MoE Lens - An Expert Is All You Need

Marmik Chaudhari[1*], **Idhant Gulati**[1*], **Nishkal Hundia**[2*], **Pranav Karra**[1*], **Shivam Raval**[3]

[1]The Pennsylvania State University, University Park, PA 16802 USA
[2]University of Maryland, College Park, MD 20742 USA
[3]Harvard University, Cambridge, MA 02138 USA

{marmik, idhant.gulati, pranavkarra}@psu.edu, nhundia@umd.edu, sraval@harvard.edu

Abstract

Mixture of Experts (MoE) models enable parameter-efficient scaling through sparse expert activations, yet optimizing their inference and memory costs remains challenging due to limited understanding of their specialization behavior. We present a systematic analysis of expert specialization in MoEs through two complementary approaches: domain-specific routing patterns and an early decoding framework that tracks expert contributions to output representations. Our analysis of the DeepSeekMoE model reveals that despite having 64 routed experts with 6 active for each layer's computation, the model predominantly relies on a few specialized experts, with the top-weighted expert's output closely approximating the full ensemble prediction. We quantitatively validate these findings through a systematic analysis of the token routing distribution, demonstrating that very few experts handle over 50% of routing decisions across different specialized domains. Hidden state similarity between single and ensemble experts for every layer is extremely high, with some layers having cosine similarity as high as 0.95 and perplexity increasing by only 5% when using a single expert across all three domains. Our results indicate that Mixture of Experts models exhibit concentrated expertise highlighting potential opportunities for inference optimization through targeted expert pruning while maintaining model performance and opening avenues towards studying localization of learned knowledge in these models.

1 Introduction

Mixture of Experts (MoE) (Shazeer et al., 2017) models offer an efficient way to scale large language models by activating only a subset of model parameters for each input. However, MoE architectures face challenges spanning from training complexity and load balancing to routing inefficiencies and memory constraints (Liu et al., 2025), with many issues arising from how inputs are routed to experts and how specialization emerges. Recent architectures like DeepSeekMoE (Dai et al., 2024) have improved expert specialization and load balancing (Wang et al., 2024), demonstrating progress in mitigating some of these challenges but fundamental questions about the expert behavior like specialization and knowledge redundancy in a MoE still remain unanswered.

Expert specialization is an emergent property first observed in vision models like AlexNet (Krizhevsky et al., 2012) and InceptionV1 (Szegedy et al., 2014), where different network branches develop specialized feature representations. Language Models like Monet (Park et al., 2024) have demonstrated that drastically scaling up the number of experts, individual experts develop specialized classes like chemical compounds, electromagnetism and diseases. MoE architectures with relatively fewer experts exhibit knowledge redundancy (Oldfield et al., 2024) where a few experts cover the same diverse, unrelated concepts. This raises the question of how to identify and analyze the experts in a MoE that develop into monosemantic units, each specializing in distinct linguistic

*Equal contributions.
Our code is available at: https://github.com/MarmikChaudhari/moe-interp/tree/moe-lens

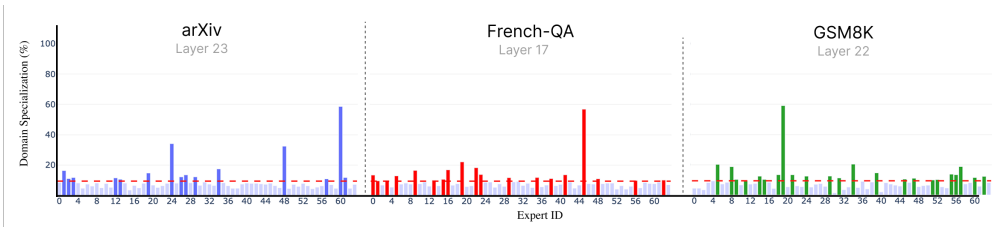

Figure 1: **Expert Specialization** in DeepSeekMoE. We visualize the distribution of tokens that are routed to an expert for our English, French-QA, and GSM8K datasets. The y-axis shows the routing percentage per expert, with the red dashed line indicating a uniform routing baseline ($\approx 9.4\%$). See Appendix A.1 for extended plots for other models and layers.

or computational domains, and how to leverage specialized experts to reduce the inference latency while preserving the knowledge and reasoning capabilities learned by the model during pre-training.

In this paper, we: (1) investigate expert specialization behavior across distinct domains by analyzing routing distributions and identifying domain-specialized experts across three data domains, (2) use an early-decoding strategy to interpret how individual experts contribute to residual stream representations at each layer, and find that a single expert is sufficient to converge to output representations in next-token prediction tasks, and (3) validate our findings through quantitative measures like cosine similarity and perplexity. Our empirical results reveal that while certain MoEs exhibit some domain specialization, they primarily rely on a small set of experts with other experts providing minimal contributions to the final predictions building towards an interpretable pruning approach that preserves the next token prediction accuracy while making the model more sparse.

2 Background

MoE Layer. In a Mixture-of-Experts (MoEs) architecture (Muennighoff et al., 2024; Dai et al., 2024; Xue et al., 2024), the Feed-Forward Network (FFN) in a Transformer is substituted with MoE layers at specified intervals. The MoE layer consists of set of n experts $E_1, ..., E_n$, each structurally identical to a standard FFN, and a learned routing network, r, which assigns routing probabilities to all n experts for each input token, x. The output of the ℓ-th MoE layer for the t-th token, $\mathbf{h}_t^\ell$, is a weighted sum of the expert outputs scaled by their corresponding routing probability across all chosen Top-k experts, E_i where $i \in$ top-k. Let $\mathbf{u}_t^\ell$ denote the hidden state of the t-th token after the l-th attention module (post-attention residual stream). Mathematically,

$$\mathbf{h}_t^\ell = \sum_{i=1}^{\text{Top-}k(r_t(x))} (r_{i,t} E_i(x)) + \mathbf{u}_t^\ell, \tag{1}$$

To analyze how experts specialize in processing different types of inputs, we define **expert specialization** as described in Muennighoff et al. (2024) to be the fraction of tokens from a particular domain D for which expert E_i is selected as one of the top-k experts.

$$\text{Expert specialization}(E_i, D) = \frac{N_{E_i,D}^{(k)}}{N_D}, \tag{2}$$

where $N_{E_i,D}^{(k)}$ is the number of tokens from domain D for which E_i is among the top-k selected experts, and N_D is the total number of tokens from domain D. We consider an expert to be specialized in D if it processes significantly more tokens than the uniform routing baseline of 6/64 $\approx 9.4\%$.

Shared Experts. Every input token is routed to the Shared Expert in DeepSeekMoE and the output of the Shared Expert, E_S is added to the output hidden state, $\mathbf{h}_t^\ell$. We primarily focus on the routed experts since the shared experts are dedicated to capturing common knowledge across varying contexts (Dai et al., 2024)

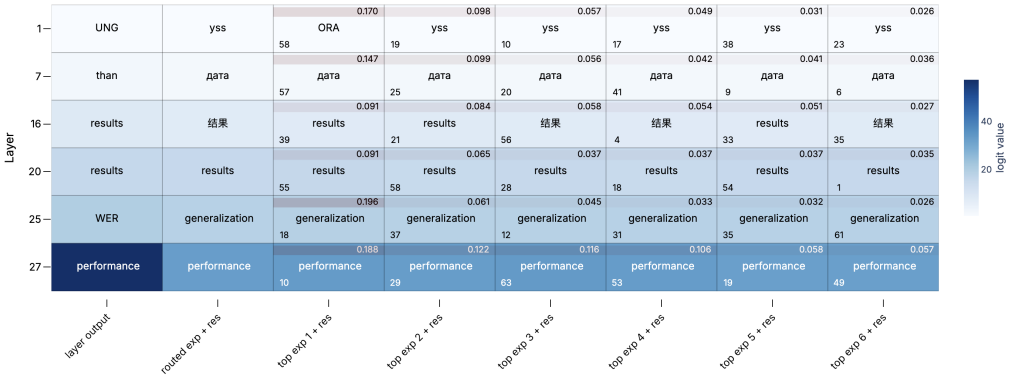

Figure 2: An example of early decoding using **LogitLens** for DeepSeekMoE on an example input: "When datasets are sufficiently large, increasing the capacity (number of parameters) of neural networks can give much better prediction". Each cell shows the top-1 token prediction after the final token "these" across layers (rows) for layer output, routed experts + residual stream for various top-k values. Color intensity indicates prediction confidence. The expert index is denoted by the lower-left subscript number and the top-right superscript indicates expert weight. See Appendix A.2 for other domains.

Early decoding using Logit Lens. Early decoding (Schuster et al., 2022) is the analysis of a model's intermediate predictions before reaching the final layer. The LogitLens (nostalgebraist, 2021) is an early decoding technique that directly decodes the hidden states at any intermediate layer ℓ for t-th token, $\mathbf{h}_t^\ell$, using the model's pretrained unembedding matrix, W_U. The resulting distribution of logits roughly converges toward the model's final prediction across layers, offering a window into how the model progressively refines its predictions.

$$\text{LogitLens}(\mathbf{h}_t^\ell) = \text{LayerNorm}(\mathbf{h}_t^\ell)W_U \tag{3}$$

To understand how expert, E_i, at a particular layer ℓ contributes to the final output representation, we further extend the LogitLens by adding post-attention residual stream for an expert $\mathbf{u}_t^\ell$ and then projecting it to the vocabulary space. The residual stream is analogous to a communication channel (Elhage et al., 2021) through which the experts incrementally refine the hidden state, $\mathbf{h}_t^\ell$. Each expert's output can be interpreted as a targeted modification to specific subspaces of the representation. By using extended LogitLens (Belrose et al., 2023) , we observe how individual experts update the prediction distribution by writing their specialized knowledge into the residual stream.

$$\text{LogitLens}^{ext}(\mathbf{h}_t^\ell) = \text{LayerNorm}(\mathbf{h}_t^\ell + \mathbf{u}_t^\ell)W_U \tag{4}$$

Notation. For layer ℓ, we represent the hidden state of the top-weighted expert (top-$k = 1$) combined with the residual output as $\mathbf{H_t^{\ell_1}}$ and the hidden state of the top-$k = 6$ weighted expert combined with the residual output as $\mathbf{H_t^{\ell_6}}$.

3 Experiments

Model. For all experiments, we use DeepSeekMoE (Dai et al., 2024) with 2 shared + 64 routed experts with top-$k = 6$. The model is pretrained on cross entropy loss combined with expert-level and device-level balance loss to prevent a router collapse. We also extended our experiments by utilizing OLMoE by Allen AI (Muennighoff et al., 2024). See Appendix A.

Datasets. We use curated subsets of seven datasets for our experiments: (1) GitHub code dataset subset of Paloma (Magnusson et al., 2024), (2) subset of Gutenberg English Dataset (Lahiri, 2014), a dataset of English books, (3) French-QA subset of French Question Answering Dataset (FQuAD) (Martin et al., 2020), a dataset on native French reading comprehension, (4) AIME Problem Sets from 1983 to 2024 (Veeraboina, 2023), (5) Chinese Fineweb Edu (Yu et al., 2025), a high-quality

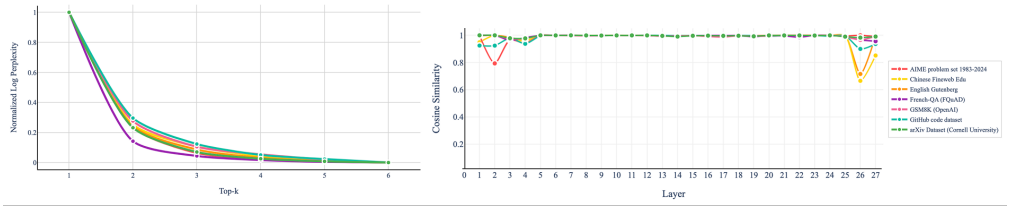

Figure 3: (Left) Normalized log perplexity across different values of top-k experts for various domains for next-token prediction task. (Right) Cosine similarity between the hidden states of $\mathbf{H}_\mathbf{t}^{\ell_1}$ and $\mathbf{H}_\mathbf{t}^{\ell_6}$ across all 27 layers shows consistently high alignment.

Chinese pre-training corpus, (6) arXiv Dataset (Clement et al., 2019), a collection of pre-print scholarly articles gathered from various scientific fields, and (7) GSM8K (Cobbe et al., 2021), a grade school math word problems dataset created by OpenAI.

Experiments. Our empirical investigation consists of three experiments: (i) For each input prompt in domain $D \in \{\text{English, Code, French}\}$, we compute Expert specialization (E_i, D) as defined in equation 2. (ii) We apply extended LogitLens (equation 4) to analyze components within MoE layers. We project three distinct hidden states to logits: (1) individual expert outputs, E_i, (2) the weighted sum of the top-$k = 6$ expert outputs, and (3) final layer outputs. For each layer ℓ, we compare the layer output, $\mathbf{h}_t^\ell$, and the top-weighted expert output combined with the residual output, $\mathbf{H}_\mathbf{t}^{\ell_1}$, for next-token prediction tasks from each domain, D. (iii) We compute the cosine similarity between the hidden state of the top weighted expert's output (expert + residual), $\mathbf{H}_\mathbf{t}^{\ell_1}$, and the hidden state of the top-$k = 6$ weighted expert outputs combined with residual output, $\mathbf{H}_\mathbf{t}^{\ell_6}$, across each domain dataset to assess the alignment between individual and combined expert contributions in the hidden space. We also analyze the perplexity of the next token prediction for each domain D and how it changes with different number of active experts to observe whether using the single most-activated expert maintains comparable loss and prediction performance to using all 6 routed experts.

4 Results

The expert specialization results in DeepSeekMoE, shown in figure 1, reveal two key patterns: First, only a small number of experts show strong specialization (significantly higher than uniform routing frequency) for any domain. Second, most experts demonstrate minimal domain-specific activity.

The extended LogitLens gives us evidence that solely projecting $\mathbf{H}_\mathbf{t}^{\ell_1}$ across layers decode to roughly the same next token prediction as the output at the end of that layer, $\mathbf{h}_t^\ell$. Furthermore, $\mathbf{H}_\mathbf{t}^{\ell_1}$ has nearly identical next-token distribution as $\mathbf{H}_\mathbf{t}^{\ell_6}$.

We also observe very high cosine similarity between $\mathbf{H}_\mathbf{t}^{\ell_1}$ and $\mathbf{H}_\mathbf{t}^{\ell_6}$ across all layers and each domain, D as shown in figure 3 (right) indicating that the top-weighted expert is contributing the most in shaping final output representation whereas the contributions of other experts are minimal in the hidden space. Concretely, $\mathbf{H}_\mathbf{t}^{\ell_1} \approx \mathbf{H}_\mathbf{t}^{\ell_6}$. The perplexity moderately increases when reducing top-$k = 6$ to 1 as demonstrated in figure 3 (right) which validates the claim that the top-weighted expert, when combined with the residual stream, produces representations closely aligned with the layer output.

5 Conclusions and Future Work

We empirically show that a single top-weighted expert, combined with the residual stream, closely approximates the full ensemble output for a layer across multiple data domains in DeepSeekMoE. Such a high degree of specialization suggests a potential for further sparsification during inference. By activating only the highest-weighted expert instead of all top-k experts and selectively prun-

ing the non-essential experts, computational costs and memory requirements can be significantly reduced while maintaining comparable model performance across a variety of tasks.

We demonstrate these findings using DeepSeekMoE as a representative MoE architecture. Examining additional MoE variants such as OLMoE (Muennighoff et al., 2024), DeepSeek-V2 (DeepSeek-AI et al., 2024), and DeepSeek-VL2 (Wu et al., 2024) can lead to a broader understanding of specialization behavior in MoE models. While our analysis uses the LogitLens approach, extending this work by using TunedLens may provide more robust token decoding through layer-wise learned transformation between intermediate and prefinal layer representations. Furthermore, these findings open several research directions: developing dynamic expert selection strategies that adapt to input complexity, analyzing the internal representation sparsity of individual experts to localize factual knowledge, and examining specialization patterns across different MoE architectures to inform future training objectives.

REFERENCES

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, 2023. URL https://arxiv.org/abs/2303.08112.

Colin B. Clement, Matthew Bierbaum, Kevin P. O'Keeffe, and Alexander A. Alemi. On the use of arxiv as a dataset, 2019.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2401.06066.

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.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt,

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

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Shibamouli Lahiri. Complexity of Word Collocation Networks: A Preliminary Structural Analysis. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 96–105, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/E14-3011.

Jiacheng Liu, Peng Tang, Wenfeng Wang, Yuhang Ren, Xiaofeng Hou, Pheng-Ann Heng, Minyi Guo, and Chao Li. A survey on inference optimization techniques for mixture of experts models, 2025. URL https://arxiv.org/abs/2412.14219.

Ian Magnusson, Akshita Bhagia, Valentin Hofmann, Luca Soldaini, Ananya Harsh Jha, Oyvind Tafjord, Dustin Schwenk, Evan Pete Walsh, Yanai Elazar, Kyle Lo, Dirk Groeneveld, Iz Beltagy, Hannaneh Hajishirzi, Noah A. Smith, Kyle Richardson, and Jesse Dodge. Paloma: A benchmark for evaluating language model fit, 2024. URL https://arxiv.org/abs/2312.10523.

d'Hoffschmidt Martin, Vidal Maxime, Belblidia Wacim, and Brendlé Tom. FQuAD: French Question Answering Dataset. *arXiv e-prints*, art. arXiv:2002.06071, Feb 2020.

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, 2024. URL https://arxiv.org/abs/2409.02060.

nostalgebraist. Interpreting GPT: The logit lens, 2021. URL https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. Accessed: 2025-02-07.

James Oldfield, Markos Georgopoulos, Grigorios G. Chrysos, Christos Tzelepis, Yannis Panagakis, Mihalis A. Nicolaou, Jiankang Deng, and Ioannis Patras. Multilinear mixture of experts: Scalable expert specialization through factorization, 2024. URL https://arxiv.org/abs/2402.12550.

Jungwoo Park, Young Jin Ahn, Kee-Eung Kim, and Jaewoo Kang. Monet: Mixture of monosemantic experts for transformers, 2024. URL https://arxiv.org/abs/2412.04139.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling, 2022. URL https://arxiv.org/abs/2207.07061.

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, 2017. URL https://arxiv.org/abs/1701.06538.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014. URL https://arxiv.org/abs/1409.4842.

Hemish Veeraboina. Aime problem set 1983-2024, 2023. URL https://www.kaggle.com/datasets/hemishveeraboina/aime-problem-set-1983-2024.

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. Auxiliary-loss-free load balancing strategy for mixture-of-experts, 2024. URL https://arxiv.org/abs/2408.15664.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, Zhenda Xie, Yu Wu, Kai Hu, Jiawei Wang, Yaofeng Sun, Yukun Li, Yishi Piao, Kang Guan, Aixin Liu, Xin Xie, Yuxiang You, Kai Dong, Xingkai Yu, Haowei Zhang, Liang Zhao, Yisong Wang, and Chong Ruan. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding, 2024. URL https://arxiv.org/abs/2412.10302.

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, 2024. URL https://arxiv.org/abs/2402.01739.

Yijiong Yu, Ziyun Dai, Zekun Wang, Wei Wang, Ran Chen, and Ji Pei. Opencsg chinese corpus: A series of high-quality chinese datasets for llm training, 2025. URL https://arxiv.org/abs/2501.08197.

A Appendix

A.1 Expert Specialization

A.1.1 DeepSeekMoE

Figure 4: **Expert Specialization (Part 1)** of DeepSeekMoE for various layers. We visualize how frequently tokens from different domains are routed to the 64 experts using top-$k = 6$ routing. The y-axis shows routing percentage per expert, with the red dashed line indicating uniform routing baseline ($\approx 9.4\%$).

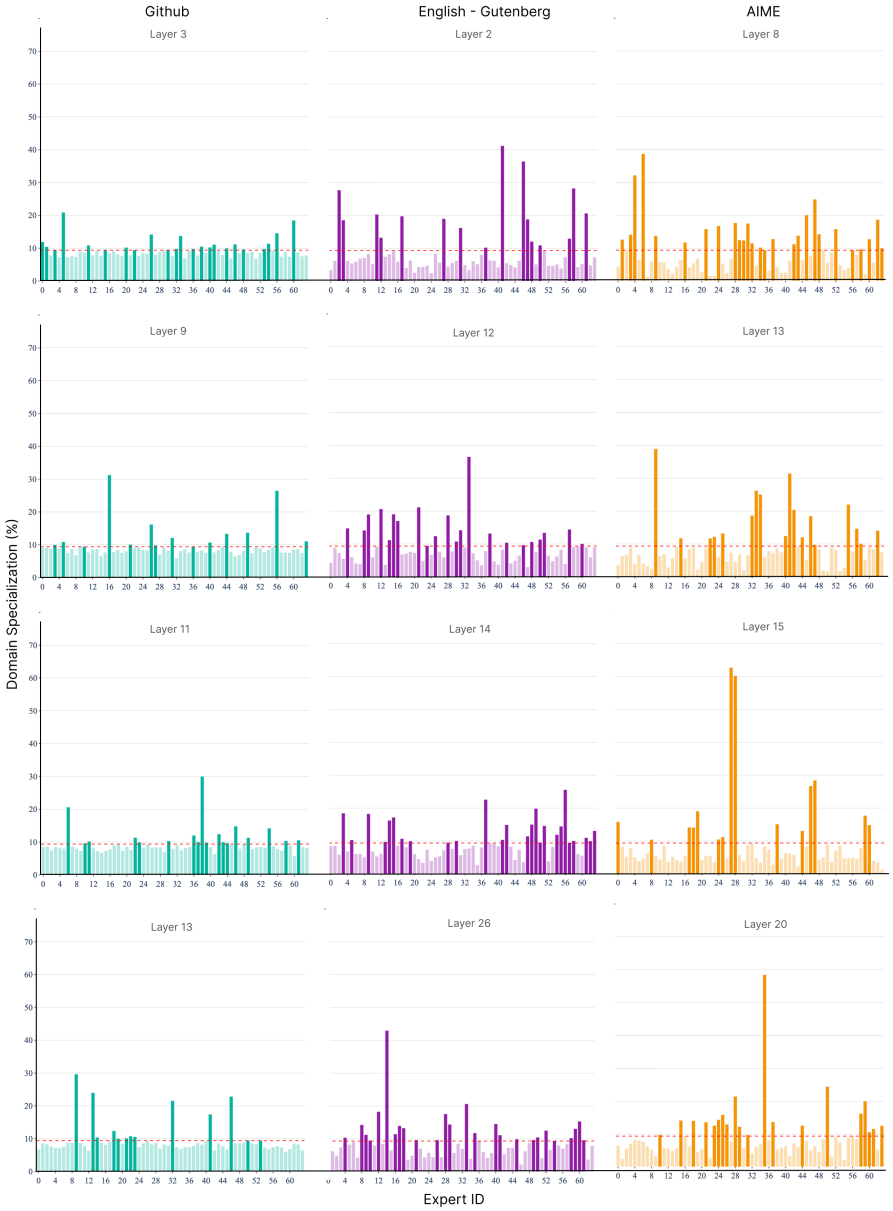

Figure 5: **Expert Specialization (Part 2)** of DeepSeekMoE for various layers. We visualize how frequently tokens from different domains are routed to the 64 experts using top-$k = 6$ routing. The y-axis shows routing percentage per expert, with the red dashed line indicating uniform routing baseline ($\approx 9.4\%$).

A.1.2 QWEN 1.5 MOE

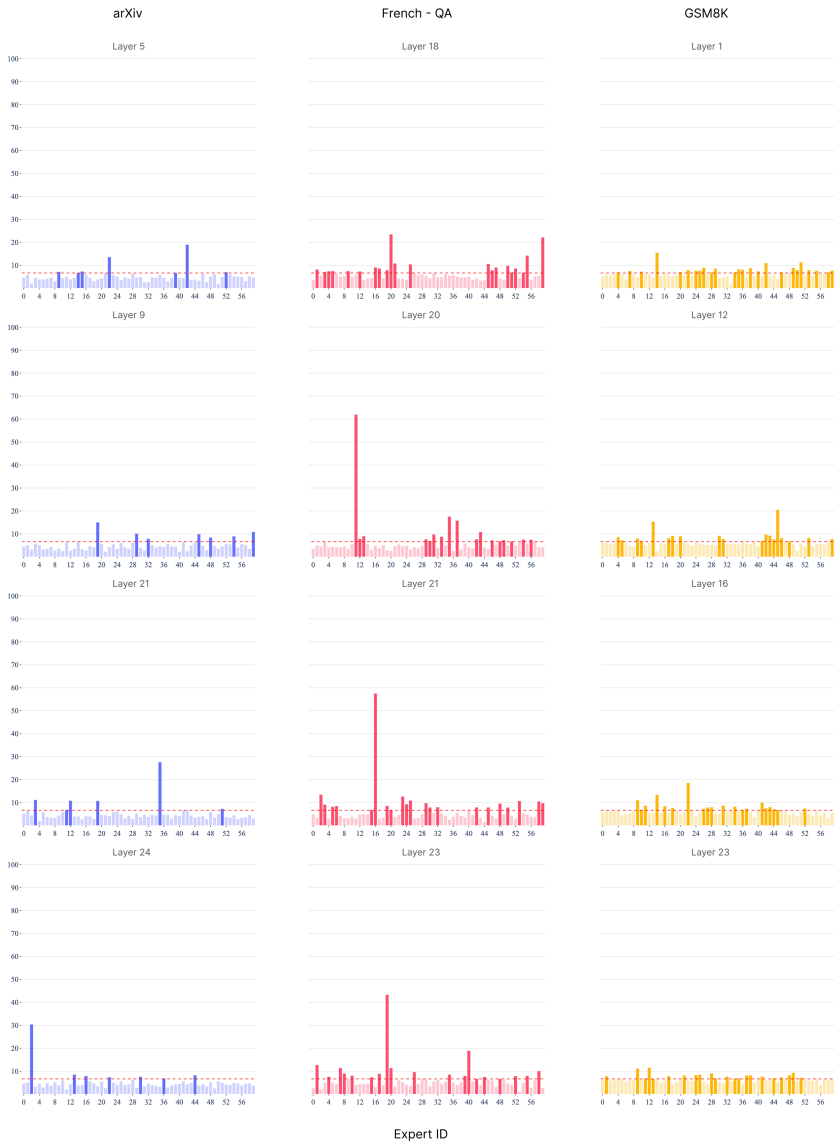

Figure 6: **Expert Specialization (Part 1)** of Qwen 1.5 MoE for various layers. We visualize how frequently tokens from different domains are routed to the 60 experts using top-$k = 4$ routing. The y-axis shows routing percentage per expert, with the red dashed line indicating uniform routing baseline ($\approx 6.67\%$).

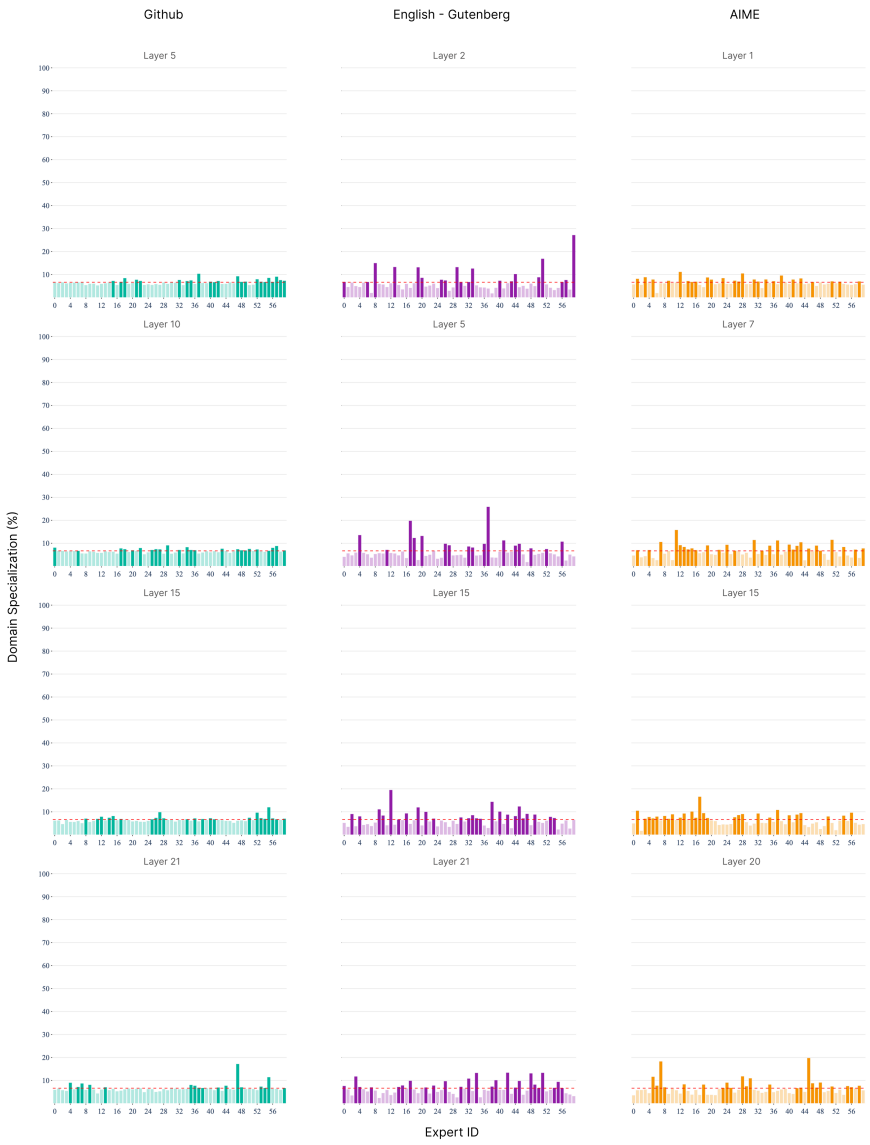

Figure 7: **Expert Specialization (Part 2)** of Qwen 1.5 MoE for various layers. We visualize how frequently tokens from different domains are routed to the 60 experts using top-$k = 4$ routing. The y-axis shows routing percentage per expert, with the red dashed line indicating uniform routing baseline ($\approx 6.67\%$).

A.1.3 OLMoE

Figure 8: **Expert Specialization (Part 2)** of OLMoE for various layers. We visualize how frequently tokens from different domains are routed to the 64 experts using top-$k = 8$ routing. The y-axis shows routing percentage per expert, with the red dashed line indicating uniform routing baseline ($\approx 12.5\%$).

Figure 9: **Expert Specialization (Part 2)** of OLMoE for various layers. We visualize how frequently tokens from different domains are routed to the 66 experts using top-$k = 8$ routing. The y-axis shows routing percentage per expert, with the red dashed line indicating uniform routing baseline ($\approx 12.5\%$).

A.2 Logit Lens

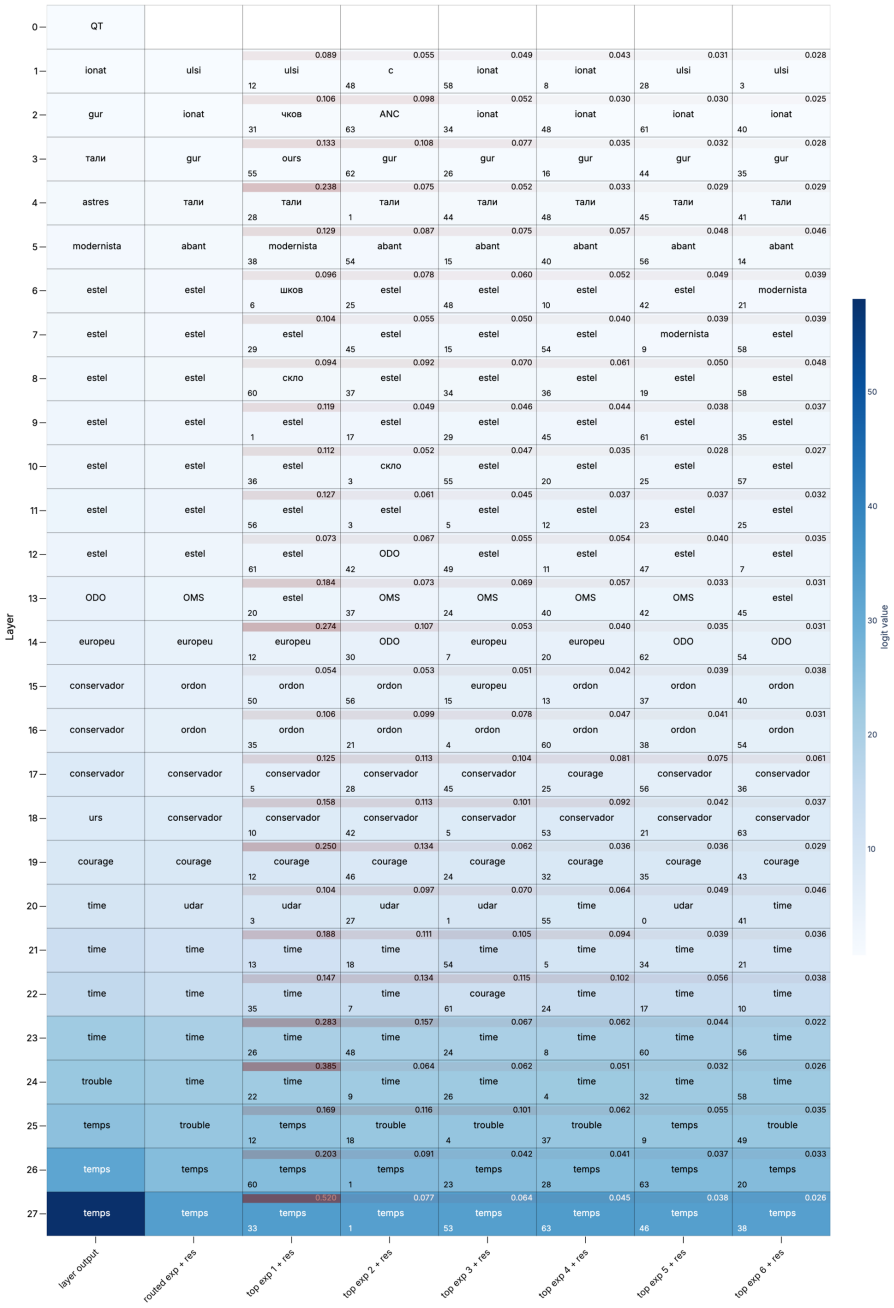

Figure 10: **LogitLens** visualization for DeepSeekMoE on the french input sequence: "Dans le silence feutre de la nuit, les etoiles semblent murmurer d'anciens secrets a ceux qui prennent le". Each cell shows the top-1 token prediction after "le" across layers (rows) for layer output, routed experts with residual stream for various top-k values. Color intensity indicates prediction confidence. The lower-left subscript indicates expert indices and the top-right superscript indicates expert weight.

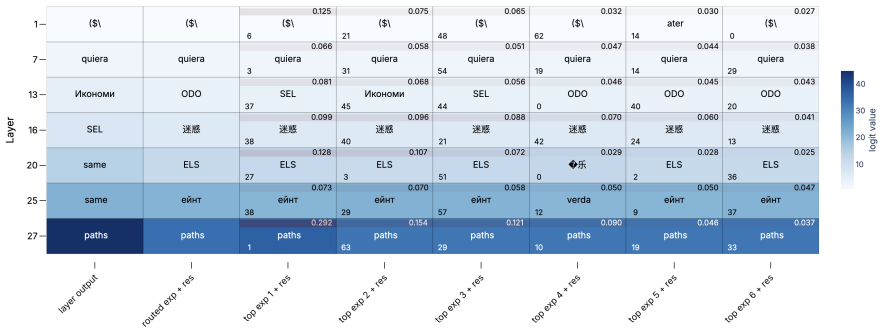

Figure 11: **LogitLens** visualization for DeepSeekMoE on the english input sequence: "One might expect language modeling performance to depend on model architecture, the size of neural models, the computing power used to train them, and the data available for this". Each cell shows the top-1 token prediction after "this" across layers (rows) for layer output, routed experts with residual stream for various top-k values. Color intensity indicates prediction confidence. The lower-left subscript indicates expert indices and the top-right superscript indicates expert weight.