Mixtral of Experts

**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, Gianna Lengyel, Guillaume Bour,
Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux,
Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao,
Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, William El Sayed**

Abstract

We introduce Mixtral 8x7B, a Sparse Mixture of Experts (SMoE) language model. Mixtral has the same architecture as Mistral 7B, with the difference that each layer is composed of 8 feedforward blocks (i.e. experts). For every token, at each layer, a router network selects two experts to process the current state and combine their outputs. Even though each token only sees two experts, the selected experts can be different at each timestep. As a result, each token has access to 47B parameters, but only uses 13B active parameters during inference. Mixtral was trained with a context size of 32k tokens and it outperforms or matches Llama 2 70B and GPT-3.5 across all evaluated benchmarks. In particular, Mixtral vastly outperforms Llama 2 70B on mathematics, code generation, and multilingual benchmarks. We also provide a model fine-tuned to follow instructions, Mixtral 8x7B – Instruct, that surpasses GPT-3.5 Turbo, Claude-2.1, Gemini Pro, and Llama 2 70B – chat model on human benchmarks. Both the base and instruct models are released under the Apache 2.0 license.

Code: https://github.com/mistralai/mistral-src
Webpage: https://mistral.ai/news/mixtral-of-experts/

1 Introduction

In this paper, we present Mixtral 8x7B, a sparse mixture of experts model (SMoE) with open weights, licensed under Apache 2.0. Mixtral outperforms Llama 2 70B and GPT-3.5 on most benchmarks. As it only uses a subset of its parameters for every token, Mixtral allows faster inference speed at low batch-sizes, and higher throughput at large batch-sizes.

Mixtral is a sparse mixture-of-experts network. It is a decoder-only model where the feedforward block picks from a set of 8 distinct groups of parameters. At every layer, for every token, a router network chooses two of these groups (the "experts") to process the token and combine their output additively. This technique increases the number of parameters of a model while controlling cost and latency, as the model only uses a fraction of the total set of parameters per token.

Mixtral is pretrained with multilingual data using a context size of 32k tokens. It either matches or exceeds the performance of Llama 2 70B and GPT-3.5, over several benchmarks. In particular,

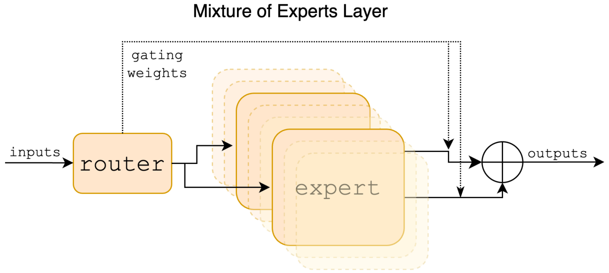


Figure 1: Mixture of Experts Layer. Each input vector is assigned to 2 of the 8 experts by a router. The layer's output is the weighted sum of the outputs of the two selected experts. In Mixtral, an expert is a standard feedforward block as in a vanilla transformer architecture.

Mixtral demonstrates superior capabilities in mathematics, code generation, and tasks that require multilingual understanding, significantly outperforming Llama 2 70B in these domains. Experiments show that Mixtral is able to successfully retrieve information from its context window of 32k tokens, regardless of the sequence length and the location of the information in the sequence.

We also present Mixtral 8x7B – Instruct, a chat model fine-tuned to follow instructions using supervised fine-tuning and Direct Preference Optimization [25]. Its performance notably surpasses that of GPT-3.5 Turbo, Claude-2.1, Gemini Pro, and Llama 2 70B – chat model on human evaluation benchmarks. Mixtral – Instruct also demonstrates reduced biases, and a more balanced sentiment profile in benchmarks such as BBQ, and BOLD.

We release both Mixtral 8x7B and Mixtral 8x7B – Instruct under the Apache 2.0 license[1], free for academic and commercial usage, ensuring broad accessibility and potential for diverse applications. To enable the community to run Mixtral with a fully open-source stack, we submitted changes to the vLLM project, which integrates Megablocks CUDA kernels for efficient inference. Skypilot also allows the deployment of vLLM endpoints on any instance in the cloud.

2 Architectural details

Mixtral is based on a transformer architecture [31] and uses the same modifications as described in [18], with the notable exceptions that Mixtral supports a fully dense context length of 32k tokens, and the feedforward blocks are replaced by Mixture-of-Expert layers (Section 2.1). The model architecture parameters are summarized in Table 1.

Parameter	Value
`dim`	4096
`n_layers`	32
`head_dim`	128
`hidden_dim`	14336
`n_heads`	32
`n_kv_heads`	8
`context_len`	32768
`vocab_size`	32000
`num_experts`	8
`top_k_experts`	2

Table 1: Model architecture.

2.1 Sparse Mixture of Experts

We present a brief overview of the Mixture of Experts layer (Figure 1). For a more in-depth overview, see [12]. The output of the MoE module for a given input x is determined by the weighted sum of the outputs of the expert networks, where the weights are given by the gating network's output. i.e. given n expert networks $\{E_0, E_i, ..., E_{n-1}\}$, the output of the expert layer is given by:

$$\sum_{i=0}^{n-1} G(x)_i \cdot E_i(x).$$

Here, $G(x)_i$ denotes the n-dimensional output of the gating network for the i-th expert, and $E_i(x)$ is the output of the i-th expert network. If the gating vector is sparse, we can avoid computing the outputs of experts whose gates are zero. There are multiple alternative ways of implementing $G(x)$ [6, 15, 35], but a simple and performant one is implemented by taking the softmax over the Top-K logits of a linear layer [28]. We use

$$G(x) := \text{Softmax}(\text{TopK}(x \cdot W_g)),$$

where $(\text{TopK}(\ell))_i := \ell_i$ if ℓ_i is among the top-K coordinates of logits $\ell \in \mathbb{R}^n$ and $(\text{TopK}(\ell))_i := -\infty$ otherwise. The value of K – the number of experts used per token – is a hyper-parameter that modulates the amount of compute used to process each token. If one increases n while keeping K fixed, one

[1] https://mistral.ai/news/mixtral-of-experts/

can increase the model's parameter count while keeping its computational cost effectively constant. This motivates a distinction between the model's total parameter count (commonly referenced as the **sparse** parameter count), which grows with n, and the number of parameters used for processing an individual token (called the **active** parameter count), which grows with K up to n.

MoE layers can be run efficiently on single GPUs with high performance specialized kernels. For example, Megablocks [13] casts the feed-forward network (FFN) operations of the MoE layer as large sparse matrix multiplications, significantly enhancing the execution speed and naturally handling cases where different experts get a variable number of tokens assigned to them. Moreover, the MoE layer can be distributed to multiple GPUs through standard Model Parallelism techniques, and through a particular kind of partitioning strategy called Expert Parallelism (EP) [28]. During the MoE layer's execution, tokens meant to be processed by a specific expert are routed to the corresponding GPU for processing, and the expert's output is returned to the original token location. Note that EP introduces challenges in load balancing, as it is essential to distribute the workload evenly across the GPUs to prevent overloading individual GPUs or hitting computational bottlenecks.

In a Transformer model, the MoE layer is applied independently per token and replaces the feed-forward (FFN) sub-block of the transformer block. For Mixtral we use the same SwiGLU architecture as the expert function $E_i(x)$ and set $K = 2$. This means each token is routed to two SwiGLU sub-blocks with different sets of weights. Taking this all together, the output y for an input token x is computed as:

$$y = \sum_{i=0}^{n-1} \text{Softmax}(\text{Top2}(x \cdot W_g))_i \cdot \text{SwiGLU}_i(x).$$

This formulation is similar to the GShard architecture [21], with the exceptions that we replace all FFN sub-blocks by MoE layers while GShard replaces every other block, and that GShard uses a more elaborate gating strategy for the second expert assigned to each token.

3 Results

We compare Mixtral to Llama, and re-run all benchmarks with our own evaluation pipeline for fair comparison. We measure performance on a wide variety of tasks categorized as follow:

- **Commonsense Reasoning (0-shot):** Hellaswag [32], Winogrande [26], PIQA [3], SIQA [27], OpenbookQA [22], ARC-Easy, ARC-Challenge [8], CommonsenseQA [30]
- **World Knowledge (5-shot):** NaturalQuestions [20], TriviaQA [19]
- **Reading Comprehension (0-shot):** BoolQ [7], QuAC [5]
- **Math:** GSM8K [9] (8-shot) with maj@8 and MATH [17] (4-shot) with maj@4
- **Code:** Humaneval [4] (0-shot) and MBPP [1] (3-shot)
- **Popular aggregated results:** MMLU [16] (5-shot), BBH [29] (3-shot), and AGI Eval [34] (3-5-shot, English multiple-choice questions only)

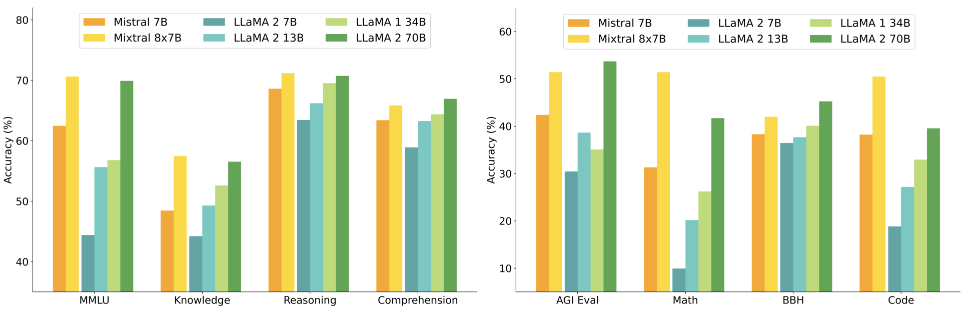


Figure 2: Performance of Mixtral and different Llama models on a wide range of benchmarks. All models were re-evaluated on all metrics with our evaluation pipeline for accurate comparison. Mixtral outperforms or matches Llama 2 70B on all benchmarks. In particular, it is vastly superior in mathematics and code generation.

Model	Active Params	MMLU	HellaS	WinoG	PIQA	Arc-e	Arc-c	NQ	TriQA	HumanE	MBPP	Math	GSM8K
LLaMA 2 7B	7B	44.4%	77.1%	69.5%	77.9%	68.7%	43.2%	17.5%	56.6%	11.6%	26.1%	3.9%	16.0%
LLaMA 2 13B	13B	55.6%	80.7%	72.9%	80.8%	75.2%	48.8%	16.7%	64.0%	18.9%	35.4%	6.0%	34.3%
LLaMA 1 33B	33B	56.8%	83.7%	76.2%	82.2%	79.6%	54.4%	24.1%	68.5%	25.0%	40.9%	8.4%	44.1%
LLaMA 2 70B	70B	69.9%	**85.4%**	**80.4%**	82.6%	79.9%	56.5%	25.4%	**73.0%**	29.3%	49.8%	13.8%	69.6%
Mistral 7B	7B	62.5%	81.0%	74.2%	82.2%	80.5%	54.9%	23.2%	62.5%	26.2%	50.2%	12.7%	50.0%
Mixtral 8x7B	13B	**70.6%**	84.4%	77.2%	**83.6%**	**83.1%**	**59.7%**	**30.6%**	71.5%	**40.2%**	**60.7%**	**28.4%**	**74.4%**

Table 2: Comparison of Mixtral with Llama. Mixtral outperforms or matches Llama 2 70B performance on almost all popular benchmarks while using 5x fewer active parameters during inference.

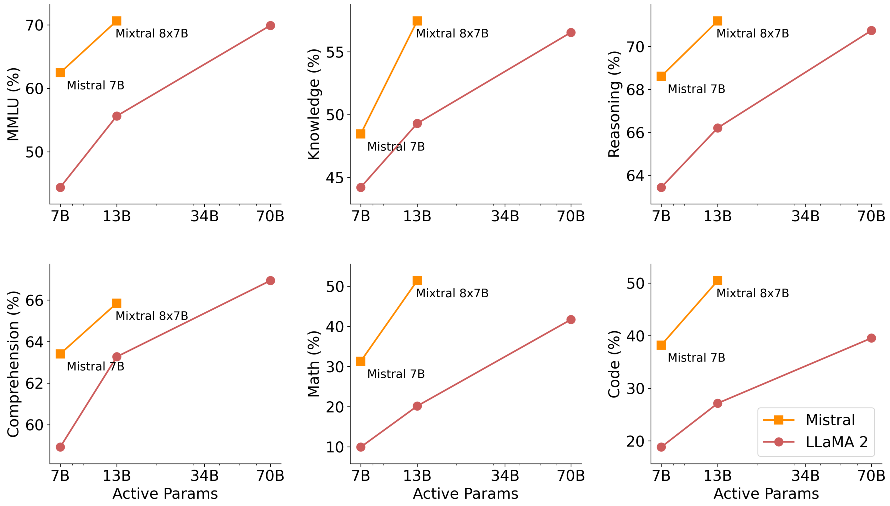


Figure 3: Results on MMLU, commonsense reasoning, world knowledge and reading comprehension, math and code for Mistral (7B/8x7B) vs Llama 2 (7B/13B/70B). Mixtral largely outperforms Llama 2 70B on all benchmarks, except on reading comprehension benchmarks while using 5x lower active parameters. It is also vastly superior to Llama 2 70B on code and math.

Detailed results for Mixtral, Mistral 7B and Llama 2 7B/13B/70B and Llama 1 34B[2] are reported in Table 2. Figure 2 compares the performance of Mixtral with the Llama models in different categories. Mixtral surpasses Llama 2 70B across most metrics. In particular, Mixtral displays a superior performance in code and mathematics benchmarks.

Size and Efficiency. We compare our performance to the Llama 2 family, aiming to understand Mixtral models' efficiency in the cost-performance spectrum (see Figure 3). As a sparse Mixture-of-Experts model, Mixtral only uses 13B active parameters for each token. With 5x lower active parameters, Mixtral is able to outperform Llama 2 70B across most categories.

Note that this analysis focuses on the active parameter count (see Section 2.1), which is directly proportional to the inference compute cost, but does not consider the memory costs and hardware utilization. The memory costs for serving Mixtral are proportional to its *sparse* parameter count, 47B, which is still smaller than Llama 2 70B. As for device utilization, we note that the SMoEs layer introduces additional overhead due to the routing mechanism and due to the increased memory loads when running more than one expert per device. They are more suitable for batched workloads where one can reach a good degree of arithmetic intensity.

Comparison with Llama 2 70B and GPT-3.5. In Table 3, we report the performance of Mixtral 8x7B compared to Llama 2 70B and GPT-3.5. We observe that Mixtral performs similarly or above the two other models. On MMLU, Mixtral obtains a better performance, despite its significantly smaller capacity (47B tokens compared to 70B). For MT Bench, we report the performance of the latest GPT-3.5-Turbo model available, `gpt-3.5-turbo-1106`.

[2] Since Llama 2 34B was not open-sourced, we report results for Llama 1 34B.

	LLaMA 2 70B	**GPT-3.5**	**Mixtral 8x7B**
MMLU (MCQ in 57 subjects)	69.9%	70.0%	**70.6%**
HellaSwag (10-shot)	**87.1%**	85.5%	86.7%
ARC Challenge (25-shot)	85.1%	85.2%	**85.8%**
WinoGrande (5-shot)	**83.2%**	81.6%	81.2%
MBPP (pass@1)	49.8%	52.2%	**60.7%**
GSM-8K (5-shot)	53.6%	57.1%	**58.4%**
MT Bench (for Instruct Models)	6.86	**8.32**	8.30

Table 3: Comparison of Mixtral with Llama 2 70B and GPT-3.5. Mixtral outperforms or matches Llama 2 70B and GPT-3.5 performance on most metrics.

Evaluation Differences. On some benchmarks, there are some differences between our evaluation protocol and the one reported in the Llama 2 paper: 1) on MBPP, we use the hand-verified subset 2) on TriviaQA, we do not provide Wikipedia contexts.

3.1 Multilingual benchmarks

Compared to Mistral 7B, we significantly upsample the proportion of multilingual data during pretraining. The extra capacity allows Mixtral to perform well on multilingual benchmarks while maintaining a high accuracy in English. In particular, Mixtral significantly outperforms Llama 2 70B in French, German, Spanish, and Italian, as shown in Table 4.

	Active		**French**			**German**			**Spanish**			**Italian**	
Model	**Params**	Arc-c	HellaS	MMLU	Arc-c	HellaS	MMLU	Arc-c	HellaS	MMLU	Arc-c	HellaS	MMLU
LLaMA 1 33B	33B	39.3%	68.1%	49.9%	41.1%	63.3%	48.7%	45.7%	69.8%	52.3%	42.9%	65.4%	49.0%
LLaMA 2 70B	70B	49.9%	72.5%	64.3%	47.3%	68.7%	64.2%	50.5%	74.5%	66.0%	49.4%	70.9%	65.1%
Mixtral 8x7B	13B	**58.2%**	**77.4%**	**70.9%**	**54.3%**	**73.0%**	**71.5%**	**55.4%**	**77.6%**	**72.5%**	**52.8%**	**75.1%**	**70.9%**

Table 4: Comparison of Mixtral with Llama on Multilingual Benchmarks. On ARC Challenge, Hellaswag, and MMLU, Mixtral outperforms Llama 2 70B on 4 languages: French, German, Spanish, and Italian.

3.2 Long range performance

To assess the capabilities of Mixtral to tackle long context, we evaluate it on the passkey retrieval task introduced in [23], a synthetic task designed to measure the ability of the model to retrieve a passkey inserted randomly in a long prompt. Results in Figure 4 (Left) show that Mixtral achieves a 100% retrieval accuracy regardless of the context length or the position of passkey in the sequence. Figure 4 (Right) shows that the perplexity of Mixtral on a subset of the proof-pile dataset [2] decreases monotonically as the size of the context increases.

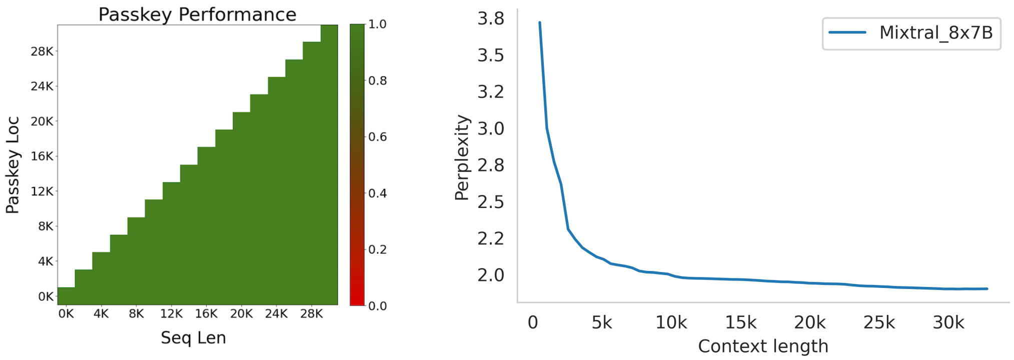


Figure 4: Long range performance of Mixtral. (Left) Mixtral has 100% retrieval accuracy of the Passkey task regardless of the location of the passkey and length of the input sequence. (Right) The perplexity of Mixtral on the proof-pile dataset decreases monotonically as the context length increases.

3.3 Bias Benchmarks

To identify possible flaws to be corrected by fine-tuning / preference modeling, we measure the base model performance on Bias Benchmark for QA (BBQ) [24] and Bias in Open-Ended Language Generation Dataset (BOLD) [10]. BBQ is a dataset of hand-written question sets that target attested social biases against nine different socially-relevant categories: age, disability status, gender identity, nationality, physical appearance, race/ethnicity, religion, socio-economic status, sexual orientation. BOLD is a large-scale dataset that consists of 23,679 English text generation prompts for bias benchmarking across five domains.

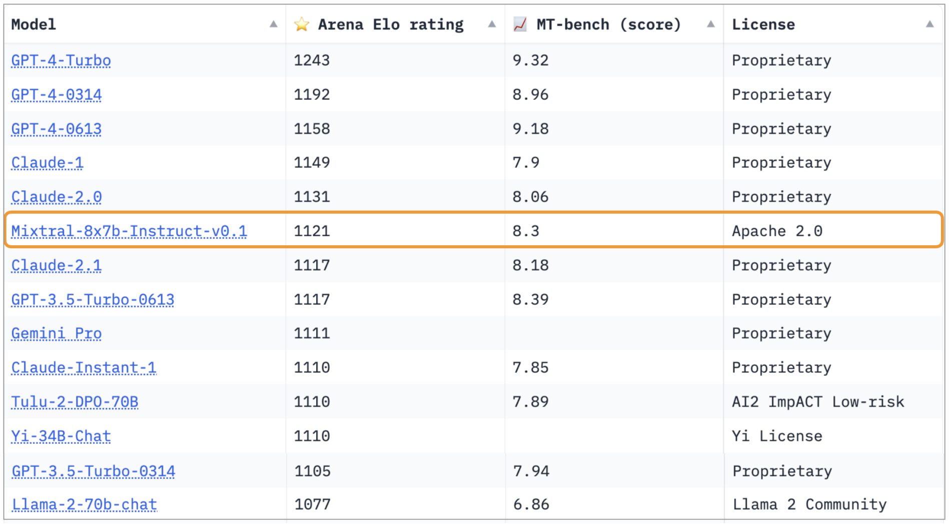
	Llama 2 70B	**Mixtral 8x7B**
BBQ accuracy	51.5%	56.0%
BOLD sentiment score (avg ± std)		
gender	0.293 ± 0.073	0.323 ±0.045
profession	0.218 ± 0.073	0.243 ± 0.087
religious_ideology	0.188 ± 0.133	0.144 ± 0.089
political_ideology	0.149 ± 0.140	0.186 ± 0.146
race	0.232 ± 0.049	0.232 ± 0.052

Figure 5: Bias Benchmarks. Compared Llama 2 70B, Mixtral presents less bias (higher accuracy on BBQ, lower std on BOLD) and displays more positive sentiment (higher avg on BOLD).

We benchmark Llama 2 and Mixtral on BBQ and BOLD with our evaluation framework and report the results in Table 5. Compared to Llama 2, Mixtral presents less bias on the BBQ benchmark (56.0% vs 51.5%). For each group in BOLD, a higher average sentiment score means more positive sentiments and a lower standard deviation indicates less bias within the group. Overall, Mixtral displays more positive sentiments than Llama 2, with similar variances within each group.

4 Instruction Fine-tuning

We train Mixtral – Instruct using supervised fine-tuning (SFT) on an instruction dataset followed by Direct Preference Optimization (DPO) [25] on a paired feedback dataset. Mixtral – Instruct reaches a score of 8.30 on MT-Bench [33] (see Table 2), making it the best open-weights model as of December 2023. Independent human evaluation conducted by LMSys is reported in Figure 6[3] and shows that Mixtral – Instruct outperforms GPT-3.5-Turbo, Gemini Pro, Claude-2.1, and Llama 2 70B chat.



Model ▲	⭐ Arena Elo rating ▲	📈 MT-bench (score) ▲	License ▲
GPT-4-Turbo	1243	9.32	Proprietary
GPT-4-0314	1192	8.96	Proprietary
GPT-4-0613	1158	9.18	Proprietary
Claude-1	1149	7.9	Proprietary
Claude-2.0	1131	8.06	Proprietary
Mixtral-8x7b-Instruct-v0.1	1121	8.3	Apache 2.0
Claude-2.1	1117	8.18	Proprietary
GPT-3.5-Turbo-0613	1117	8.39	Proprietary
Gemini Pro	1111		Proprietary
Claude-Instant-1	1110	7.85	Proprietary
Tulu-2-DPO-70B	1110	7.89	AI2 ImpACT Low-risk
Yi-34B-Chat	1110		Yi License
GPT-3.5-Turbo-0314	1105	7.94	Proprietary
Llama-2-70b-chat	1077	6.86	Llama 2 Community

Figure 6: LMSys Leaderboard. (Screenshot from Dec 22, 2023) Mixtral 8x7B Instruct v0.1 achieves an Arena Elo rating of 1121 outperforming Claude-2.1 (1117), all versions of GPT-3.5-Turbo (1117 best), Gemini Pro (1111), and Llama-2-70b-chat (1077). Mixtral is currently the best open-weights model by a large margin.

[3] https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard

5 Routing analysis

In this section, we perform a small analysis on the expert selection by the router. In particular, we are interested to see if during training some experts specialized to some specific domains (e.g. mathematics, biology, philosophy, etc.).

To investigate this, we measure the distribution of selected experts on different subsets of The Pile validation dataset [14]. Results are presented in Figure 7, for layers 0, 15, and 31 (layers 0 and 31 respectively being the first and the last layers of the model). Surprisingly, we do not observe obvious patterns in the assignment of experts based on the topic. For instance, at all layers, the distribution of expert assignment is very similar for ArXiv papers (written in Latex), for biology (PubMed Abstracts), and for Philosophy (PhilPapers) documents.

Only for DM Mathematics we note a marginally different distribution of experts. This divergence is likely a consequence of the dataset's synthetic nature and its limited coverage of the natural language spectrum, and is particularly noticeable at the first and last layers, where the hidden states are very correlated to the input and output embeddings respectively.

This suggests that the router does exhibit some structured syntactic behavior. Figure 8 shows examples of text from different domains (Python code, mathematics, and English), where each token is highlighted with a background color corresponding to its selected expert. The figure shows that words such as 'self' in Python and 'Question' in English often get routed through the same expert even though they involve multiple tokens. Similarly, in code, the indentation tokens are always assigned to the same experts, particularly at the first and last layers where the hidden states are more correlated to the input and output of the model.

We also note from Figure 8 that consecutive tokens are often assigned the same experts. In fact, we observe some degree of positional locality in The Pile datasets. Table 5 shows the proportion of consecutive tokens that get the same expert assignments per domain and layer. The proportion of repeated

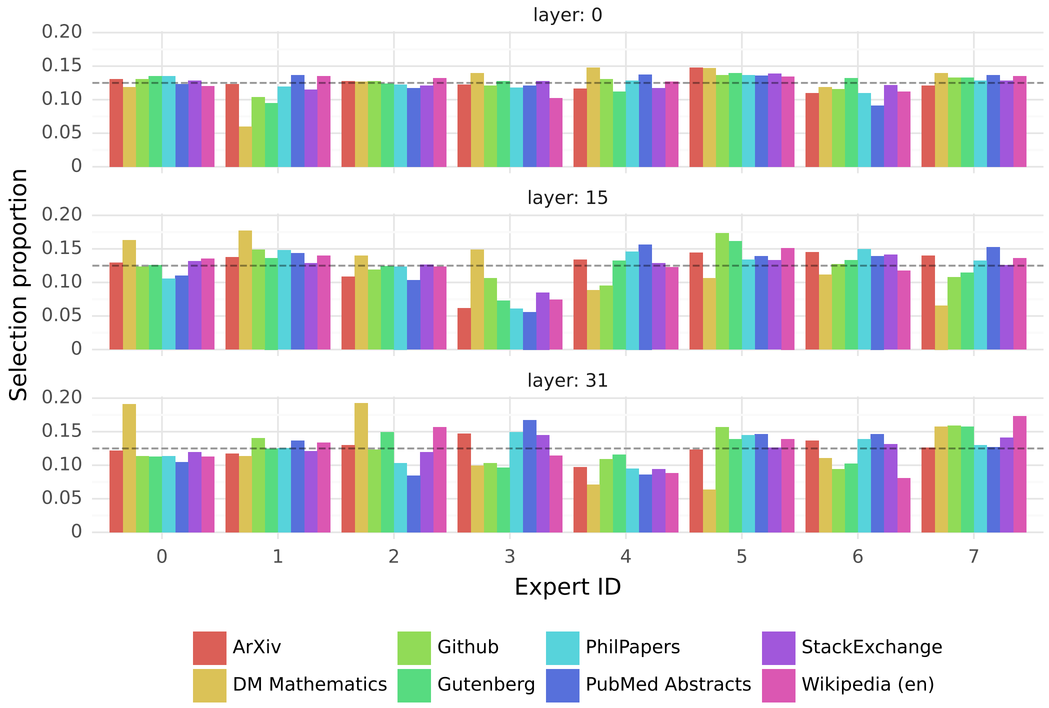


Figure 7: Proportion of tokens assigned to each expert on different domains from The Pile dataset for layers 0, 15, and 31. The gray dashed vertical line marks $1/8$, i.e. the proportion expected with uniform sampling. Here, we consider experts that are either selected as a first or second choice by the router. A breakdown of the proportion of assignments done in each case cane be seen in Figure 9 in the Appendix.

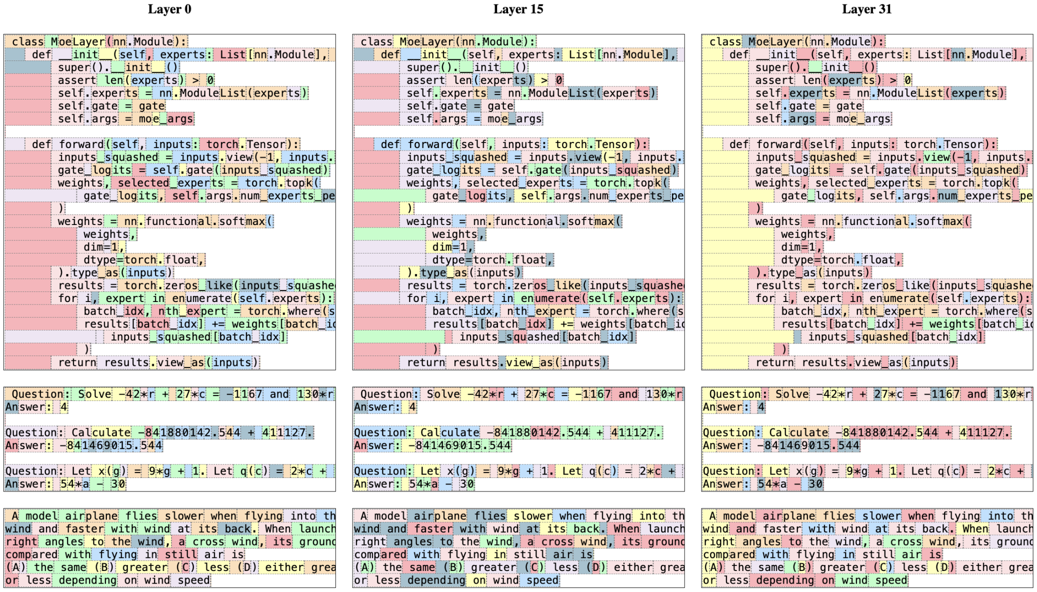
	First choice			First or second choice		
	Layer 0	Layer 15	Layer 31	Layer 0	Layer 15	Layer 31
ArXiv	14.0%	27.9%	22.7%	46.5%	62.3%	52.9%
DM Mathematics	14.1%	28.4%	19.7%	44.9%	67.0%	44.5%
Github	14.9%	28.1%	19.7%	49.9%	66.9%	49.2%
Gutenberg	13.9%	26.1%	26.3%	49.5%	63.1%	52.2%
PhilPapers	13.6%	25.3%	22.1%	46.9%	61.9%	51.3%
PubMed Abstracts	14.2%	24.6%	22.0%	48.6%	61.6%	51.8%
StackExchange	13.6%	27.2%	23.6%	48.2%	64.6%	53.6%
Wikipedia (en)	14.4%	23.6%	25.3%	49.8%	62.1%	51.8%

Table 5: Percentage of expert assignment repetitions. We evaluate the proportion of times the same expert is assigned to a token i and its following token $i+1$. We report whether the first chosen expert is the same, or whether the same expert is observed as first or second choice in consecutive tokens. For reference, the expected proportion of repetitions in the case of random assignments is $\frac{1}{8} = 12.5\%$ for "First choice" and $1 - \frac{6}{8}\frac{5}{7} \approx 46\%$ for "First and second choice". Repetitions at the first layer are close to random, but are significantly higher at layers 15 and 31. The high number of repetitions shows that expert choice exhibits high temporal locality at these layers.

consecutive assignments is significantly higher than random for higher layers. This has implications in how one might optimize the model for fast training and inference. For example, cases with high locality are more likely to cause over-subscription of certain experts when doing Expert Parallelism. Conversely, this locality can be leveraged for caching, as is done in [11]. A more complete view of these same expert frequency is provided for all layers and across datasets in Figure 10 in the Appendix.

6 Conclusion

In this paper, we introduced Mixtral 8x7B, the first mixture-of-experts network to reach a state-of-the-art performance among open-source models. Mixtral 8x7B Instruct outperforms Claude-2.1, Gemini Pro, and GPT-3.5 Turbo on human evaluation benchmarks. Because it only uses two experts at each time step, Mixtral only uses 13B active parameters per token while outperforming the previous best model using 70B parameters per token (Llama 2 70B). We are making our trained and fine-tuned models publicly available under the Apache 2.0 license. By sharing our models, we aim to facilitate the development of new techniques and applications that can benefit a wide range of industries and domains.



```
class MoeLayer(nn.Module):
    def __init__(self, experts: List[nn.Module],
        super().__init__()
        assert len(experts) > 0
        self.experts = nn.ModuleList(experts)
        self.gate = gate
        self.args = moe_args

    def forward(self, inputs: torch.Tensor):
        inputs_squashed = inputs.view(-1, inputs.
        gate_logits = self.gate(inputs_squashed)
        weights, selected_experts = torch.topk(
            gate_logits, self.args.num_experts_pe
        )
        weights = nn.functional.softmax(
            weights,
            dim=1,
            dtype=torch.float,
        ).type_as(inputs)
        results = torch.zeros_like(inputs_squashe
        for i, expert in enumerate(self.experts):
            batch_idx, nth_expert = torch.where(s
            results[batch_idx] += weights[batch_id
                inputs_squashed[batch_idx]
            )
        return results.view_as(inputs)
```

```
Question: Solve -42*r + 27*c = -1167 and 130*r
Answer: 4

Question: Calculate -841880142.544 + 411127.
Answer: -841469015.544

Question: Let x(g) = 9*g + 1. Let q(c) = 2*c +
Answer: 54*a - 30
```

```
A model airplane flies slower when flying into th
wind and faster with wind at its back. When launch
right angles to the wind, a cross wind, its ground
compared with flying in still air is
(A) the same (B) greater (C) less (D) either grea
or less depending on wind speed
```

Figure 8: Text samples where each token is colored with the first expert choice. The selection of experts appears to be more aligned with the syntax rather than the domain, especially at the initial and final layers.

Acknowledgements

We thank the CoreWeave and Scaleway teams for technical support as we trained our models. We are grateful to NVIDIA for supporting us in integrating TensorRT-LLM and Triton and working alongside us to make a sparse mixture of experts compatible with TensorRT-LLM.

References

[1] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

[2] Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*, 2023.

[3] Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, pages 7432–7439, 2020.

[4] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

[5] Eunsol Choi, He He, Mohit Iyyer, Mark Yatskar, Wen-tau Yih, Yejin Choi, Percy Liang, and Luke Zettlemoyer. Quac: Question answering in context. *arXiv preprint arXiv:1808.07036*, 2018.

[6] Aidan Clark, Diego De Las Casas, Aurelia Guy, Arthur Mensch, Michela Paganini, Jordan Hoffmann, Bogdan Damoc, Blake Hechtman, Trevor Cai, Sebastian Borgeaud, et al. Unified scaling laws for routed language models. In *International Conference on Machine Learning*, pages 4057–4086. PMLR, 2022.

[7] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*, 2019.

[8] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

[9] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[10] Jwala Dhamala, Tony Sun, Varun Kumar, Satyapriya Krishna, Yada Pruksachatkun, Kai-Wei Chang, and Rahul Gupta. Bold: Dataset and metrics for measuring biases in open-ended language generation. In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pages 862–872, 2021.

[11] Artyom Eliseev and Denis Mazur. Fast inference of mixture-of-experts language models with offloading. *arXiv preprint arXiv:2312.17238*, 2023.

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

[13] Trevor Gale, Deepak Narayanan, Cliff Young, and Matei Zaharia. Megablocks: Efficient sparse training with mixture-of-experts. *arXiv preprint arXiv:2211.15841*, 2022.

[14] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

[15] Hussein Hazimeh, Zhe Zhao, Aakanksha Chowdhery, Maheswaran Sathiamoorthy, Yihua Chen, Rahul Mazumder, Lichan Hong, and Ed Chi. Dselect-k: Differentiable selection in the mixture of experts with applications to multi-task learning. *Advances in Neural Information Processing Systems*, 34:29335–29347, 2021.

[16] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

[17] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[18] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[19] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[20] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, pages 453–466, 2019.

[21] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. Gshard: Scaling giant models with conditional computation and automatic sharding. *arXiv preprint arXiv:2006.16668*, 2020.

[22] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*, 2018.

[23] Amirkeivan Mohtashami and Martin Jaggi. Landmark attention: Random-access infinite context length for transformers. *arXiv preprint arXiv:2305.16300*, 2023.

[24] Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel R Bowman. Bbq: A hand-built bias benchmark for question answering. *arXiv preprint arXiv:2110.08193*, 2021.

[25] Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.

[26] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, pages 99–106, 2021.

[27] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *arXiv preprint arXiv:1904.09728*, 2019.

[28] 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*, 2017.

[29] Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

[30] Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*, 2018.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[32] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

[33] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

[34] Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. Agieval: A human-centric benchmark for evaluating foundation models. *arXiv preprint arXiv:2304.06364*, 2023.

[35] Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, et al. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114, 2022.

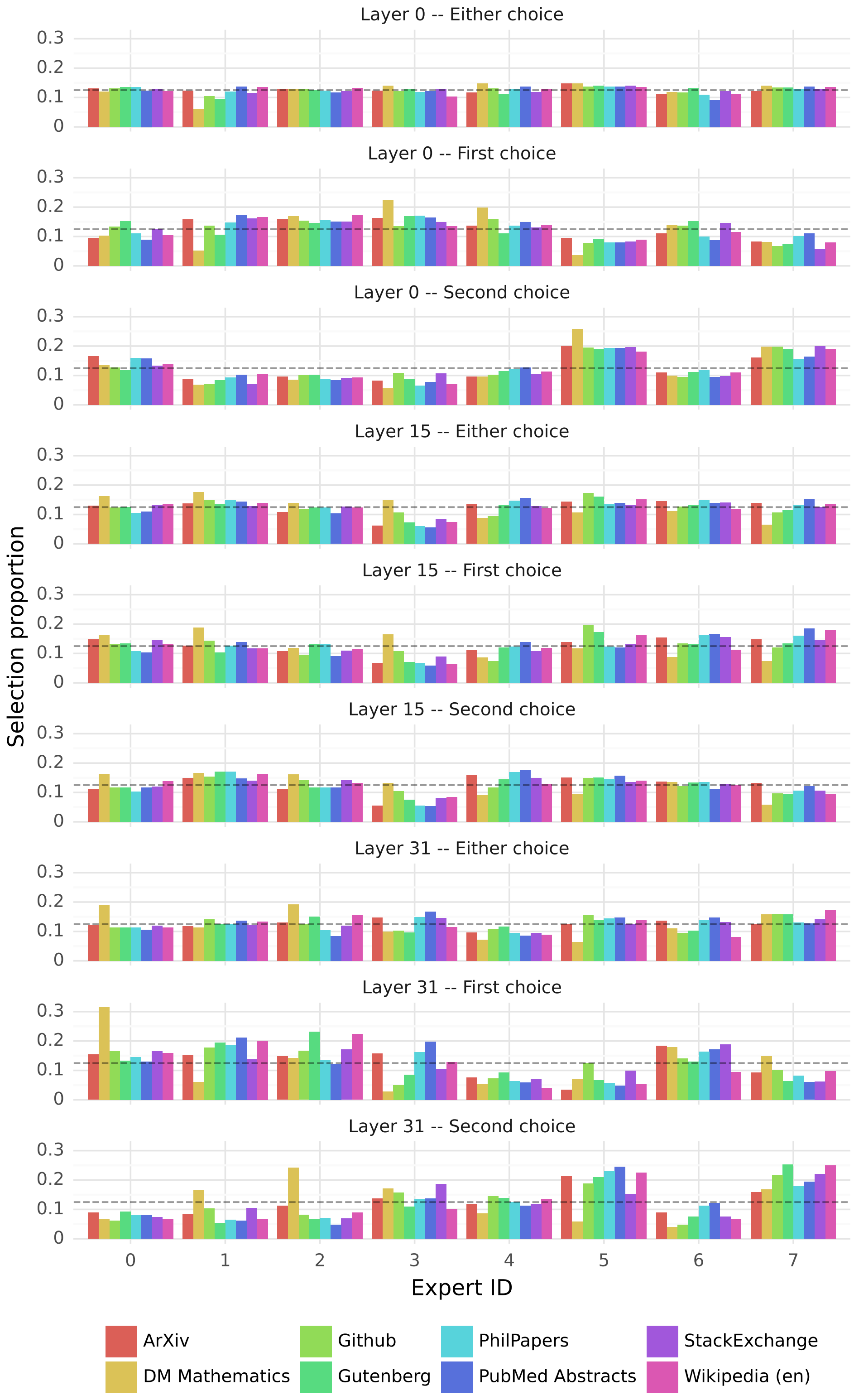


Figure 9: Proportion of tokens assigned to each expert on different subsets from The Pile dataset, separated by whether the expert was selected as first or second choice, or either. The "Either choice" case is equivalent to Figure 7. The gray dashed vertical line marks $\frac{1}{8}$, i.e. the proportion expected with uniform sampling.

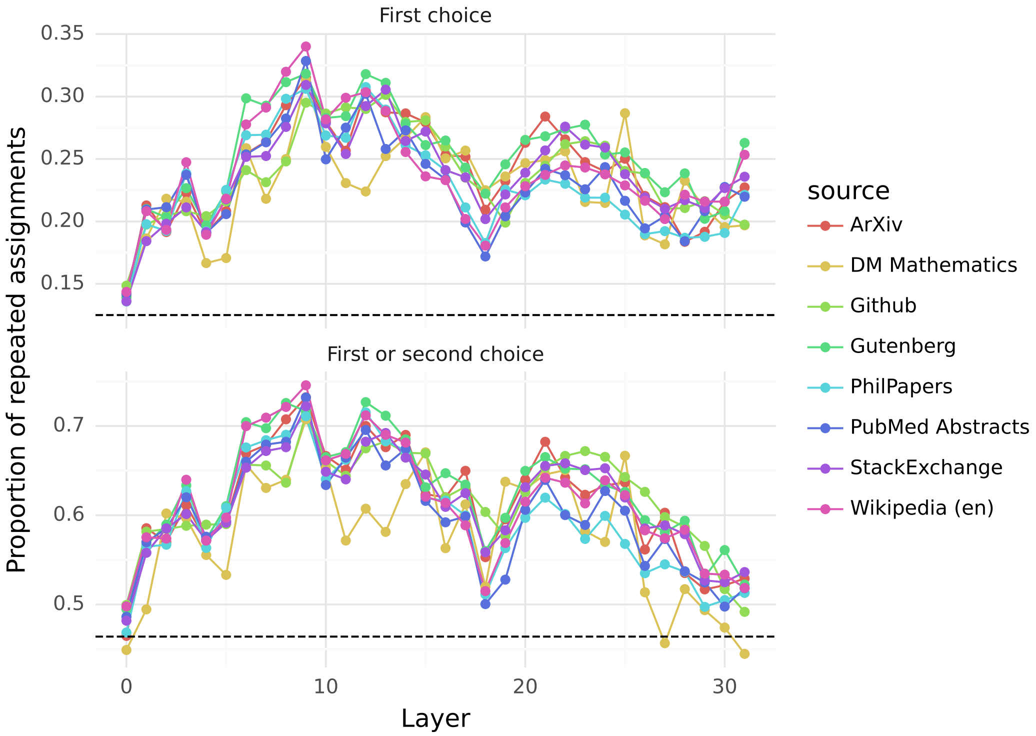


Figure 10: Repeated consecutive assignments per MoE layer. Repeated assignments occur a lot more often than they would with uniform assignments (materialized by the dashed lines). Patterns are similar across datasets with less repetitions for DM Mathematics.