MoE Lens - An Expert Is All You Need

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

[1]Penn State University [2]University of Maryland [3]Harvard University

TL;DR: We discover that specialization in Mixture of Experts models is localized among a few experts across various domains, with the top-weighted expert closely approximating full ensemble's next token predictions.

Introduction

- The original aim of Mixture of Experts (MoEs) was to allow models to scale capacity without proportional increases in computational costs by sparsely activating a subset of parameters (experts) for each input token.
- However, scaling MoEs and expert count effectively has faced limitations due to its discrete expert routing such as training instability and forced generalization across multiple distinct semantic concepts for each expert, hindering specialization.
- Current MoEs face a fundamental tension between achieving fine-grained specialization through larger expert counts and the observed parameter inefficiency where only a small subset of experts handle most computation across domains.
- Significant progress in interpretability methods, such as Logit Lens, has shown success in uncovering the activations of language models - what insights can we gain when we apply these techniques to analyze expert specialization in MoEs?

Methods

1. Tracing token routing distribution

Number of tokens from domain D for which expert E_i is selected as one of the Top-k experts

$$\text{Expert specialization}(E_i, D) = \frac{\overset{\uparrow}{N^{(k)}_{E_i,D}}}{\underset{\downarrow}{N_D}}$$

Total number of tokens from domain D

- We compute expert specialization by measuring the fraction of tokens from each domain routed to each expert.
- Domains were represented by distinct datasets: GSM8K & AIME, Github subset of Paloma, FrenchQA (FQuAD), subset of Chinese Fineweb Edu, Guttenberg English for Math, Code, French, Chinese and English domain respectively.
- We classify an expert as domain specialized if its routing frequency is significantly higher than a uniform routing baseline.

2. Logit Lens for analyzing expert contributions

Post-attention residual stream at layer ℓ for token t

$$\underset{\downarrow}{\text{LogitLens}^{ext}(h_t^{\ell})} = \text{LayerNorm}(h_t^{\ell} + \overset{\uparrow}{u_t^{\ell}})\underset{\downarrow}{W_U}$$

Hidden states at any intermediate layer ℓ for t-th token

Model's pretrained unembedding matrix

- For each layer ℓ and token t, we apply Logit Lens to project two different hidden states to the vocab space: individual expert output E_i, weighted sum of Top-k expert output (H_t^{ℓ}).
- We also incorporate post-attention residual stream for projecting individual expert outputs.
- We perform these experiments on **DeepSeek-MoE, OLMoE, and Qwen 1.5 MoE** to validate our findings across different MoE implementations.

Results

1. Few experts dominate token routing across domains

The routing distribution reveals two key patterns in MoEs: **(1)** only a small number of experts show strong specialization for any domain. **(2)** most experts demonstrate minimal domain-specific activity.

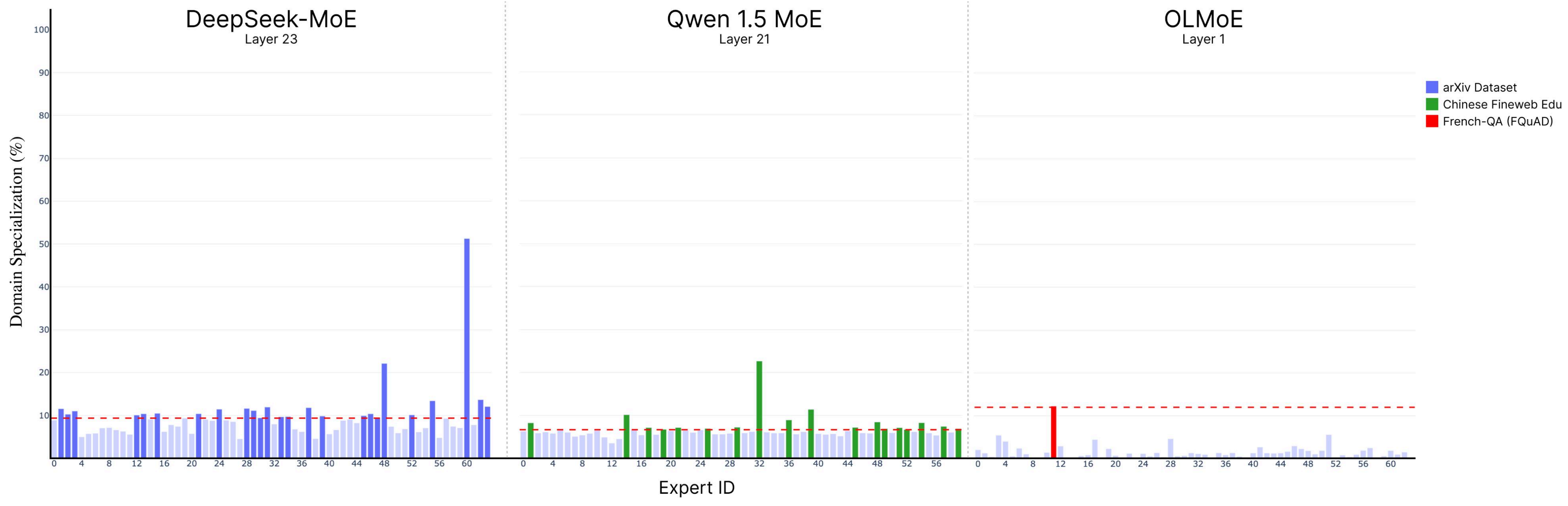
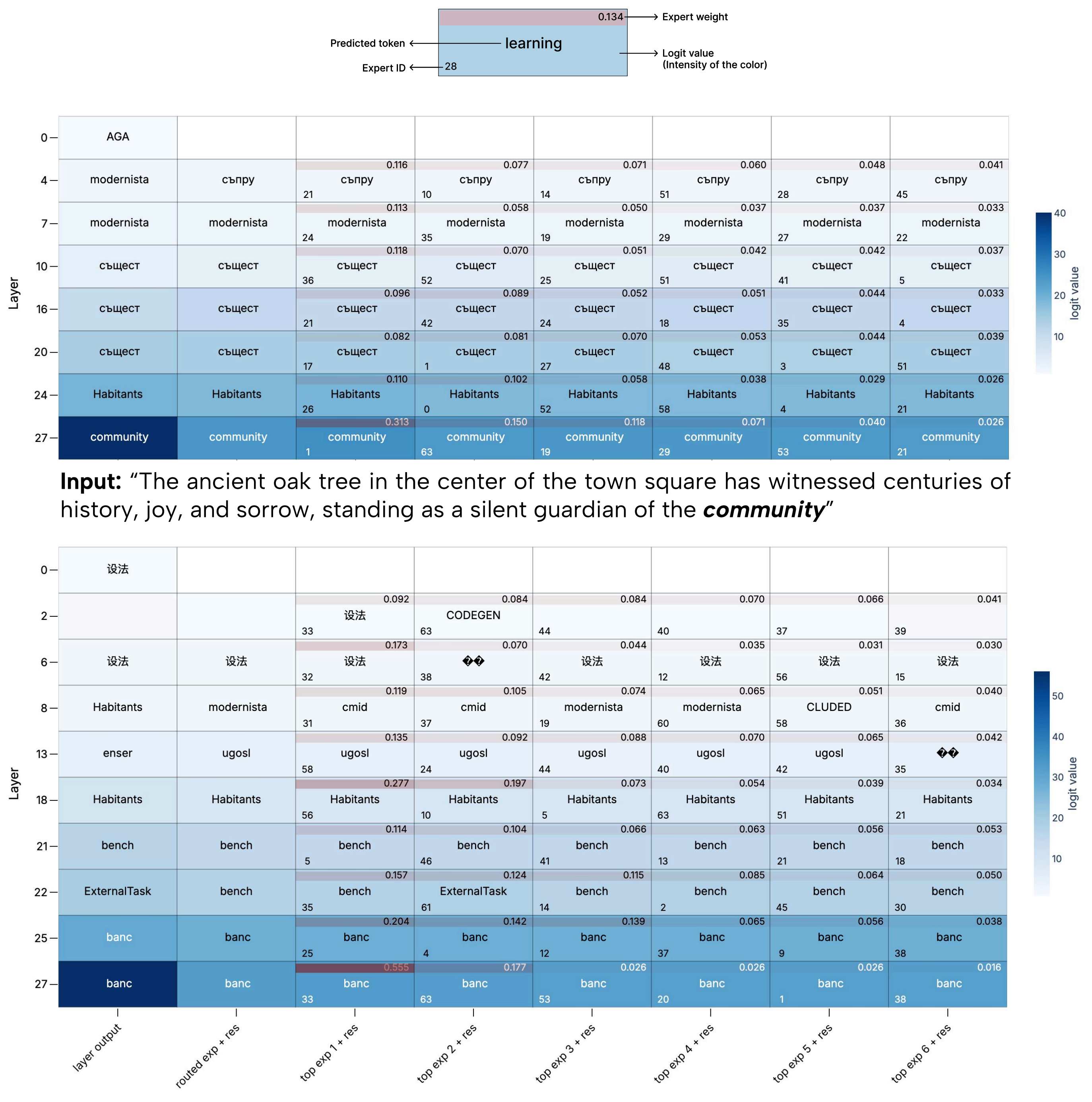


Figure 1: Expert Specialization across Multiple MoE Architectures. We visualize domain-specific routing patterns for DeepSeek-MoE, Qwen 1.5 MoE, and OLMoE models across three different domains. The y-axis shows domain specialization percentage for each expert, with the red dashed line indicating the uniform routing baseline (~9.375% for DeepSeek-MoE, ~6.67% for Qwen 1.5 MoE, and ~12.5% for OLMoE)

2. Top-weighted expert approximates full ensemble's output

The extended Logit Lens gives us evidence that solely projecting the hidden state of the top weighted expert's output, $H_t^{\ell_1}$ across layers decode to roughly the same next token prediction as the output at the end of that layer, i.e, the hidden state of the Top-k weighted expert outputs, $H_t^{\ell_6}$.



Layer	layer output	routed exp + res	top exp 1 + res	top exp 2 + res	top exp 3 + res	top exp 4 + res	top exp 5 + res	top exp 6 + res
0	AGA							
4	modernista	cъпру	0.116 cъпру 21	0.077 cъпру 10	0.071 cъпру 14	0.060 cъпру 51	0.048 cъпру 28	0.041 cъпру 45
7	modernista	modernista	0.113 modernista 24	0.058 modernista 35	0.050 modernista 19	0.037 modernista 29	0.037 modernista 27	0.033 modernista 22
10	същест	същест	0.118 същест 36	0.070 същест 52	0.051 същест 25	0.042 същест 51	0.042 същест 41	0.037 същест 5
16	същест	същест	0.096 същест 21	0.089 същест 42	0.052 същест 24	0.051 същест 18	0.044 същест 35	0.033 същест 4
20	същест	същест	0.082 същест 17	0.081 същест 1	0.070 същест 27	0.053 същест 48	0.044 същест 3	0.039 същест 51
24	Habitants	Habitants	0.110 Habitants 26	0.102 Habitants 0	0.058 Habitants 52	0.038 Habitants 58	0.029 Habitants 4	0.026 Habitants 21
27	community	community	0.313 community 1	0.150 community 63	0.118 community 19	0.071 community 29	0.040 community 53	0.026 community 21

Input: "The ancient oak tree in the center of the town square has witnessed centuries of history, joy, and sorrow, standing as a silent guardian of the ***community***"

Layer	layer output	routed exp + res	top exp 1 + res	top exp 2 + res	top exp 3 + res	top exp 4 + res	top exp 5 + res	top exp 6 + res
0	设法							
2			0.092 设法 33	0.084 CODEGEN 63	0.084 44	0.070 40	0.066 37	0.041 39
6	设法	设法	0.173 设法 32	0.070 �� 38	0.044 设法 42	0.035 设法 12	0.031 设法 56	0.030 设法 15
8	Habitants	modernista	0.119 cmid 31	0.105 cmid 37	0.074 modernista 19	0.065 modernista 60	0.051 CLUDED 58	0.040 cmid 36
13	enser	ugosl	0.135 ugosl 58	0.092 ugosl 24	0.088 ugosl 44	0.070 ugosl 40	0.065 ugosl 42	0.042 �� 35
18	Habitants	Habitants	0.277 Habitants 56	0.197 Habitants 10	0.073 Habitants 5	0.054 Habitants 63	0.039 Habitants 51	0.034 Habitants 21
21	bench	bench	0.114 bench 5	0.104 bench 46	0.066 bench 41	0.063 bench 13	0.056 bench 21	0.053 bench 18
22	ExternalTask	bench	0.157 bench 35	0.124 ExternalTask 61	0.115 bench 14	0.085 bench 2	0.064 bench 45	0.050 bench 30
25	banc	banc	0.204 banc 25	0.142 banc 4	0.139 banc 12	0.065 banc 37	0.056 banc 9	0.038 banc 38
27	banc	banc	0.555 banc 33	0.177 banc 63	0.026 banc 53	0.026 banc 20	0.026 banc 1	0.016 banc 38

Input: "Les enfants jouent dans le jardin public pendant que leurs parents discutent tranquillement sur un ***banc***"

Figure 2.1-2: Logit Lens visualization for DeepSeek-MoE on the three different input sequences. Each cell shows the top-1 token prediction 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.

3. Cosine similarity and perplexity

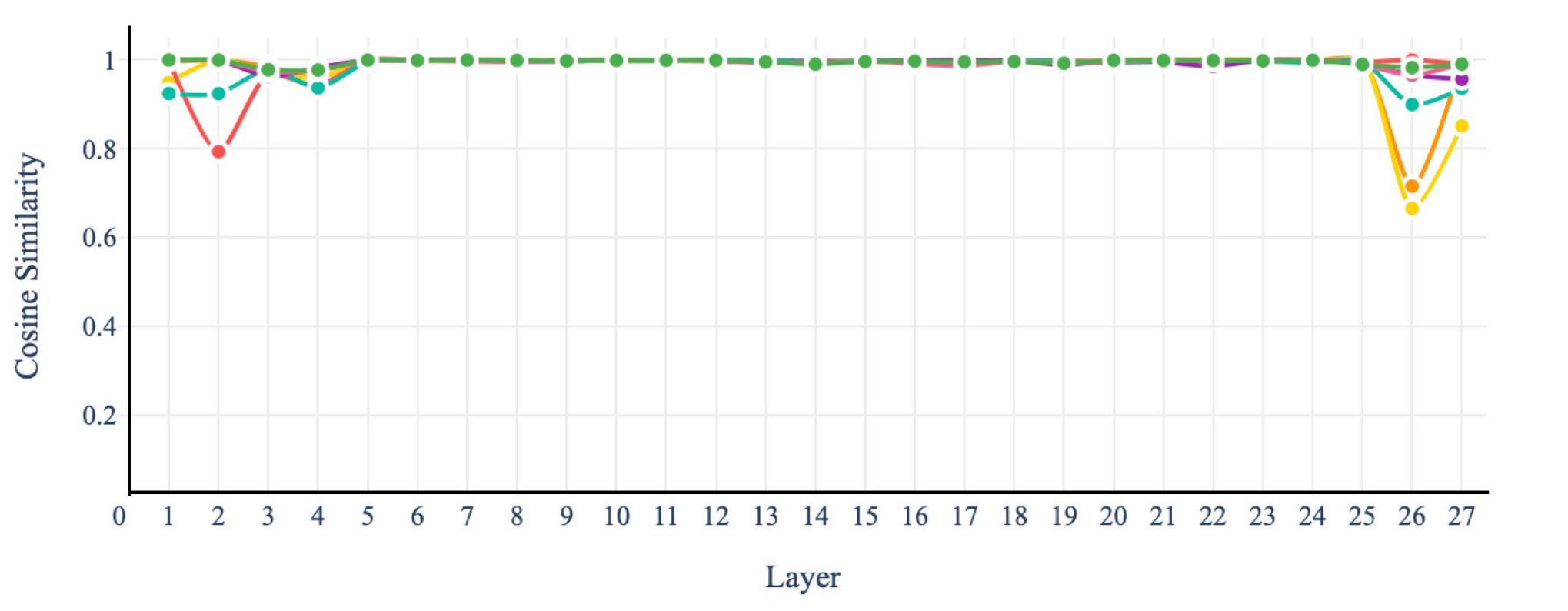


Figure 3: Cosine similarity between the hidden states of $\mathbf{H}_t^{\ell_1}$ and $\mathbf{H}_t^{\ell_6}$ across all 27 layers shows consistently high alignment

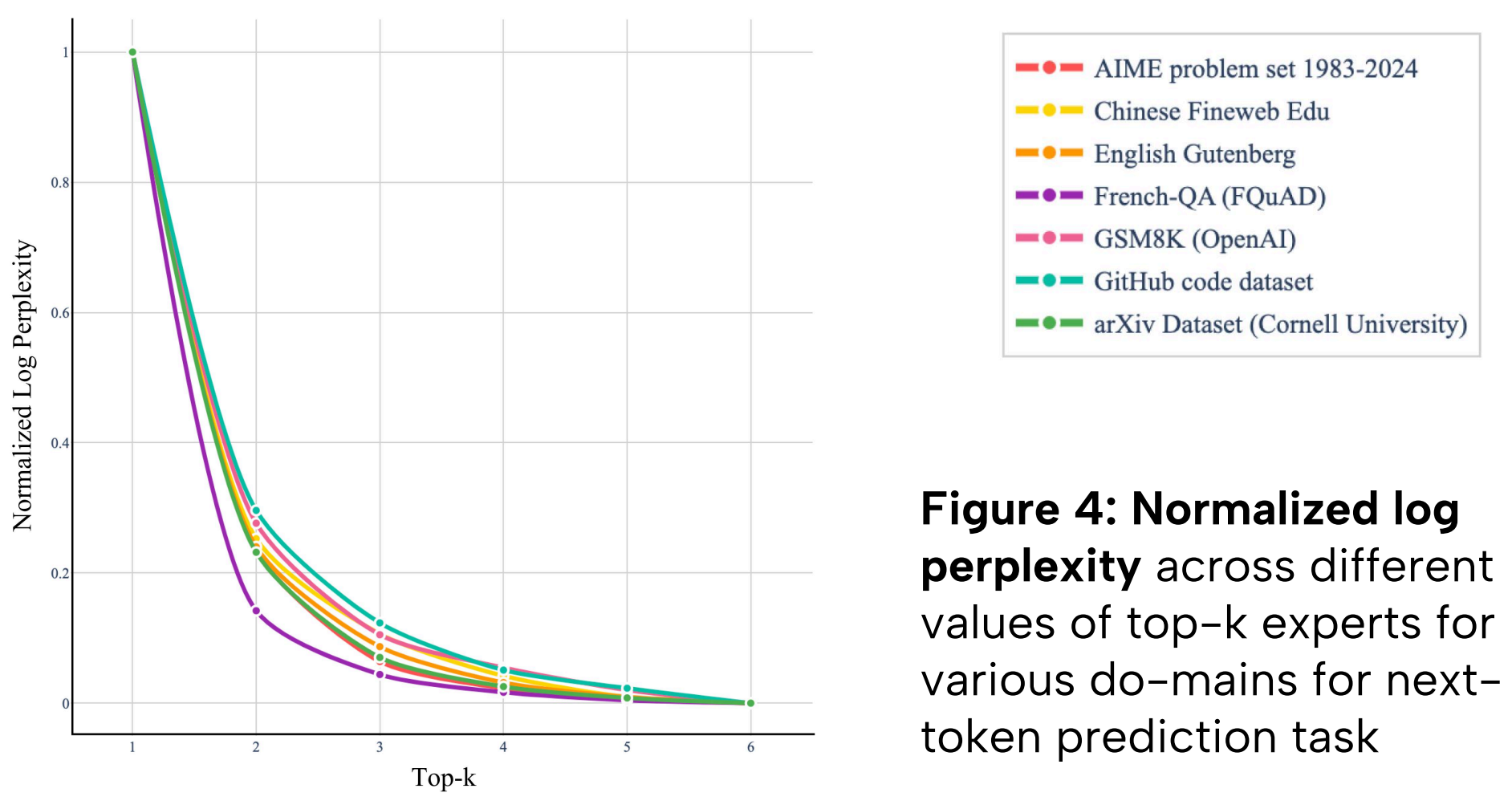


Figure 4: Normalized log perplexity across different values of top-k experts for various do-mains for next-token prediction task

We also observe very high cosine similarity between $H_t^{\ell_1}$ and $H_t^{\ell_6}$ across all layers and each domain 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.

The perplexity moderately increases when reducing $\text{Top-k} = 6 \rightarrow 1$ which validates the claim that the top-weighted expert, when combined with the residual stream, produces representations closely aligned with the layer output.