olson et al., 2025), POS (Antoine et al., 2024), or sequence position (Bershatsky & Oseledets, 2025). Such disparate findings suggest that individual experts may simply be the wrong unit of analysis.

Tracking which expert a token visits at a single layer yields little structure. But we find that tracking the *sequence* of experts across layers tells a different story. Figure 1 shows the routing paths of 500 instances of the token `:` in three distinct syntactic roles[2]. Despite identical surface form, the three uses follow visibly distinct trajectories: coherent bundles that diverge based on contextual function, not token identity. Individual experts are polysemantic; expert **paths** are not.

What mechanism produces this structure, and why has prior work missed it? Existing analyses primarily study correlations between routing and external variables, without isolating what *causally drives* expert selection. Routing is fully determined by $R_l^\top h_l$, the product of a routing matrix and the hidden state h_l. Since h_l encodes language, syntax, and semantics (Tenney et al., 2019), correlations between routing and such variables are unsurprising. But this linearity is not just a limitation—it is an opportunity.

[*]Corresponding author.

[1]At time of writing, all top-10 models with disclosed architectures on ARTIFICIALANALYSIS (Artificial Analysis, 2025) are MoEs.

[2]Sequences are generated for balanced category representation; samples in Section G.

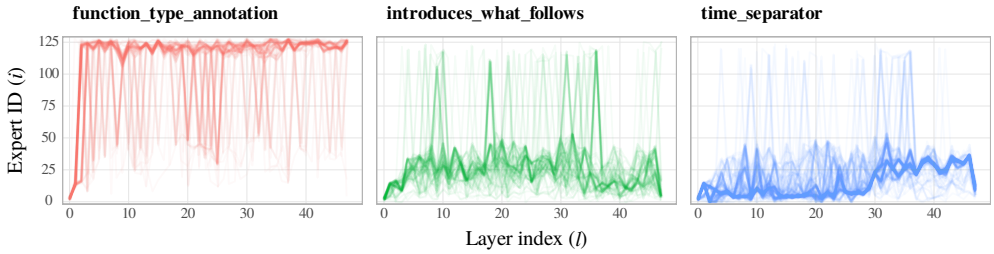

Figure 1: **One token, three programs.** We plot 500 top-1 routing paths of the token `:` through 48 MoE layers of QWEN3-30B-A3B, separated by contextual function: Python type annotation (*left*), introductory colon (*center*), time separator (*right*). Despite identical token ID, the three uses produce distinct expert trajectories. Expert IDs are reordered at each layer using a shared Sugiyama-style layout to minimize edge crossings (Section A). The same expert ordering is used for all panels.

An LLM's residual stream is both **state** and **instruction**: it encodes the context being processed, and it determines what computation is applied next. In dense models these roles are entangled. MoEs make them separable: because routing is a linear read of h_l, we can derive exactly which components causally influence expert selection (the **control** signal) and which are invisible to the router (the **content** channel). This decomposition lets us trace how MoEs organize computation, from the geometry of routing to the monosemantic paths of Figure 1.

Our contributions:

- **A causal decomposition of the residual stream** (Section 2). We derive a parameter-free decomposition that separates each layer's hidden state into orthogonal *control* (router-visible) and *content* (router-blind) channels. This enables direct analysis of what information causally drives expert selection versus what is merely carried through.
- **A rotating computational strategy** (Section 3). Across 6 models, we find a shared strategy: routing is dominated by a tiny subspace of the hidden state that rotates from layer to layer. This rotation mechanism forces compositional specialization.
- **Expert paths are monosemantic** (Section 4). While individual experts remain polysemantic, multi-layer expert paths cluster tokens by semantic function across languages and surface forms. Control-subspace clusters group $4\times$ more unique tokens than content clusters at equal interpretability, confirming that the decomposition isolates the signal responsible for this structure.

2 Separating State from Instruction

Which features of the residual stream actually drive routing? Routing is a linear function of the residual stream: $s_l = R_l^\top h_l$. This linearity is crucial—it means the router can only access information that lies in the row space of its weight matrix R_l. Anything orthogonal to that subspace is invisible to routing, regardless of what it encodes. We exploit this to exactly separate each layer's residual stream h_l into two components: what the router can read, and what it cannot.

$$h_l = \underbrace{h_l^{\text{vis}}}_{\text{router-visible}} + \underbrace{h_l^{\text{blind}}}_{\text{router-blind}}, \qquad \langle h_l^{\text{vis}}, h_l^{\text{blind}} \rangle = 0.$$

The decomposition is straightforward: we compute the SVD of the routing matrix R_l and project h_l onto its row space to obtain h_l^{vis}; the remainder is h_l^{blind} (full details in Section B).

By construction, $R_l^T h_l^{\text{blind}} = 0$: the router-blind component **cannot** influence expert selection. This is not a statistical claim, but a mathematical guarantee. Figure 2 illustrates how the two components flow through the MoE layer.

We validate the decomposition empirically in Section 3, where it reveals a surprising structure: a control signal that rotates across layers while content accumulates.

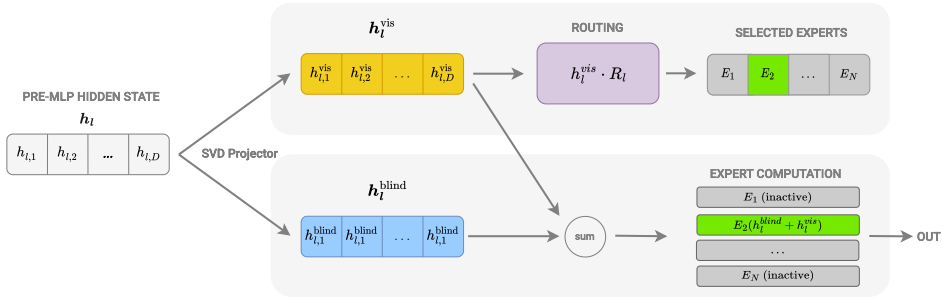

Figure 2: **Control and content in an MoE layer.** We decompose the residual stream h_l into orthogonal components via SVD of the routing matrix. The router-visible component (h_l^{vis}, top) is the *control* signal: it alone determines expert selection. The router-blind component (h_l^{blind}, bottom) is the *content*: invisible to routing, but processed by the selected expert alongside h_l^{vis}.

3 How MoEs Organize Computation

We apply our decomposition across six modern MoE architectures spanning 7B–106B total parameters, with 32–128 experts per layer and diverse designs including shared experts, hybrid attention, and varied sparsities. All analyses use ∼2m tokens sampled from C4 and HPLT (Section C details models and data). Across all models, we find the same computational motif.

Routers amplify, they don't integrate. How does a router distill a high-dimensional residual stream h_l into expert choices? One might expect it to integrate subtle signals distributed across the full representation.

Instead, routers converge on a simpler strategy: they amplify what is already loud. Across all models and layers, router weights concentrate on the dimensions of h_l that already carry the highest activation magnitude ($\rho \approx 0.6$; full analysis in Section D). This compresses the effective decision well below the router's capacity: probes trained on just the top **2%** of hidden dimensions (ranked by activation magnitude) predict the top-1 expert with **38–78%** accuracy; random subsets of the same size achieve **1–4%**. This compression has consequences: no single routing decision can express fine-grained specialization.

Control rotates; content accumulates. If routing is driven by a thin slice of h_l, what role does the rest play? Our decomposition lets us answer this directly, separating features that causally drive routing from those that cannot influence it[3].

We train probes to predict expert choice at the current layer $E(l)$ and the next layer $E(l+1)$ from each channel (Figure 3). The results reveal a clear information pipeline. h_l^{vis} predicts $E(l)$ near-perfectly (∼99%), validating the decomposition: the control signal is causally sufficient. But it contains almost no information about $E(l+1)$ (∼35%)—what drives routing at one layer becomes irrelevant at the next.

The pattern flips for content. h_l^{blind}, which *cannot* influence routing at layer l, is the strongest predictor of the next layer's expert choice (∼65–75%). This is the hand-off: **experts transform content to prepare the next control signal.** Control is continually re-derived from content. Cross-layer cosine similarity confirms the asymmetry: h_l^{vis} changes substantially between adjacent layers while h_l^{blind} remains stable across depth (Section E).

Routing bypasses surface features. What does the control signal actually encode? We probe both channels for language, token ID, and sequence position, features commonly studied in prior work on expert specialization.

Across all six models and every layer, these features reside primarily in h_l^{blind}, not h_l^{vis} (Section F). The disparity is largest at middle layers, where surface features are nearly absent from the control

[3] We probe the top-1 expert throughout.

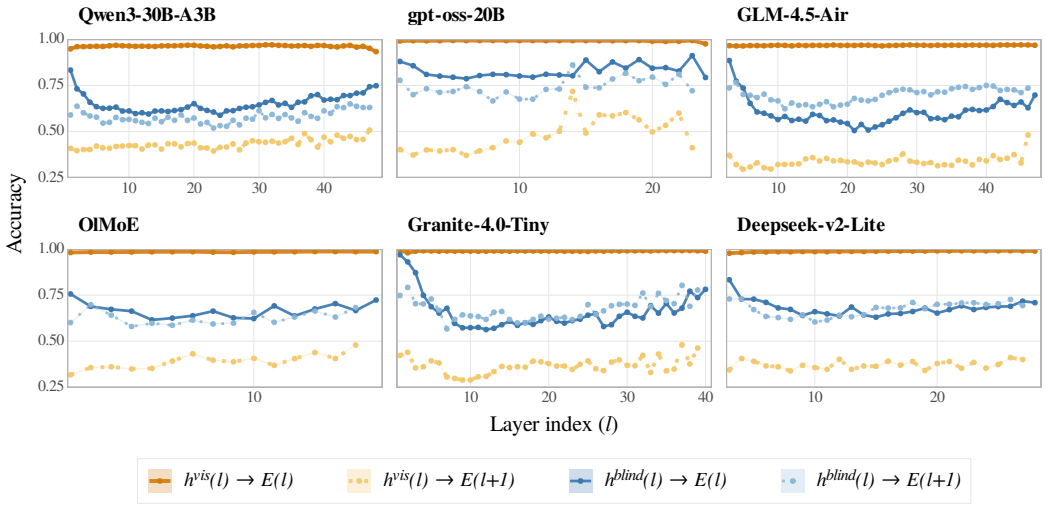

Figure 3: **Routing features are ephemeral.** Probes predict the top-1 expert at the current layer $E(l)$ or next layer $E(l+1)$ from each channel. The control signal h_l^{vis} predicts $E(l)$ near-perfectly (dark orange, $\sim$99%) but carries almost no information about $E(l+1)$ (light orange, $\sim$35%). The content channel h_l^{blind} shows the reverse: weak for the current layer (dark blue, $\sim$67%), it is the strongest predictor of the next layer's choice (light blue, $\sim$65%).

signal; they re-enter only at the earliest and final layers, where routing interfaces with token-level representations. Language, token identity, and position are carried in content but do not drive expert selection through the core of the network[4].

This resolves a puzzle from prior work: routing correlates only weakly with such features *because it is not reading them*. Individual routing decisions are coarse—each one collapses many tokens onto the same expert, discarding the detail that makes h_l rich. But because control rotates, each layer abstracts along a different axis. As we show next, these coarse decisions compose across layers into precise semantic signatures.

4 Expert Paths as Monosemantic Representations

If each routing decision is a coarse categorization, and each layer categorizes along a different axis, then sequences of routing decisions should compose into fine-grained semantic signatures. We test this by examining **expert paths**: for a contiguous band of L layers, each token's path is the tuple of L top-1 expert IDs it traverses. We group tokens sharing identical paths and examine the resulting clusters.

Paths cluster by meaning, not surface form. We collect paths over $\sim$10M multilingual tokens (EN/ZH/ES) from C4 and HPLT, using GPT-OSS-20B across mid-layers $l = 8, \ldots, 16$—the depth range where surface features are most absent from the control signal (Section 3).

The results are striking (Figure 4). Tokens sharing a path form tightly coherent semantic groups that cut across languages and surface forms. One path clusters exhibition-related tokens (展, 展览, `Exhibition`, `Showcase`); another groups `overseas`, 海外, and 境外 despite no lexical overlap. Paths for nutritional substances group `cholesterol`, `sugar`, 血脂, and `grasa` across three languages; paths for brand names group `Adidas`, 联想, and `McDonald`. These clusters are representative of thousands of similarly coherent groups, and reflect *what a token does in context*, not what it looks like—as previewed for the token `":"` in Figure 1, where identical surface form yields distinct paths for distinct functions.

[4]This may explain findings in Li & Zhou (2025), where routing-derived embeddings boost h_l-derived embeddings: routing strips away surface features, revealing core semantic data.

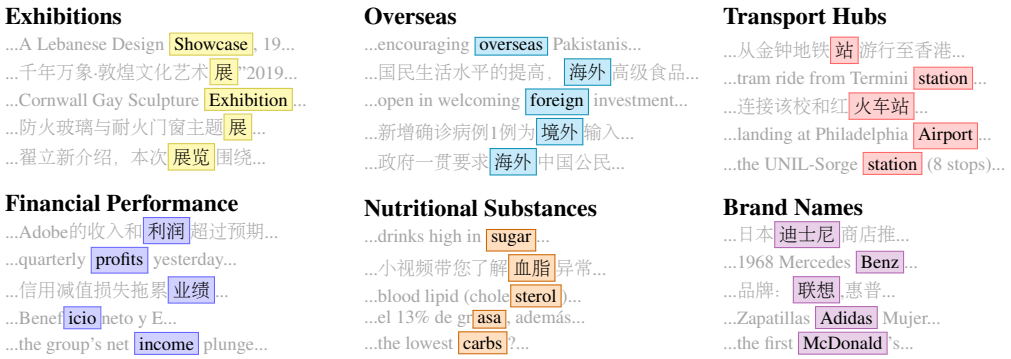

Figure 4: **Expert paths are monosemantic.** Each group shows tokens (highlighted) sharing an identical mid-layer expert path in `gpt-oss-20b`. Context is grayed; highlighted tokens are the routed unit. Paths cluster by function across languages and surface forms: overseas, 海外, and 境外 share no lexical overlap but follow the same computation trace.

The control subspace is the source. Our decomposition predicts this: if paths compose control signals, then clustering in the control subspace should recover semantic structure, while clustering in content should not. We test this by independently clustering tokens in h_l^{vis}, h_l^{blind}, and h_l at each layer (k-means, k=32 to match expert count), then grouping tokens that share the same cluster assignments across layers 8–16.[5] We sample 100 cross-layer clusters from each subspace, filter those rated uninterpretable by automated scoring (∼5%), and measure lexical diversity among the remainder.

All three subspaces produce interpretable clusters (≥95% pass screening). The distinction is *what organizes them*. h_l^{blind} clusters are coherent but trivially so: they consist almost entirely of repeated instances of the same token (1.1 unique token IDs per cluster, out of 10 token samples per cluster). h_l clusters show a similar pattern (1.4 unique IDs). h_l^{vis} clusters, by contrast, group diverse tokens by shared function (4.3 unique IDs)—4× the diversity of content clusters. Content preserves lexical identity; **control encodes semantic function**. The natural unit of interpretability in MoEs is not the expert—it is the trajectory.

5 Discussion

MoEs separate computation into two orthogonal streams: a low-rank control signal that selects experts, and a content channel that carries the payload. Control rotates across layers while content accumulates, with each expert transforming today's content into tomorrow's control signal. This hand-off forces compositional specialization, producing expert paths that cluster tokens by semantic function across languages and surface forms.

Implications for neural control. The control subspace is small, causally sufficient, and orthogonal to content—a natural target for intervention. Perturbing control could steer expert selection without corrupting content; modifying content could alter what is computed on while leaving routing unchanged. Whether this orthogonality survives into downstream behavior is an open question, but the decomposition provides the right coordinate system for asking it.

Broader perspective. Dense models entangle "what to compute" and "what to compute on" in a single residual stream. MoEs, by introducing discrete routing, make this separation recoverable—and our results suggest the models themselves exploit it, composing simple control signals into rich semantic structure. We suspect the state–instruction duality is not unique to MoEs: dense models face the same organizational pressure, but without a discrete routing decision to make the separation identifiable. The decomposition tools may differ, but the question generalizes: understanding how representations *control* computation, not just *encode* information, may be essential to interpreting modern neural networks broadly.

[5] Exact path matching applies only to h_l^{vis}; continuous clustering enables a fair comparison across subspaces.

REFERENCES

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily, Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Elie Antoine, Frédéric Béchet, and Philippe Langlais. Part-of-speech sensitivity of routers in mixture of experts models, 2024. URL https://arxiv.org/abs/2412.16971.

Artificial Analysis. Artificial Analysis: AI Model & API Providers Analysis. https://artificialanalysis.ai/, 2025. Independent AI benchmarking and analysis platform providing model intelligence, performance, cost comparisons, and industry insights.

Daniel Bershatsky and Ivan Oseledets. On the spatial structure of mixture-of-experts in transformers, 2025. URL https://arxiv.org/abs/2504.04444.

Laurie Burchell, Ona de Gibert, Nikolay Arefyev, Mikko Aulamo, Marta Bañón, Pinzhen Chen, Mariia Fedorova, Liane Guillou, Barry Haddow, Jan Hajič, Jindřich Helcl, Erik Henriksson, Mateusz Klimaszewski, Ville Komulainen, Andrey Kutuzov, Joona Kytöniemi, Veronika Laippala, Petter Mæhlum, Bhavitvya Malik, Farrokh Mehryary, Vladislav Mikhailov, Nikita Moghe, Amanda Myntti, Dayyán O'Brien, Stephan Oepen, Proyag Pal, Jousia Piha, Sampo Pyysalo, Gema Ramírez-Sánchez, David Samuel, Pavel Stepachev, Jörg Tiedemann, Dušan Variš, Tereza Vojtěchová, and Jaume Zaragoza-Bernabeu. An expanded massive multilingual dataset for high-performance language technologies (hplt), 2025. URL https://arxiv.org/abs/2503.10267.

DeepSeek-AI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Dengr, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jin Chen, Jingyang Yuan, Junjie Qiu, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruizhe Pan, Runxin Xu, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Size Zheng, T. Wang, Tian Pei, Tian Yuan, Tianyu Sun, W. L. Xiao, Wangding Zeng, Wei An, Wen Liu, Wenfeng Liang, Wenjun Gao, Wentao Zhang, X. Q. Li, Xiangyue Jin, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaojin Shen, Xiaokang Chen, Xiaosha Chen, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Liu, Xin Xie, Xingkai Yu, Xinnan Song, Xinyi Zhou, Xinyu Yang, Xuan Lu, Xuecheng Su, Y. Wu, Y. K. Li, Y. X. Wei, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Li, Yaohui Wang, Yi Zheng, Yichao Zhang, Yiliang Xiong, Yilong Zhao, Ying He, Ying Tang, Yishi Piao,

Yixin Dong, Yixuan Tan, Yiyuan Liu, Yongji Wang, Yongqiang Guo, Yuchen Zhu, Yuduan Wang, Yuheng Zou, Yukun Zha, Yunxian Ma, Yuting Yan, Yuxiang You, Yuxuan Liu, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhen Huang, Zhen Zhang, Zhenda Xie, Zhewen Hao, Zhihong Shao, Zhiniu Wen, Zhipeng Xu, Zhongyu Zhang, Zhuoshu Li, Zihan Wang, Zihui Gu, Zilin Li, and Ziwei Xie. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model, 2024. URL https://arxiv.org/abs/2405.04434.

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.

Ziyue Li and Tianyi Zhou. Your mixture-of-experts LLM is secretly an embedding model for free. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=eFGQ97z5Cd.

Mayank Mishra, Matt Stallone, Gaoyuan Zhang, Yikang Shen, Aditya Prasad, Adriana Meza Soria, Michele Merler, Parameswaran Selvam, Saptha Surendran, Shivdeep Singh, Manish Sethi, Xuan-Hong Dang, Pengyuan Li, Kun-Lung Wu, Syed Zawad, Andrew Coleman, Matthew White, Mark Lewis, Raju Pavuluri, Yan Koyfman, Boris Lublinsky, Maximilien de Bayser, Ibrahim Abdelaziz, Kinjal Basu, Mayank Agarwal, Yi Zhou, Chris Johnson, Aanchal Goyal, Hima Patel, Yousaf Shah, Petros Zerfos, Heiko Ludwig, Asim Munawar, Maxwell Crouse, Pavan Kapanipathi, Shweta Salaria, Bob Calio, Sophia Wen, Seetharami Seelam, Brian Belgodere, Carlos Fonseca, Amith Singhee, Nirmit Desai, David D. Cox, Ruchir Puri, and Rameswar Panda. Granite code models: A family of open foundation models for code intelligence, 2025. URL https://arxiv.org/abs/2405.04324.

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, 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, 2025. URL https://arxiv.org/abs/2409.02060.

Matthew Lyle Olson, Neale Ratzlaff, Musashi Hinck, Man Luo, Sungduk Yu, Chendi Xue, and Vasudev Lal. Probing semantic routing in large mixture-of-expert models, 2025. URL https://arxiv.org/abs/2502.10928.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

Kozo Sugiyama, Shojiro Tagawa, and Mitsuhiko Toda. Methods for visual understanding of hierarchical system structures. *IEEE Transactions on Systems, Man, and Cybernetics*, 11(2):109–125, 1981. doi: 10.1109/TSMC.1981.4308636.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452/.

Fuzhao Xue, Zian Zheng, Yao Fu, Jinjie Ni, Zangwei Zheng, Wangchunshu Zhou, and Yang You. Openmoe: An early effort on open mixture-of-experts language models. *arXiv preprint arXiv:2402.01739*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger

Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, Kedong Wang, Lucen Zhong, Mingdao Liu, Rui Lu, Shulin Cao, Xiaohan Zhang, Xuancheng Huang, Yao Wei, Yean Cheng, Yifan An, Yilin Niu, Yuanhao Wen, Yushi Bai, Zhengxiao Du, Zihan Wang, Zilin Zhu, Bohan Zhang, Bosi Wen, Bowen Wu, Bowen Xu, Can Huang, Casey Zhao, Changpeng Cai, Chao Yu, Chen Li, Chendi Ge, Chenghua Huang, Chenhui Zhang, Chenxi Xu, Chenzheng Zhu, Chuang Li, Congfeng Yin, Daoyan Lin, Dayong Yang, Dazhi Jiang, Ding Ai, Erle Zhu, Fei Wang, Gengzheng Pan, Guo Wang, Hailong Sun, Haitao Li, Haiyang Li, Haiyi Hu, Hanyu Zhang, Hao Peng, Hao Tai, Haoke Zhang, Haoran Wang, Haoyu Yang, He Liu, He Zhao, Hongwei Liu, Hongxi Yan, Huan Liu, Huilong Chen, Ji Li, Jiajing Zhao, Jiamin Ren, Jian Jiao, Jiani Zhao, Jianyang Yan, Jiaqi Wang, Jiayi Gui, Jiayue Zhao, Jie Liu, Jijie Li, Jing Li, Jing Lu, Jingsen Wang, Jingwei Yuan, Jingxuan Li, Jingzhao Du, Jinhua Du, Jinxin Liu, Junkai Zhi, Junli Gao, Ke Wang, Lekang Yang, Liang Xu, Lin Fan, Lindong Wu, Lintao Ding, Lu Wang, Man Zhang, Minghao Li, Minghuan Xu, Mingming Zhao, Mingshu Zhai, Pengfan Du, Qian Dong, Shangde Lei, Shangqing Tu, Shangtong Yang, Shaoyou Lu, Shijie Li, Shuang Li, Shuang-Li, Shuxun Yang, Sibo Yi, Tianshu Yu, Wei Tian, Weihan Wang, Wenbo Yu, Weng Lam Tam, Wenjie Liang, Wentao Liu, Xiao Wang, Xiaohan Jia, Xiaotao Gu, Xiaoying Ling, Xin Wang, Xing Fan, Xingru Pan, Xinyuan Zhang, Xinze Zhang, Xiuqing Fu, Xunkai Zhang, Yabo Xu, Yandong Wu, Yida Lu, Yidong Wang, Yilin Zhou, Yiming Pan, Ying Zhang, Yingli Wang, Yingru Li, Yinpei Su, Yipeng Geng, Yitong Zhu, Yongkun Yang, Yuhang Li, Yuhao Wu, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yuxuan Zhang, Zezhen Liu, Zhen Yang, Zhengda Zhou, Zhongpei Qiao, Zhuoer Feng, Zhuorui Liu, Zichen Zhang, Zihan Wang, Zijun Yao, Zikang Wang, Ziqiang Liu, Ziwei Chai, Zixuan Li, Zuodong Zhao, Wenguang Chen, Jidong Zhai, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models, 2025. URL https://arxiv.org/abs/2508.06471.

A Routing Path Visualization

We visualize expert paths as a layered directed acyclic graph, where nodes are experts at each layer and weighted edges represent token transitions between adjacent layers. The primary challenge is edge crossing: raw expert IDs are arbitrary, so even coherent routing structure appears as unintelligible spaghetti-plots when plotted with default ordering. We believe this visualization barrier is a key reason multi-layer routing paths have been underexplored despite their interpretive value.

We address this with a Sugiyama-style layered graph layout (Sugiyama et al., 1981), a standard technique that reorders nodes at each layer to minimize edge crossings. We initialize expert positions by global usage frequency, then perform forward and backward barycenter sweeps: each expert's vertical position is updated to the weighted mean of its neighbors'positions, where weights reflect token flow. A single pass suffices for a stable layout.

Critically, we compute one shared layout from the *aggregate* flow of all tokens, pooled across categories, and hold it fixed for all visualizations. In Figure 1, for example, the same expert ordering is used for all three panels. This ensures that visible differences in path coherence reflect genuine routing structure, not per-category layout optimization.

B Causally decomposing the residual stream

This appendix provides the full derivation of the decomposition introduced in Section 2.

The residual stream in a MoE serves two distinct purposes. It is simultaneously a **control signal** that directs routing (*what computation to perform*) and a **content payload** that is processed by the chosen experts (*what data is operated on*). Here, we introduce a parameter-free decomposition based on SVD that precisely separates these two channels, splitting the pre-MLP residual stream h_l into two orthogonal components:

$$h_l = \underbrace{h_l^{\text{vis}}}_{\text{router–visible channel}} + \underbrace{h_l^{\text{blind}}}_{\text{router–blind channel}}, \qquad \langle h_l^{\text{vis}}, h_l^{\text{blind}} \rangle = 0.$$

This allows us to **causally analyze what features drive expert specialization**. Due to the linearity of the routing gate, we can accomplish this without making modeling assumptions. We decompose h_l into a router-visible channel that causally drives routing and a router-blind channel invisible to the router, as follows.

Methodology. A MoE router makes its routing decision at layer l by computing the linear product of the routing matrix R_l with the residual stream h_l:

$$s_l = R_l^\top h_l.$$

Crucially, this linearity means the router can only "see" information in h_l that lies in the row space of its weight matrix $R_l \in \mathbb{R}^{N \times D}$.

We exploit this linearity via SVD to decompose the residual stream into two orthogonal components: what the router can see (router-visible) and what it cannot (router-blind). Figure 2 illustrates how components flow through the MoE layer at inference time, when considered through the lens of this decomposition.

1. **Computing the router's basis.** Let $R_l \in \mathbb{R}^{N \times D}$ be the routing matrix for layer l, where each row contains the weights for one of the N experts. We compute its singular value decomposition:

$$R_l = U_l \, \Sigma_l \, V_l^\top, \qquad \Sigma_l = \text{diag}(\sigma_1 \geq \cdots \geq \sigma_r > 0),$$

where $r = \text{rank}(R_l) \leq \min(N, D)$. We retain only the r right-singular vectors in $V_l \in \mathbb{R}^{D \times r}$ corresponding to non-zero singular values.

2. **Orthogonal projectors.** The projector onto the router's row space is:

$$P_l = V_l V_l^\top \in \mathbb{R}^{D\times D}, \qquad P_l^2 = P_l,\ P_l^\top = P_l.$$

We decompose each residual stream as:

$$h_l^{\text{vis}} = P_l\, h_l, \qquad h_l^{\text{blind}} = (I - P_l)\, h_l.$$

By construction, $R_l h_l^{\text{blind}} = 0$, meaning the router cannot access any information in h_l^{blind}. Thus, *all routing decisions depend solely on* h_l^{vis}, while h_l^{blind} is passed along with h_l^{vis} to the routed experts for computation.

This decomposition mathematically guarantees that only h_l^{vis} can causally influence expert selection; any relationship between h_l^{blind} and routing outcomes is purely correlative, not causal.

C Models and Data

To ensure our findings represent general principles of MoE design and not artifacts of a specific architecture, we analyze a diverse suite of high-performance models. Our selection spans different architectural families, attention mechanisms, parameter counts, and sparsity configurations. We test `Qwen3-30B-A3B` (Yang et al., 2025), `gpt-oss-20b` (Agarwal et al., 2025), `GLM-4.5-Air` (Zeng et al., 2025), `OlMoE` (Muennighoff et al., 2025), `Granite-4-Tiny` (Mishra et al., 2025), and `Deepseek-v2-Lite` (DeepSeek-AI et al., 2024).

Table 1: Summary of models analyzed. L = number of MoE layers (i.e., decoder layers excluding dense layers); D = hidden dimension, E = experts per MoE layer; k = active experts per layer; I = expert intermediate dimension (expressed as a fraction of D).

Model	Active/Total Params	L	D	E	k	I	Shared Experts	Notable characteristics
Qwen3-30B-A3B	3B / 30B	48	2048	128	8	$\frac{3}{8}D$	–	Highly sparse activation
GPT-OSS-20B	4B / 20B	24	2880	32	4	D	–	Low-precision experts
GLM-4.5-Air	12B / 106B	45	4096	128	8	$\frac{11}{32}D$	✓	Dense first layer
OLMoE	1B / 7B	16	2048	64	8	$\frac{1}{2}D$	–	–
Granite-4-Tiny	1B / 7B	40	1536	62	6	$\frac{1}{3}D$	–	Mamba2/Transformer hybrid
DeepSeek-v2-Lite	2B / 16B	26	2048	64	6	$\frac{11}{16}D$	✓	Multihead latent attention

Models. All models are loaded with default bfloat16 weights, excluding GPT-OSS-20B (loaded with its recommended MXFP4 expert precision) and GLM-4.5-Air (loaded in quantized FP8). For each, we use the first supported attention implementation of FlashAttention-2 if supported by the model, and standard SDPA otherwise (except GPT-OSS-20B, which we load with recommended FlashAttention3).

Dataset. All results from Section 3 use a 2 million token dataset sampled at sequence level from a mix of the C4 (Raffel et al., 2023) and HPLTv2 (Burchell et al., 2025) datasets. For each model, we run forward passes, collect representations h_l and expert selections at each layer l. For path analysis results, we use a 10m token dataset sampled from the same sources.

D Routing is driven by a low-dimensional control subspace

This appendix provides full methodology and results for the signal amplification finding summarized in Section 3.

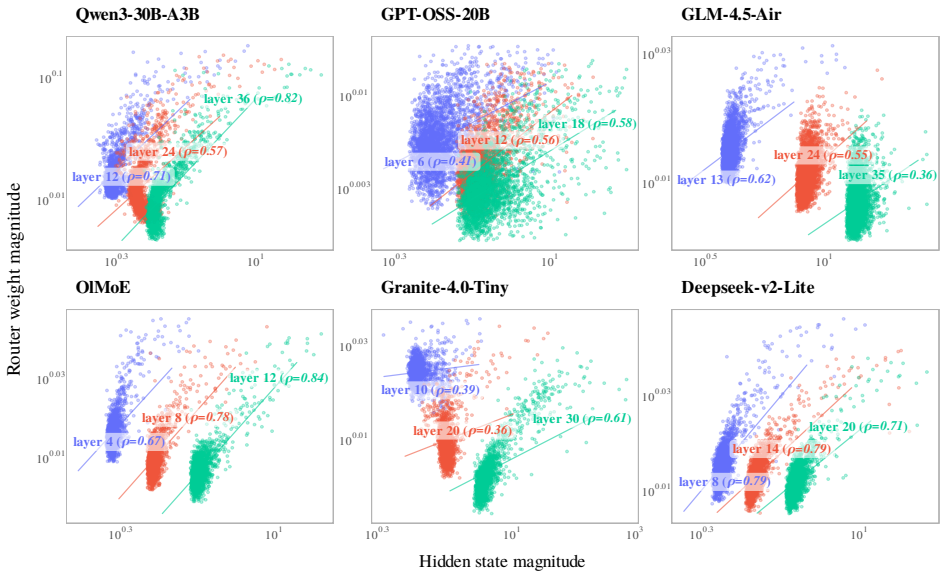

Figure 5: Per-dimension **hidden-state magnitude** $M^h_{l,d}$ (x-axis) versus **router-weight magnitude** $M^R_{l,d}$ (y-axis), both on log scales. One dot = one hidden dim d in layer l. Colors represent layers. Strong positive correlations are observed, indicating that router weights are largest on dimensions where the representation already has the highest magnitude.

We show that routing is a surprisingly low-dimensional process. MoEs learn a convergent strategy of **signal amplification**: routers learn to primarily listen to a tiny "control subspace" of dimensions that already have the highest magnitude, effectively amplifying the strongest features in the residual stream. We quantify this across layers and models.

Setup. To test this amplification hypothesis, for each layer l and dimension $d \in \{1, \ldots, D\}$ we measure:

1. **Residual stream magnitude**. $M^h_{l,d} = \mathbb{E}_{tokens}\left[|h_{l,d}|\right]$: the average magnitude of representations h_l at dimension d, averaged across ∼500k tokens sampled from standard text datasets (see Section C).

2. **Router weight magnitude**. $M^R_{l,d} = \frac{1}{N}\sum_{e=1}^{N}|R_{l,d,e}|$: the average magnitude of router weights R_l at dimension d, averaged across N experts.

We then compute the layer-wise Pearson correlation $\rho_l = corr(M^h_{l,\bullet}, M^R_{l,\bullet})$. A strong positive correlation would suggest routers implement an amplification dynamic, with routing driven by a small subspace of dimensions that already dominate the representation.

Findings. Across all models and layers, we observe strong positive amplification, with $\rho_l \approx 0.60$ on average. This pattern holds consistently across early and late layers, and from smaller to larger models, indicating a convergent strategy across diverse MoE architectures.

Figure 5 visualizes this relationship for quartile layers (e.g., layers 12, 24, 36 in a 48-layer model), while Table 2 provides more detailed values.

This reveals that expert selection does not primarily rely on scanning the full D-dimensional space for subtle patterns, but instead operates within a low-dimensional, high-magnitude control subspace. The router's decision is thus driven less by a distributed signal network and more by the amplified signal of a few dominant features.

Table 2: Layer-by-layer ρ_l values. Every fourth layer is listed for brevity; a – indicates a MoE layer at that position does not exist in that model.

	ρ_l					
Layer	QWEN3	GPT-OSS	GLM-4.5	OLMOE	GRANITE	DSV2-LITE
1	+0.46	+0.22	–	+0.70	+0.22	–
5	+0.74	+0.33	+0.11	+0.67	+0.33	+0.81
9	+0.64	+0.55	+0.53	+0.82	+0.22	+0.79
13	+0.68	+0.54	+0.61	–	+0.57	+0.81
17	+0.73	+0.56	+0.61	–	+0.53	+0.80
21	+0.56	+0.36	+0.63	–	+0.46	+0.72
25	+0.54	–	+0.52	–	+0.79	+0.63
29	+0.82	–	+0.42	–	+0.83	–
33	+0.76	–	+0.39	–	+0.76	–
37	+0.87	–	+0.33	–	+0.69	–
41	+0.82	–	+0.38	–	–	–
45	+0.75	–	+0.42	–	–	–

Causal validation. While these correlations are highly suggestive, they do not prove causation. Are these few dominant dimensions *causally sufficient* to drive the routing decision? We test this by training simple probes to predict the top-1 expert ID using only a tiny fraction of the representation.

For each layer l, we rank dimensions by their average magnitude $M^h_{l,d}$, then train two logistic regression probes:

1. **High-mag probe**: uses only the highest-magnitude 2% of dimensions in h_l.
2. **Baseline probe**: uses a random 2% of dimensions (results averaged across 10 random draws).

The results, shown in Table 3, are unambiguous: a probe seeing only the top 2% of dimensions can predict the router's choice with 30–80% accuracy, while the random baseline remains near chance (1–2%). For a model like OLMOE with $D = 2048$, this means a probe using just 40 dimensions can predict which of the 64 experts will be top-1 with 50% accuracy.

This demonstrates that a small, high-magnitude subspace is not just correlated with routing decisions, but causally sufficient to drive them.

Table 3: Top-1 expert ID accuracy of a probe trained on the **top 2% high-mag dimensions** (HIGH) vs. a probe trained on a **random 2% baseline** (BASE). Every fourth layer shown for brevity; a – indicates a MoE layer does not exist at that position in the model.

	QWEN3		GPT-OSS		GLM-4.5		OLMOE		GRANITE		DSV2-LITE	
Layer	HIGH	BASE	HIGH	BASE	HIGH	BASE	HIGH	BASE	HIGH	BASE	HIGH	BASE
1	**0.61**	0.01	**0.61**	0.06	–	–	**0.49**	0.02	**0.72**	0.02	–	–
5	**0.51**	0.02	**0.57**	0.06	**0.41**	0.01	**0.48**	0.03	**0.38**	0.04	**0.60**	0.02
9	**0.47**	0.02	**0.66**	0.04	**0.39**	0.01	**0.54**	0.03	**0.39**	0.03	**0.51**	0.02
13	**0.48**	0.02	**0.61**	0.04	**0.39**	0.01	**0.57**	0.03	**0.37**	0.02	**0.50**	0.02
17	**0.51**	0.02	**0.78**	0.06	**0.38**	0.01	–	–	**0.40**	0.03	**0.53**	0.02
21	**0.51**	0.01	**0.64**	0.04	**0.36**	0.01	–	–	**0.45**	0.04	**0.53**	0.02
25	**0.52**	0.02	–	–	**0.31**	0.01	–	–	**0.47**	0.03	**0.50**	0.02
29	**0.55**	0.02	–	–	**0.32**	0.01	–	–	**0.46**	0.05	–	–
33	**0.53**	0.02	–	–	**0.31**	0.01	–	–	**0.50**	0.06	–	–
37	**0.54**	0.02	–	–	**0.33**	0.01	–	–	**0.47**	0.03	–	–
41	**0.55**	0.01	–	–	**0.41**	0.02	–	–	–	–	–	–
45	**0.61**	0.03	–	–	**0.40**	0.01	–	–	–	–	–	–

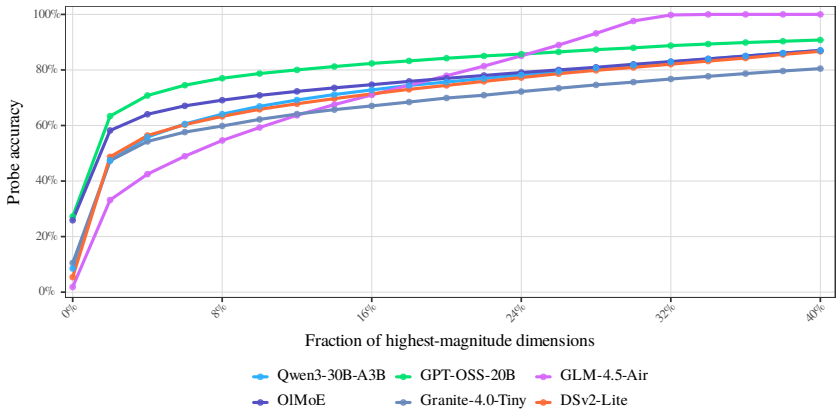

Figure 6: Probe accuracy versus fraction of highest-magnitude hidden dimensions used. Results shown for each model's midpoint layer. Accuracy saturates quickly, with the top 5% of dimensions capturing the vast majority of routing information, demonstrating the low-rank nature of routing.

How many dimensions are truly necessary? As shown in Figure 6, probe accuracy saturates quickly, with rapidly diminishing returns after using the top 2-5% of dimensions. This saturation profile is remarkably consistent across all tested models, despite significant variations in architecture.

E FEATURE ROTATION

Setup. We use the same token samples from Section D and decompose their hidden states into the router-visible h_l^{vis} and router-blind h_l^{blind} channels. If routing signals are indeed transient, we expect the router-visible channel to be less stable across layers than the router-blind channel.

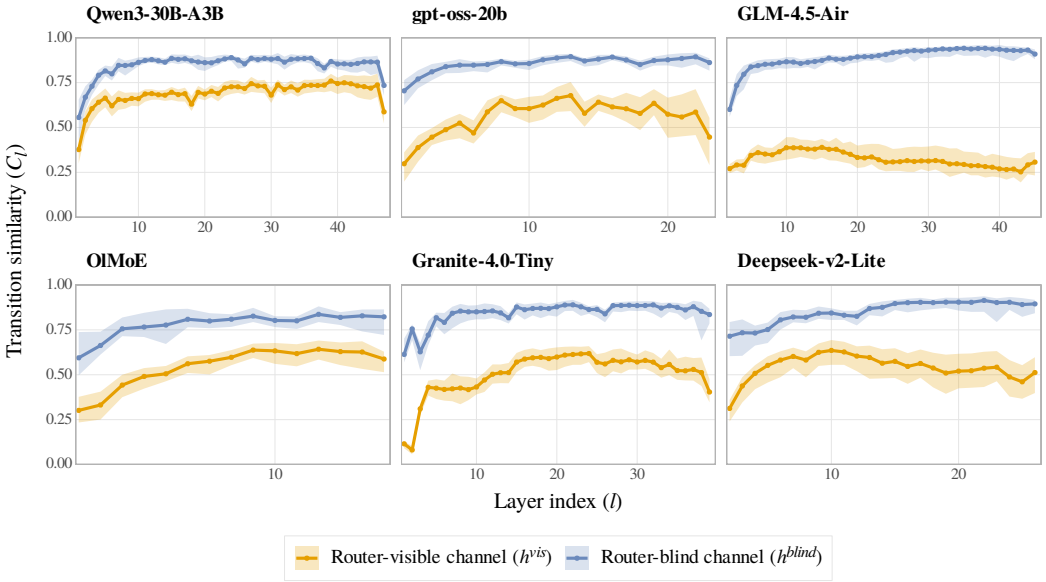

Figure 7: Average cross-layer cosine similarities C_l^{vis} and C_l^{blind} (shaded: 95% bootstrap CIs).

For every adjacent pair of layers $(l, l+1)$ we measure the average cosine similarity within each channel:

$$C_l^{\text{vis}} = \cos\big(h_l^{\text{vis}},\ h_{l+1}^{\text{vis}}\big), \qquad C_l^{\text{blind}} = \cos\big(h_l^{\text{blind}},\ h_{l+1}^{\text{blind}}\big).$$

High similarity (≈ 1) indicates a stable, accumulating representation, while low similarity suggests a transient, rewritten signal.

Findings. Figure 7 confirms a stark difference in stability. The router-blind channel (h^{blind}) is highly stable, with cross-layer similarity stabilizing at 0.75-0.90. This indicates that the information content not used for routing forms a continuous memory stream, accumulating and preserving features across layers.

In contrast, the router-visible channel (h^{vis}) is significantly less stable. This demonstrates that the specific features used to make a routing decision are transient and substantially change from one layer to the next.

F CHANNEL PROBES

We probe both channels for three surface features: language, token ID, and sequence position. We report normalized mutual information (MI%) to enable comparison across features with different cardinalities.

Across all six models and all three features (Figures 8 to 10), the same pattern holds: surface features are encoded primarily in h_l^{blind}, with h_l^{vis} carrying substantially less information. The disparity is largest at middle layers and narrows at the earliest and final layers, consistent with the interpretation discussed in Section 3.

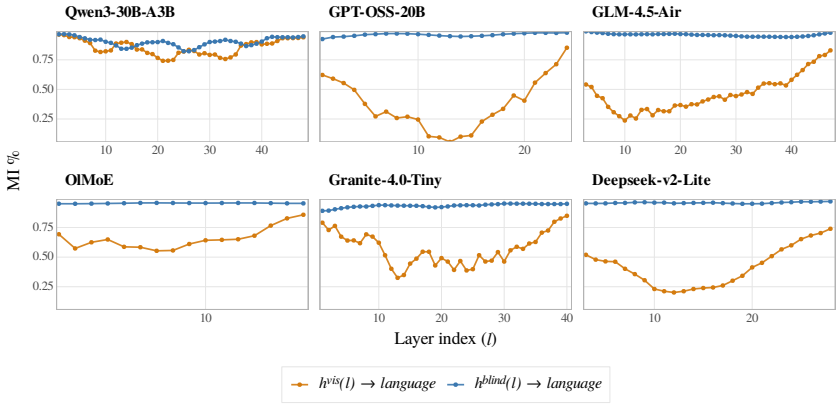

Figure 8: **Language probe.** Normalized MI% between channel representations and language label. h_l^{blind} (blue) encodes language consistently; h_l^{vis} (orange) carries substantially less, especially at middle layers.

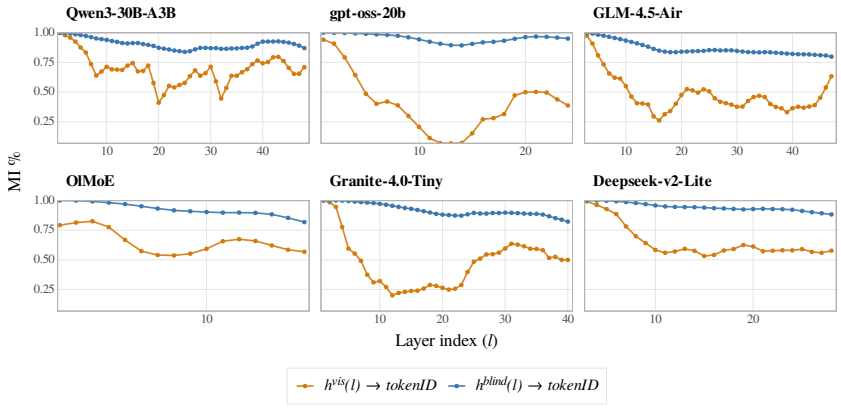

Figure 9: **Token ID probe results.** Normalized MI% between channel representations and token ID (filtered for top 100 token IDs).

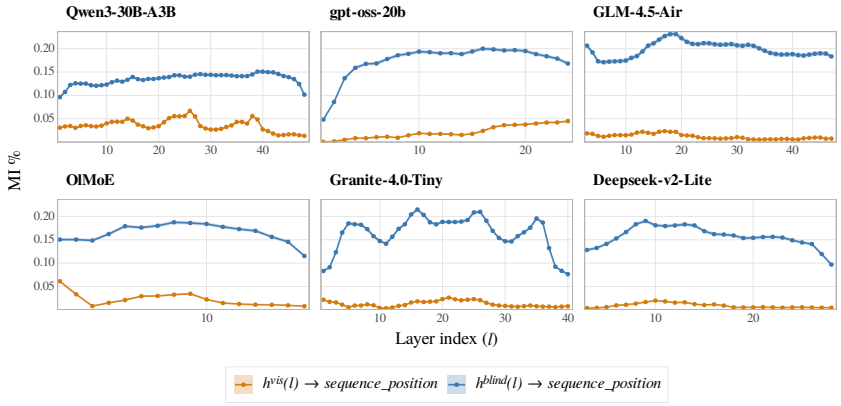

Figure 10: **Sequence position probe results.** Normalized MI% between channel representations and token position ID.

G Colon Token Dataset

To generate balanced data for Figure 1, we iteratively prompt several auxiliary LLMs to produce text containing the token **" : "** in each of three syntactic roles with high diversity, then randomly sample across auxiliary generators. Below are representative samples.

Type annotation.

```
def multiply_values(a: int, b: int) -> int: return a * b
```

```
function greetUser(name: string): string { return ‘Hello, ${name}’; }
```

Introductory colon.

```
Please provide the following information: user ID, date of last login, and account status.
```

The library's new guidelines are as follows: no food or beverages, no loud conversations, and no unapproved flyers.

Time separator.

Meeting scheduled at 09:30 tomorrow. Please arrive by 09:20 to set up.

[SERVER LOG] 2025-07-12 14:03:52: User session started. Session validated.

H LLM USAGE

We used an LLM for: (i) language editing and clarity and (ii) minimal code assistance. All technical content, experiments, and claims were implemented, checked, and verified by the authors. Any LLM-produced text/code was reviewed, edited, and validated; references and factual statements were manually verified. No LLM is an author; the authors take full responsibility for the content.