MAI 5301 - Presentation
Week 5

Background

Mixture of Experts (MoE) is a neural network design where only a subset of parameters is used per input

Instead of one large model handling everything, multiple “expert” networks specialize, and a router selects which expert to use

MoE enables very large model capacity without increasing computation for every input

Sparse activation means most model parameters are inactive per token

Interest in MoE grew as dense models became computationally expensive to scale

MoE research aimed to increase model size without increasing FLOPs per token

Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity

Research Motivation

Modern language models improve with scale, but dense Transformers are computationally inefficient

Increasing parameter count dramatically increases training cost

Existing sparse approaches show promise but suffer from:

- Routing complexity
- High communication overhead
- Training instability

Research goal: Achieve massive model scale without increasing per-token computation

Core Hypothesis

Parameter count is an independent axis of scaling (Data, Computation, Parameter)

Model quality can improve without increasing FLOPs per token

Sparse activation enables:

- Trillion-parameter models
- Constant computation per example

Hypothesis: Simple, stable sparse routing can outperform dense and traditional MoE models

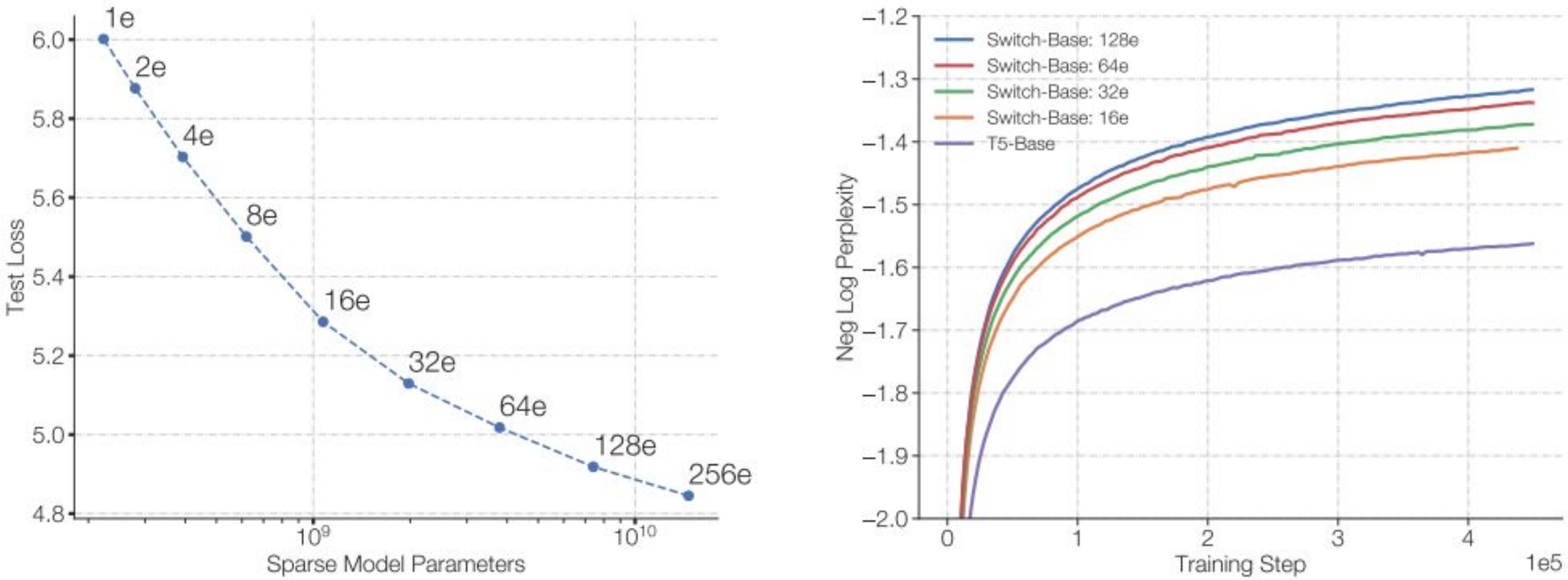


Figure 1: Scaling and sample efficiency of Switch Transformers. Left Plot: Scaling properties for increasingly sparse (more experts) Switch Transformers. Right Plot: Negative log perplexity comparing Switch Transformers to T5 (Raffel et al., 2019) models using the same compute budget.

Switch Transformer

Introduces a simplified Mixture-of-Experts architecture

Routes each token to a single expert (k = 1)

Maintains:

- Differentiable routing
- Sparse activation

Eliminates unnecessary complexity found in prior MoE designs

Feature	Prior MoE	Switch Transformer
Number of experts per token	Often 2 or more experts per input	1 expert per token
Routing complexity	More complicated: each token could go to multiple experts → more communication between machines	Simpler routing → pick just 1 expert, less communication
Training stability	Harder: using multiple experts caused imbalances (some experts got overloaded, others underused)	They added expert balancing techniques to make training stable
Efficiency / speed	Slower, because multiple experts had to run and synchronize	Faster: only 1 expert runs per token → less computation, less memory overhead
Scaling	Harder to scale beyond hundreds of billions of parameters	Designed to scale to a trillion parameters with minimal extra compute

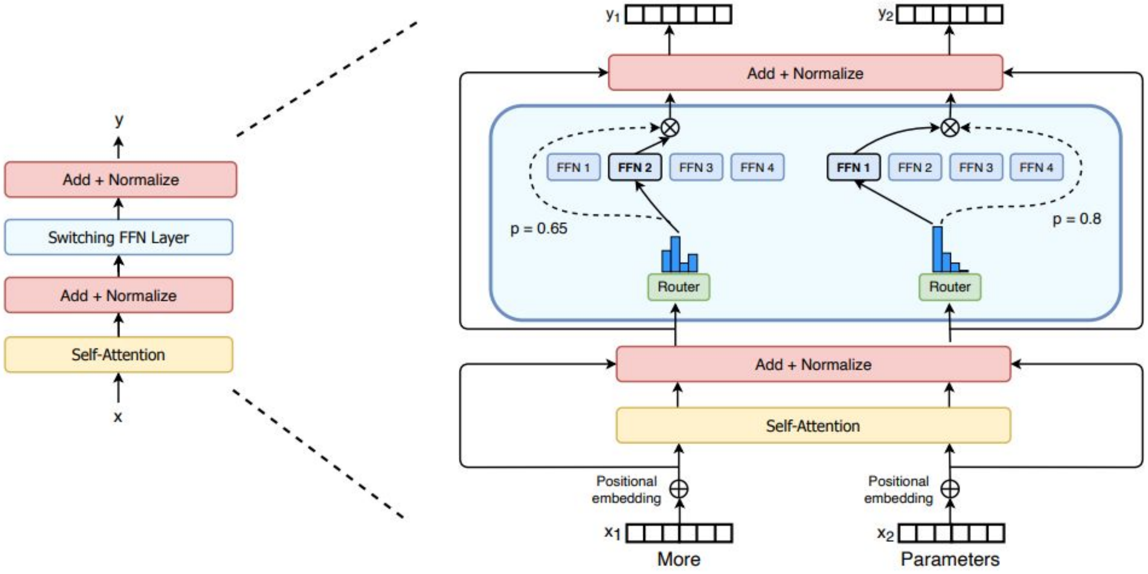


Figure 2: Illustration of a Switch Transformer encoder block. We replace the dense feed forward network (FFN) layer present in the Transformer with a sparse Switch FFN layer (light blue). The layer operates independently on the tokens in the sequence. We diagram two tokens (x_1 = "More" and x_2 = "Parameters" below) being routed (solid lines) across four FFN experts, where the router independently routes each token. The switch FFN layer returns the output of the selected FFN multiplied by the router gate value (dotted-line).

Why Single-Expert Routing Works

Prior work assumed multiple experts were required for learning

We demonstrate in the Switch Layer:

- Comparable or better model quality
- Reduced routing computation
- Lower communication overhead

Benefits of the Switch layer:

- Faster training
- Lower memory usage
- Easier implementation

Efficient Distributed Design

Experts are distributed across devices

Model parameters grow with hardware scale

Each device maintains:

- Fixed memory usage
- Fixed computation per token

Enables efficient training on:

- Large TPU clusters
- Small numbers of cores

Expert Capacity and Token Routing

Each expert processes a fixed number of tokens

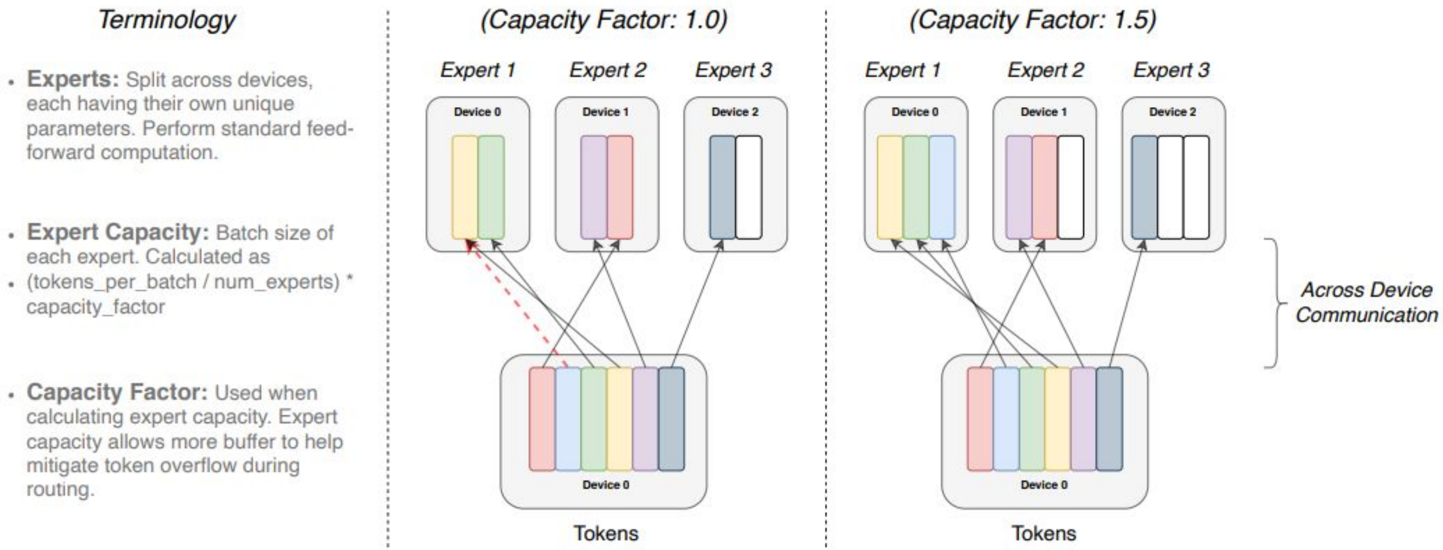
Capacity factor balances:

- Dropped tokens
- Wasted computation

Empirical findings:

- Dropped tokens remain under 1 percent
- Model quality scales reliably across expert counts

Load balancing enforced via auxiliary loss



Terminology

- **Experts:** Split across devices, each having their own unique parameters. Perform standard feed-forward computation.
- **Expert Capacity:** Batch size of each expert. Calculated as
- (tokens_per_batch / num_experts) * capacity_factor
- **Capacity Factor:** Used when calculating expert capacity. Expert capacity allows more buffer to help mitigate token overflow during routing.

Figure 3: Illustration of token routing dynamics. Each expert processes a fixed batch-size of tokens modulated by the *capacity factor*. Each token is routed to the expert with the highest router probability, but each expert has a fixed batch size of (total_tokens / num_experts) × capacity_factor. If the tokens are unevenly dispatched then certain experts will overflow (denoted by dotted red lines), resulting in these tokens not being processed by this layer. A larger capacity factor alleviates this overflow issue, but also increases computation and communication costs (depicted by padded white/empty slots).

Load Balancing as a Stability Mechanism

Auxiliary loss encourages uniform token distribution across experts

Prevents expert collapse and overload

Scales cleanly with number of experts

Does not interfere with main learning objective

Enables stable training across:

- Hundreds to thousands of experts

Empirical Performance Results

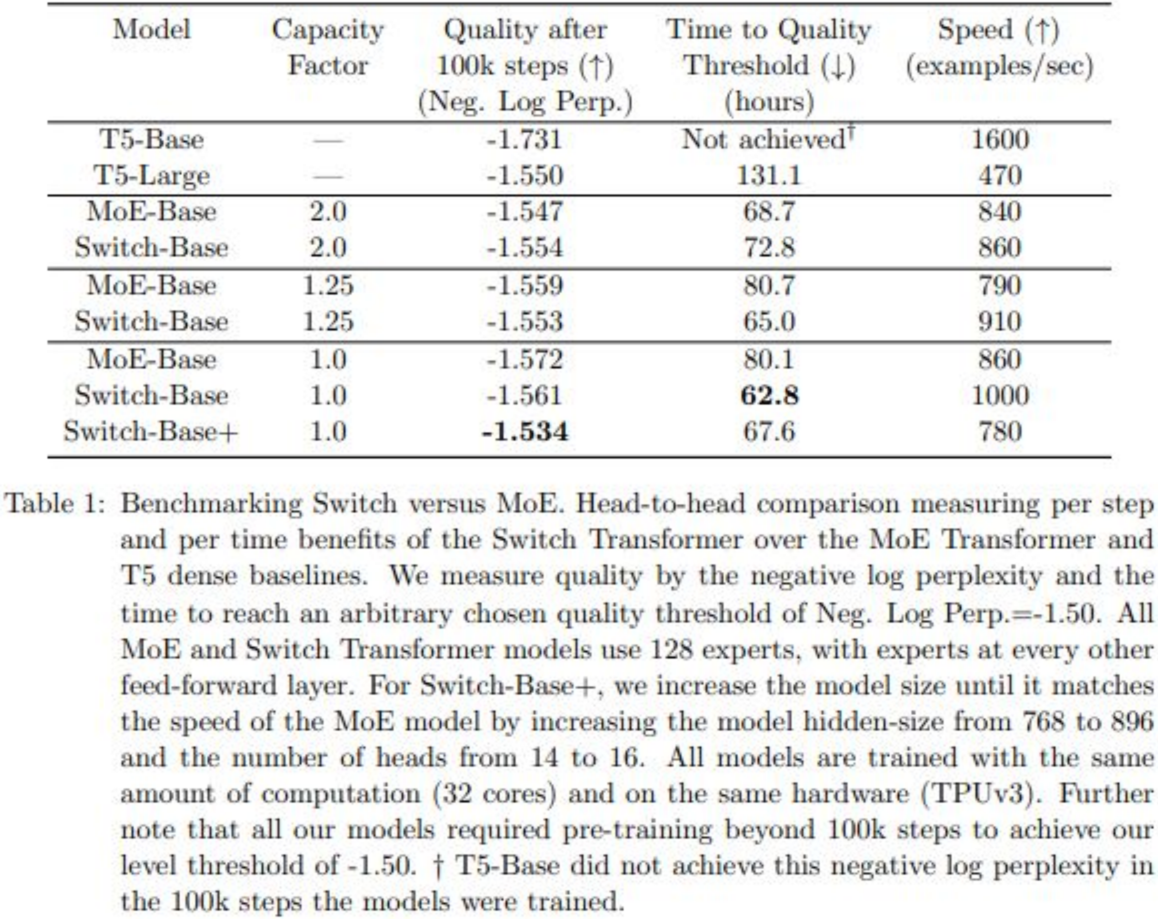
Switch Transformer:

- Outperforms dense Transformers at equal FLOPs
- Outperforms traditional MoE models in speed-quality tradeoff

Achieves:

- Over 7× pre-training speedup
- Lower computational footprint

Performs best at lower expert capacity factors



Model	Capacity Factor	Quality after 100k steps (↑) (Neg. Log Perp.)	Time to Quality Threshold (↓) (hours)	Speed (↑) (examples/sec)
T5-Base	—	-1.731	Not achieved†	1600
T5-Large	—	-1.550	131.1	470
MoE-Base	2.0	-1.547	68.7	840
Switch-Base	2.0	-1.554	72.8	860
MoE-Base	1.25	-1.559	80.7	790
Switch-Base	1.25	-1.553	65.0	910
MoE-Base	1.0	-1.572	80.1	860
Switch-Base	1.0	-1.561	**62.8**	1000
Switch-Base+	1.0	**-1.534**	67.6	780

Table 1: Benchmarking Switch versus MoE. Head-to-head comparison measuring per step and per time benefits of the Switch Transformer over the MoE Transformer and T5 dense baselines. We measure quality by the negative log perplexity and the time to reach an arbitrary chosen quality threshold of Neg. Log Perp.=-1.50. All MoE and Switch Transformer models use 128 experts, with experts at every other feed-forward layer. For Switch-Base+, we increase the model size until it matches the speed of the MoE model by increasing the model hidden-size from 768 to 896 and the number of heads from 14 to 16. All models are trained with the same amount of computation (32 cores) and on the same hardware (TPUv3). Further note that all our models required pre-training beyond 100k steps to achieve our level threshold of -1.50. † T5-Base did not achieve this negative log perplexity in the 100k steps the models were trained.

Training Stability Innovations

Sparse routing introduces numerical instability

Addressed through:

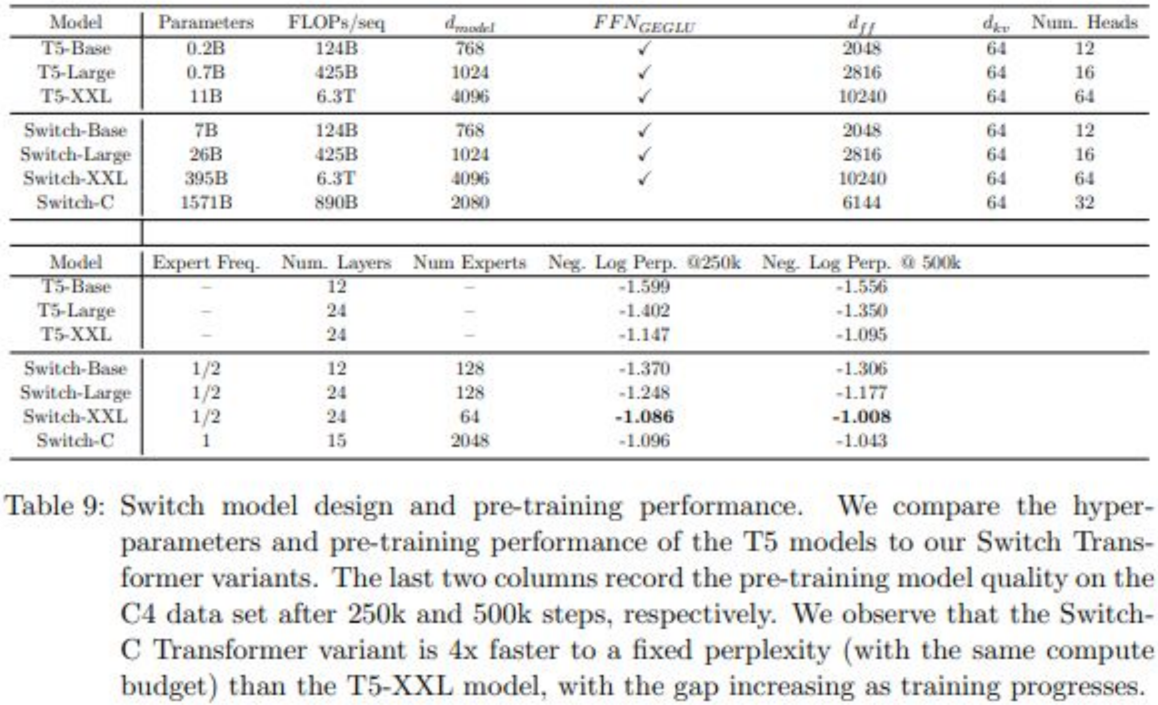
- Selective float32 precision only in routing
- Retaining float16 everywhere else

Achieves:

- Stability of float32
- Speed of mixed precision training

Model (precision)	Quality (Neg. Log Perp.) (↑)	Speed (Examples/sec) (↑)
Switch-Base (float32)	-1.718	1160
Switch-Base (bfloat16)	-3.780 [*diverged*]	**1390**
Switch-Base (Selective precision)	**-1.716**	1390

Table 2: Selective precision. We cast the local routing operations to float32 while preserving bfloat16 precision elsewhere to stabilize our model while achieving nearly equal speed to (unstable) bfloat16-precision training. We measure the quality of a 32 expert model after a fixed step count early in training its speed performance. For both Switch-Base in float32 and with Selective prevision we notice similar learning dynamics.



Model	Parameters	FLOPs/seq	d_{model}	FFN_{GEGLU}	d_{ff}	d_{kv}	Num. Heads
T5-Base	0.2B	124B	768	✓	2048	64	12
T5-Large	0.7B	425B	1024	✓	2816	64	16
T5-XXL	11B	6.3T	4096	✓	10240	64	64
Switch-Base	7B	124B	768	✓	2048	64	12
Switch-Large	26B	425B	1024	✓	2816	64	16
Switch-XXL	395B	6.3T	4096	✓	10240	64	64
Switch-C	1571B	890B	2080		6144	64	32

Model	Expert Freq.	Num. Layers	Num Experts	Neg. Log Perp. @250k	Neg. Log Perp. @ 500k
T5-Base	–	12	–	-1.599	-1.556
T5-Large	–	24	–	-1.402	-1.350
T5-XXL	–	24	–	-1.147	-1.095
Switch-Base	1/2	12	128	-1.370	-1.306
Switch-Large	1/2	24	128	-1.248	-1.177
Switch-XXL	1/2	24	64	**-1.086**	**-1.008**
Switch-C	1	15	2048	-1.096	-1.043

Table 9: Switch model design and pre-training performance. We compare the hyper-parameters and pre-training performance of the T5 models to our Switch Transformer variants. The last two columns record the pre-training model quality on the C4 data set after 250k and 500k steps, respectively. We observe that the Switch-C Transformer variant is 4x faster to a fixed perplexity (with the same compute budget) than the T5-XXL model, with the gap increasing as training progresses.

Improved Initialization Strategy

Standard Transformer initialization caused instability

Reduced initialization scale by 10×

Results:

- Higher model quality
- Dramatically reduced variance across runs

Enables training from:

- 200 million parameters
- To over one trillion parameters

Fine-Tuning and Regularization

Sparse models are prone to overfitting on small datasets

Introduced expert-specific dropout

Strategy:

- High dropout inside experts
- Lower dropout elsewhere

Improves performance across downstream tasks

Model (dropout)	GLUE	CNNDM	SQuAD	SuperGLUE
T5-Base (d=0.1)	82.9	**19.6**	83.5	72.4
Switch-Base (d=0.1)	84.7	19.1	**83.7**	**73.0**
Switch-Base (d=0.2)	84.4	19.2	**83.9**	**73.2**
Switch-Base (d=0.3)	83.9	19.6	83.4	70.7
Switch-Base (d=0.1, ed=0.4)	**85.2**	**19.6**	**83.7**	**73.0**

Table 4: Fine-tuning regularization results. A sweep of dropout rates while fine-tuning Switch Transformer models pre-trained on 34B tokens of the C4 data set (higher numbers are better). We observe that using a lower standard dropout rate at all non-expert layer, with a much larger dropout rate on the expert feed-forward layers, to perform the best.

Scaling Observations of Switch Transformers

Increasing the number of experts improves model efficiency without increasing FLOPs per token.

Sparse models learn faster and are more sample efficient than dense counterparts.

Switch-Base 64 expert model achieves T5-Base performance in 1/7th of the steps.

Larger models also improve sample efficiency they learn faster for the same number of tokens.

On a time basis, Switch Transformers yield substantial speed-ups, even compared to larger dense models.

Scaling expert count shows diminishing returns, but still outperforms FLOP-matched dense models.

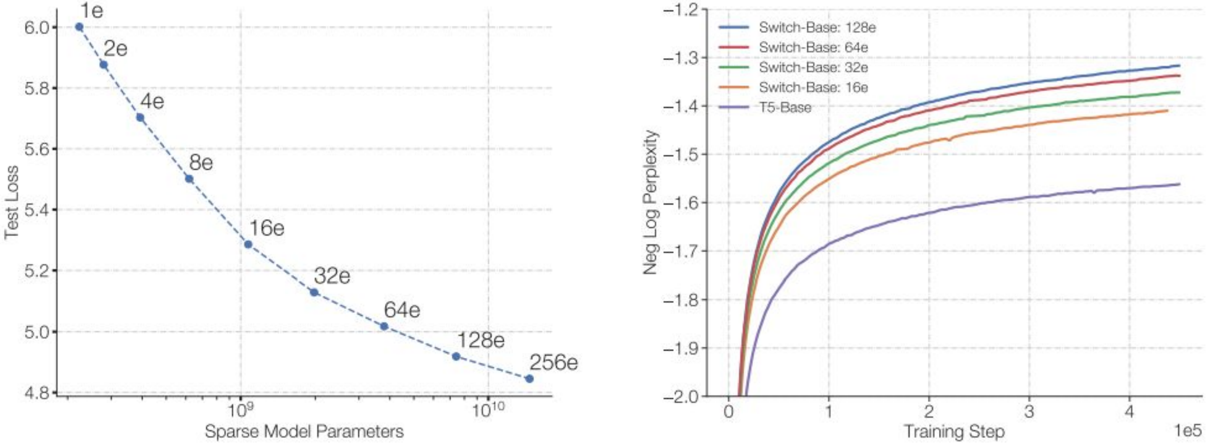


Figure 4: Scaling properties of the Switch Transformer. Left Plot: We measure the quality improvement, as measured by perplexity, as the parameters increase by scaling the number of experts. The top-left point corresponds to the T5-Base model with 223M parameters. Moving from top-left to bottom-right, we double the number of experts from 2, 4, 8 and so on until the bottom-right point of a 256 expert model with 14.7B parameters. Despite all models using an equal computational budget, we observe consistent improvements scaling the number of experts. Right Plot: Negative log perplexity per step sweeping over the number of experts. The dense baseline is shown with the purple line and we note improved sample efficiency of our Switch-Base models.

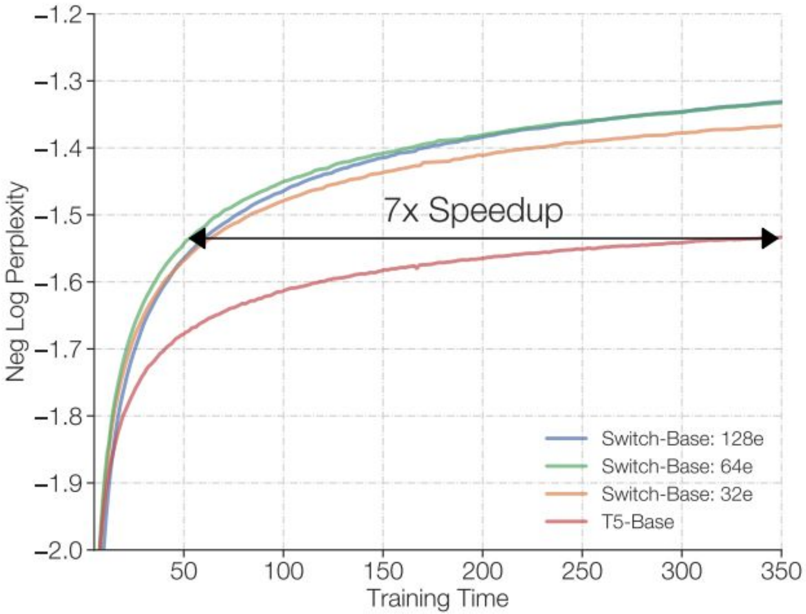


Figure 5: Speed advantage of Switch Transformer. All models trained on 32 TPUv3 cores with equal FLOPs per example. For a fixed amount of computation and training time, Switch Transformers significantly outperform the dense Transformer baseline. Our 64 expert Switch-Base model achieves the same quality in *one-seventh* the time of the T5-Base and continues to improve.

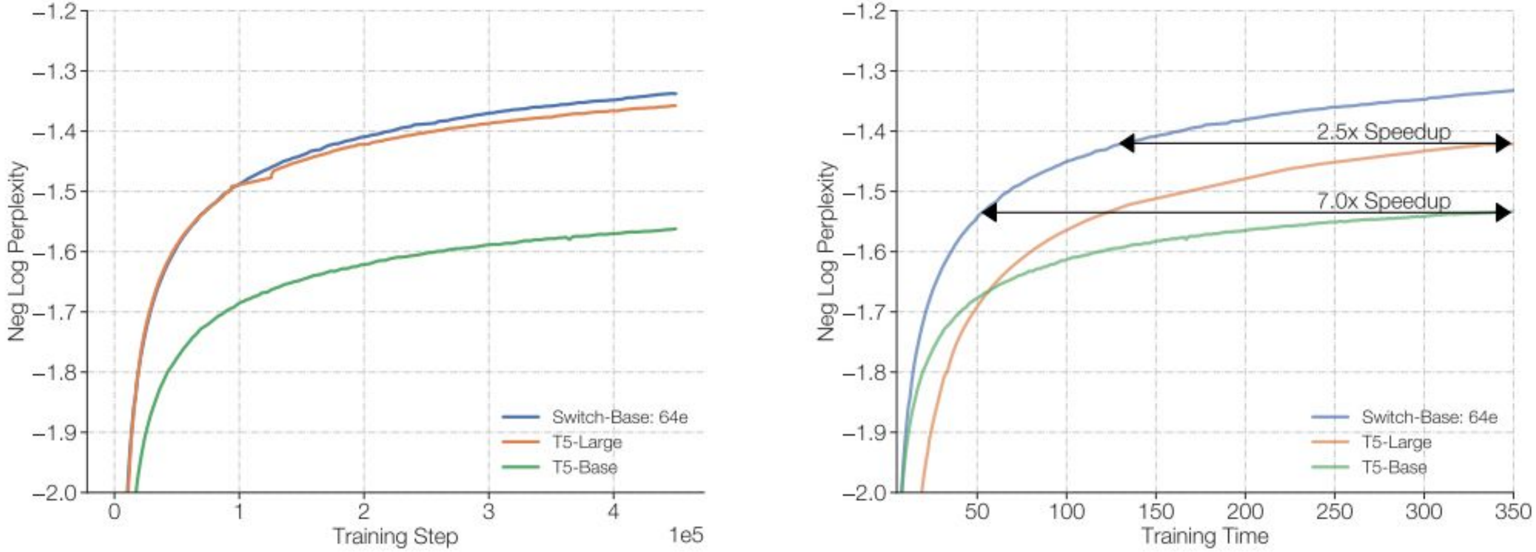


Figure 6: Scaling Transformer models with Switch layers or with standard dense model scaling. Left Plot: Switch-Base is more sample efficient than both the T5-Base, and T5-Large variant, which applies 3.5x more FLOPS per token. Right Plot: As before, on a wall-clock basis, we find that Switch-Base is still faster, and yields a 2.5x speedup over T5-Large.

Fine-Tuning Results

Switch Transformers fine-tuned on diverse NLP tasks:

- Question Answering (SQuAD, Natural Questions, TriviaQA, WebQuestions)
- Summarization (CNN/DailyMail, XSum)
- Natural Language Inference & Common Sense Reasoning (SuperGLUE, Winogrande)

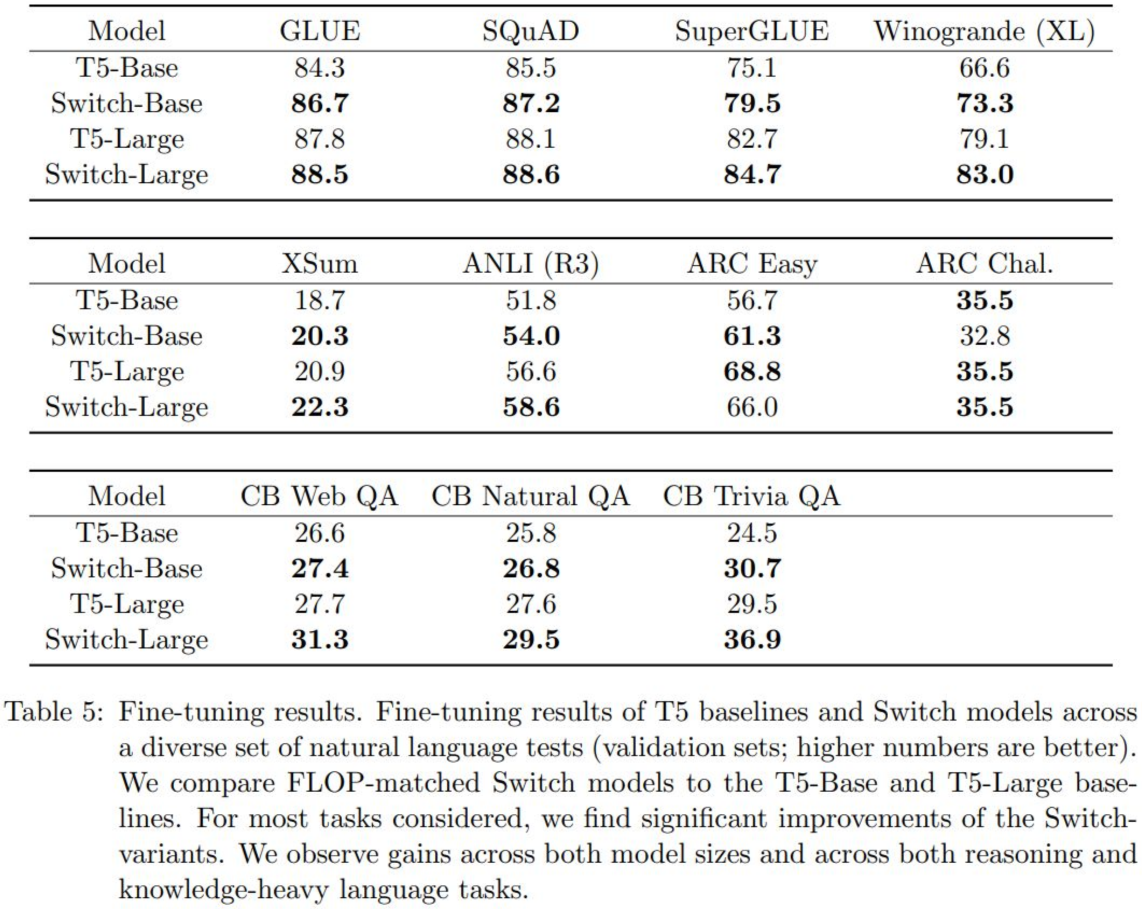
Performance improvements over T5 baselines:

- SuperGLUE: FLOP-matched Switch-Base +4.4% vs T5-Base, +2% vs T5-Large
- Large gains in Winogrande, TriviaQA, and XSum

Fine-Tuning Results

Fine-tuning technique: Expert Dropout

- Larger dropout in expert layers (0.4), smaller dropout in non-expert layers (0.1)
- Helps prevent overfitting on small downstream datasets



Model	GLUE	SQuAD	SuperGLUE	Winogrande (XL)
T5-Base	84.3	85.5	75.1	66.6
Switch-Base	**86.7**	**87.2**	**79.5**	**73.3**
T5-Large	87.8	88.1	82.7	79.1
Switch-Large	**88.5**	**88.6**	**84.7**	**83.0**

Model	XSum	ANLI (R3)	ARC Easy	ARC Chal.
T5-Base	18.7	51.8	56.7	**35.5**
Switch-Base	**20.3**	**54.0**	**61.3**	32.8
T5-Large	20.9	56.6	**68.8**	**35.5**
Switch-Large	**22.3**	**58.6**	66.0	**35.5**

Model	CB Web QA	CB Natural QA	CB Trivia QA
T5-Base	26.6	25.8	24.5
Switch-Base	**27.4**	**26.8**	**30.7**
T5-Large	27.7	27.6	29.5
Switch-Large	**31.3**	**29.5**	**36.9**

Table 5: Fine-tuning results. Fine-tuning results of T5 baselines and Switch models across a diverse set of natural language tests (validation sets; higher numbers are better). We compare FLOP-matched Switch models to the T5-Base and T5-Large baselines. For most tasks considered, we find significant improvements of the Switch-variants. We observe gains across both model sizes and across both reasoning and knowledge-heavy language tasks.

Model Distillation Results

Large sparse models distilled into small dense models

Achieved:

- Up to 99 percent parameter reduction
- Retained 30 percent of quality gains

Enables practical deployment of sparse-model benefits

Technique	Parameters	Quality (↑)
T5-Base	223M	-1.636
Switch-Base	3,800M	-1.444
Distillation	223M	(3%) -1.631
+ Init. non-expert weights from teacher	223M	(20%) -1.598
+ 0.75 mix of hard and soft loss	223M	(29%) -1.580
Initialization Baseline (no distillation)		
Init. non-expert weights from teacher	223M	-1.639

Table 6: Distilling Switch Transformers for Language Modeling. Initializing T5-Base with the non-expert weights from Switch-Base and using a loss from a mixture of teacher and ground-truth labels obtains the best performance. We can distill 30% of the performance improvement of a large sparse model with 100x more parameters back into a small dense model. For a final baseline, we find no improvement of T5-Base initialized with the expert weights, but trained normally without distillation.

	Dense	Sparse				
Parameters	223M	1.1B	2.0B	3.8B	7.4B	14.7B
Pre-trained Neg. Log Perp. (↑)	-1.636	-1.505	-1.474	-1.444	-1.432	-1.427
Distilled Neg. Log Perp. (↑)	—	-1.587	-1.585	-1.579	-1.582	-1.578
Percent of Teacher Performance	—	37%	32%	30 %	27 %	28 %
Compression Percent	—	82 %	90 %	95 %	97 %	99 %

Table 7: Distillation compression rates. We measure the quality when distilling large sparse models into a dense baseline. Our baseline, T5-Base, has a -1.636 Neg. Log Perp. quality. In the right columns, we then distill increasingly large sparse models into this same architecture. Through a combination of weight-initialization and a mixture of hard and soft losses, we can shrink our sparse teachers by 95%+ while preserving 30% of the quality gain. However, for significantly better and larger pre-trained teachers, we expect larger student models would be necessary to achieve these compression rates.

Model	Parameters	FLOPS	SuperGLUE (↑)
T5-Base	223M	124B	74.6
Switch-Base	7410M	124B	81.3
Distilled T5-Base	223M	124B	(30%) 76.6

Table 8: Distilling a fine-tuned SuperGLUE model. We distill a Switch-Base model fine-tuned on the SuperGLUE tasks into a T5-Base model. We observe that on smaller data sets our large sparse model can be an effective teacher for distillation. We find that we again achieve 30% of the teacher's performance on a 97% compressed model.

Multilingual Learning

Pre-training on 101 languages (mC4 dataset); 107 tasks due to script variants.

FLOP-matched mSwitch-Base vs. mT5-Base:

- Improved negative log perplexity across all 101 languages.
- Average speed-up of 5x, 91% of languages achieve ≥4x speed-up.

Switch Transformers demonstrate strong multi-task and multilingual capabilities

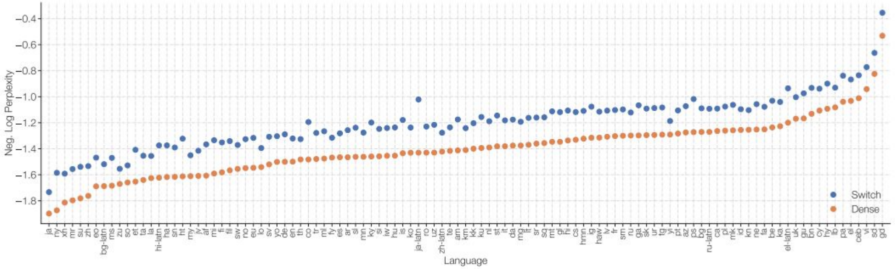


Figure 7: Multilingual pre-training on 101 languages. Improvements of Switch T5 Base model over dense baseline when multi-task training on 101 languages. We observe Switch Transformers to do quite well in the multi-task training setup and yield improvements on all 101 languages.

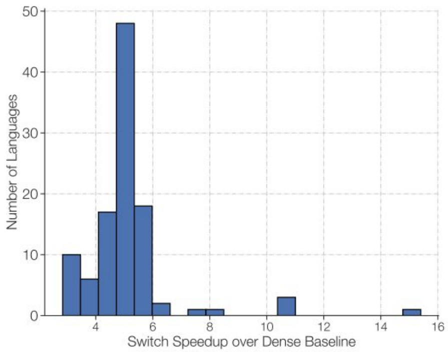


Figure 8: Multilingual pre-training on 101 languages. We histogram for each language, the step speedup of Switch Transformers over the FLOP matched T5 dense baseline to reach the same quality. Over all 101 languages, we achieve a mean step speed-up over mT5-Base of 5x and, for 91% of languages, we record a 4x, or greater, speedup to reach the final perplexity of mT5-Base.

Data Parallelism and Expert Routing

Sparse models split computation across cores using data, model, and expert parallelism.

Data parallelism: all cores handle different batches; minimal communication until gradient aggregation.

Expert parallelism: each core holds one expert; local router assigns tokens → reduces communication.

Combined expert + data parallelism:

- Tokens routed to correct expert (all-to-all communication).
- Expert layers handle variable batch sizes (capacity factor).

Efficient mapping is critical for large-scale sparse training.

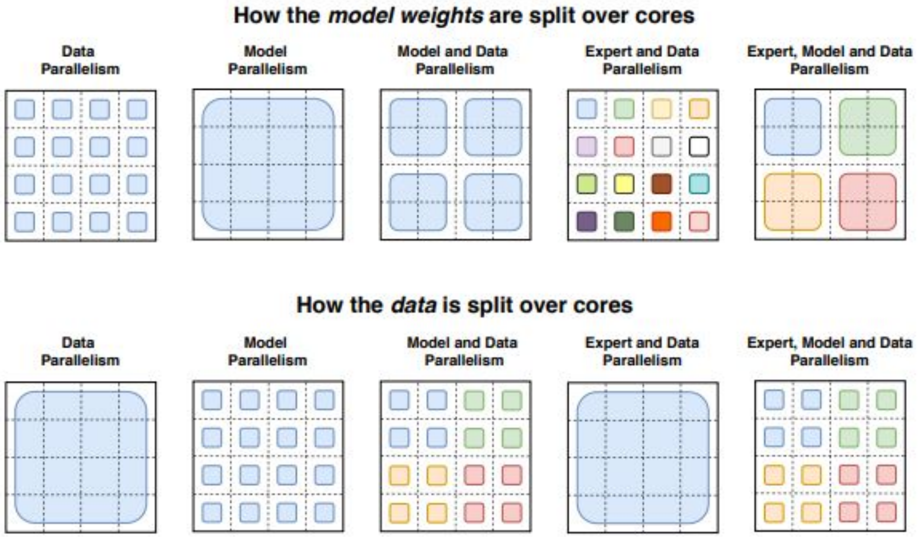

How the *model weights* are split over cores
Data Parallelism
Model Parallelism
Model and Data Parallelism
Expert and Data Parallelism
Expert, Model and Data Parallelism
How the *data* is split over cores
Data Parallelism
Model Parallelism
Model and Data Parallelism
Expert and Data Parallelism
Expert, Model and Data Parallelism

Figure 9: Data and weight partitioning strategies. Each 4×4 dotted-line grid represents 16 cores and the shaded squares are the data contained on that core (either model weights or batch of tokens). We illustrate both how the model weights and the data tensors are split for each strategy. **First Row:** illustration of how *model weights* are split across the cores. Shapes of different sizes in this row represent larger weight matrices in the Feed Forward Network (FFN) layers (e.g larger d_{ff} sizes). Each color of the shaded squares identifies a unique weight matrix. The number of parameters *per core* is fixed, but larger weight matrices will apply more computation to each token. **Second Row:** illustration of how the *data batch* is split across cores. Each core holds the same number of tokens which maintains a fixed memory usage across all strategies. The partitioning strategies have different properties of allowing each core to either have the same tokens or different tokens across cores, which is what the different colors symbolize.

Scaling to Trillion Parameters

Combined:

- Data parallelism
- Model parallelism
- Expert parallelism

Successfully trained trillion-parameter models

Achieved:

- 4× speedup over strong dense baselines

Demonstrates feasibility of extreme-scale language models

Model	Parameters	FLOPs/seq	d_{model}	FFN_{GEGLU}	d_{ff}	d_{kv}	Num. Heads
T5-Base	0.2B	124B	768	✓	2048	64	12
T5-Large	0.7B	425B	1024	✓	2816	64	16
T5-XXL	11B	6.3T	4096	✓	10240	64	64
Switch-Base	7B	124B	768	✓	2048	64	12
Switch-Large	26B	425B	1024	✓	2816	64	16
Switch-XXL	395B	6.3T	4096	✓	10240	64	64
Switch-C	1571B	890B	2080		6144	64	32

Model	Expert Freq.	Num. Layers	Num Experts	Neg. Log Perp. @250k	Neg. Log Perp. @ 500k
T5-Base	–	12	–	-1.599	-1.556
T5-Large	–	24	–	-1.402	-1.350
T5-XXL	–	24	–	-1.147	-1.095
Switch-Base	1/2	12	128	-1.370	-1.306
Switch-Large	1/2	24	128	-1.248	-1.177
Switch-XXL	1/2	24	64	**-1.086**	**-1.008**
Switch-C	1	15	2048	-1.096	-1.043

Table 9: Switch model design and pre-training performance. We compare the hyper-parameters and pre-training performance of the T5 models to our Switch Transformer variants. The last two columns record the pre-training model quality on the C4 data set after 250k and 500k steps, respectively. We observe that the Switch-C Transformer variant is 4x faster to a fixed perplexity (with the same compute budget) than the T5-XXL model, with the gap increasing as training progresses.

Scaling to Trillion Parameters

Switch Transformers successfully scale to trillion-parameter models using sparse expert architectures

A 1.6 trillion–parameter model (Switch-C) was trained without observed instability

Sparse scaling increases parameters without increasing FLOPs per token, enabling feasibility

However, not all trillion-scale configurations are equally stable

The Switch-XXL model, with ~10× higher FLOPs per sequence, showed sporadic training instability

Due to instability, Switch-XXL was not trained for the full 1M steps, unlike T5-XXL

Stability at extreme scale remains an open research challenge

Research Viability: Final Claim

Switch Transformer:

- Simplifies sparse modeling
- Improves stability and efficiency
- Scales to unprecedented sizes

Proves sparse activation is:

- Practical
- Efficient
- Superior at scale

Establishes a new paradigm for large language model training

Q&A

Mixtral of Experts

Abstract

Mixtral 8x7B is a Sparse Mixture of Experts (SMoE) language model

Architecture based on Mistral 7B with Mixture-of-Experts feedforward layers

Each layer contains 8 experts; only 2 experts are activated per token

Each token accesses 47B parameters, but only 13B are active during inference

Trained with a 32k token context window

Matches or outperforms Llama 2 70B and GPT-3.5 across benchmarks

Strong advantages in mathematics, code generation, and multilingual tasks

Instruction-tuned variant outperforms GPT-3.5 Turbo, Claude 2.1, Gemini Pro

Released under Apache 2.0 license

Introduction

Open-weight sparse Mixture-of-Experts model

Designed for high performance with reduced inference cost

Faster inference at low batch sizes and higher throughput at large batches

Decoder-only transformer architecture

Router network dynamically selects experts per token and layer

Multilingual pretraining with large context window

Demonstrates strong long-context retrieval ability

Instruction-tuned version trained with supervised fine-tuning and preference optimization

Fully supported by open-source inference infrastructure

Architectural Details

Model dimension: 4096

Number of layers: 32

Attention heads: 32 (8 key-value heads)

Hidden dimension: 14336

Context length: 32768 tokens

Vocabulary size: 32000

Number of experts per layer: 8

Experts activated per token: 2

Transformer backbone with Mixture-of-Experts feedforward layers

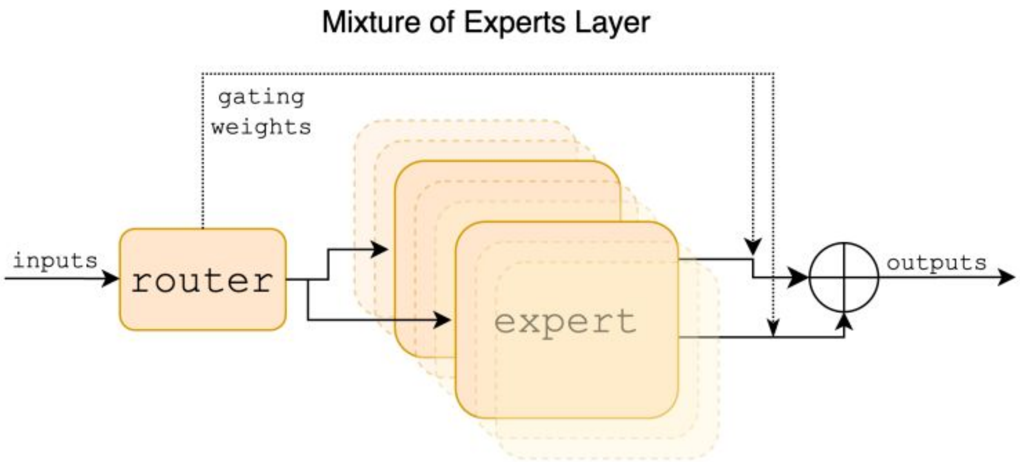


Figure 1: Mixture of Experts Layer. Each input vector is assigned to 2 of the 8 experts by a router. The layer's output is the weighted sum of the outputs of the two selected experts. In Mixtral, an expert is a standard feedforward block as in a vanilla transformer architecture.

Sparse Mixture of Experts

Each feedforward layer contains multiple expert networks

A gating (router) network selects the top experts per token

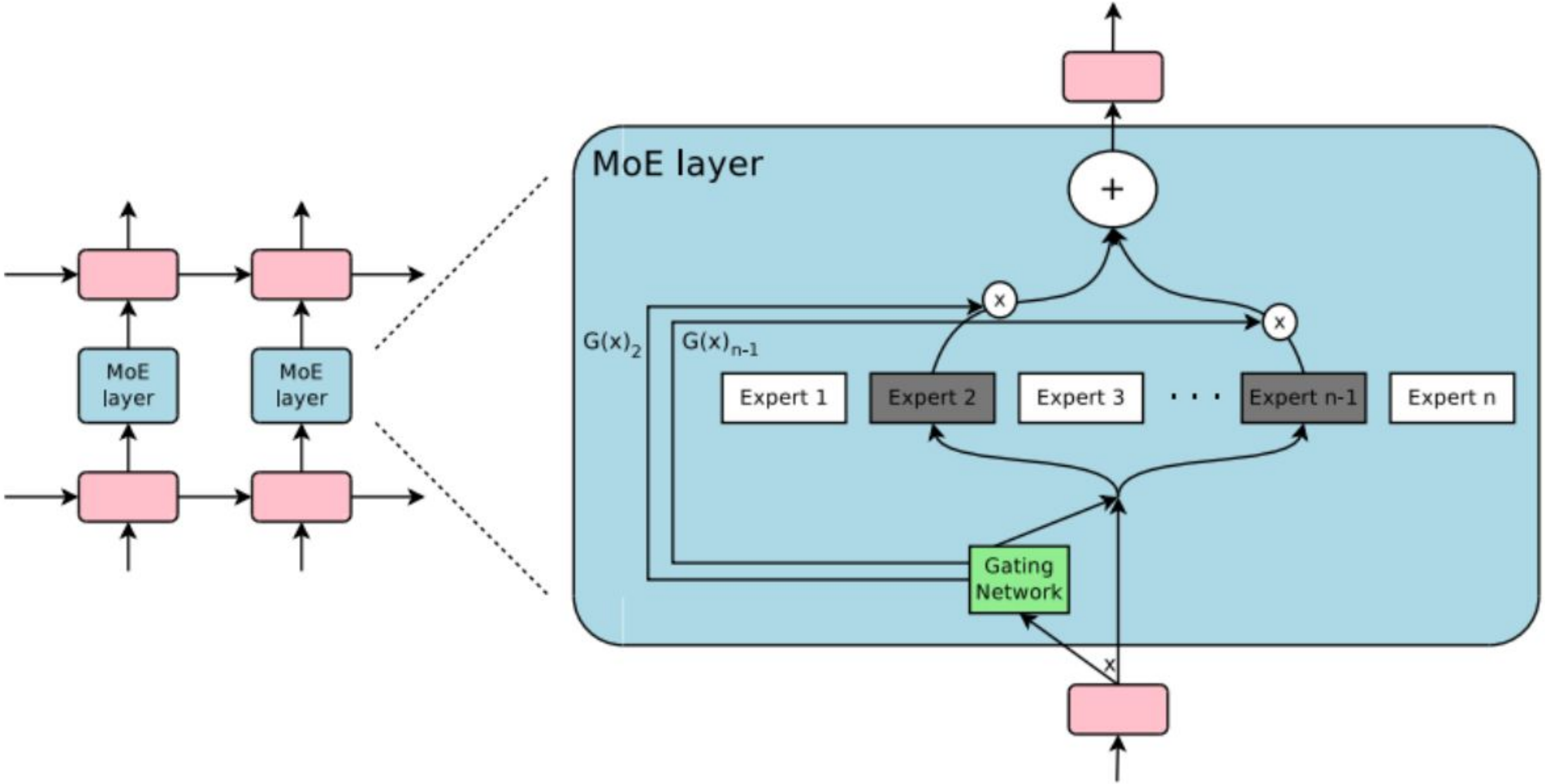
Outputs from selected experts are combined additively

Sparse activation avoids unnecessary computation

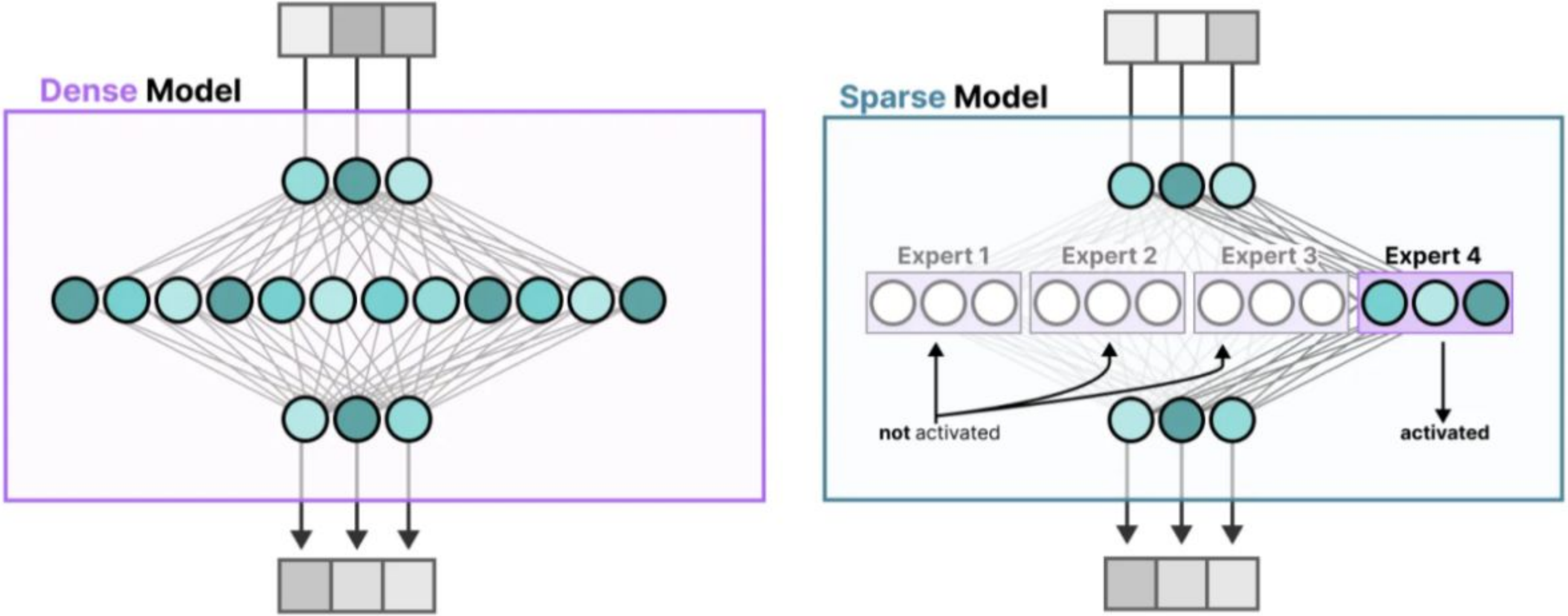
Gating implemented using Top-K selection with softmax

Number of active experts controls compute cost

Enables large model capacity with efficient inference



Core Architecture of Mixture of Expert Layer



Courtesy: https://newsletter.maartengrootendorst.com/p/a-visual-guide-to-mixture-of-experts

Results

Evaluated across diverse benchmark categories

Commonsense reasoning

World knowledge

Reading comprehension

Mathematics

Code generation

Aggregated benchmark suites

Mixtral surpasses Llama 2 70B on most metrics

Especially strong in math and code benchmarks

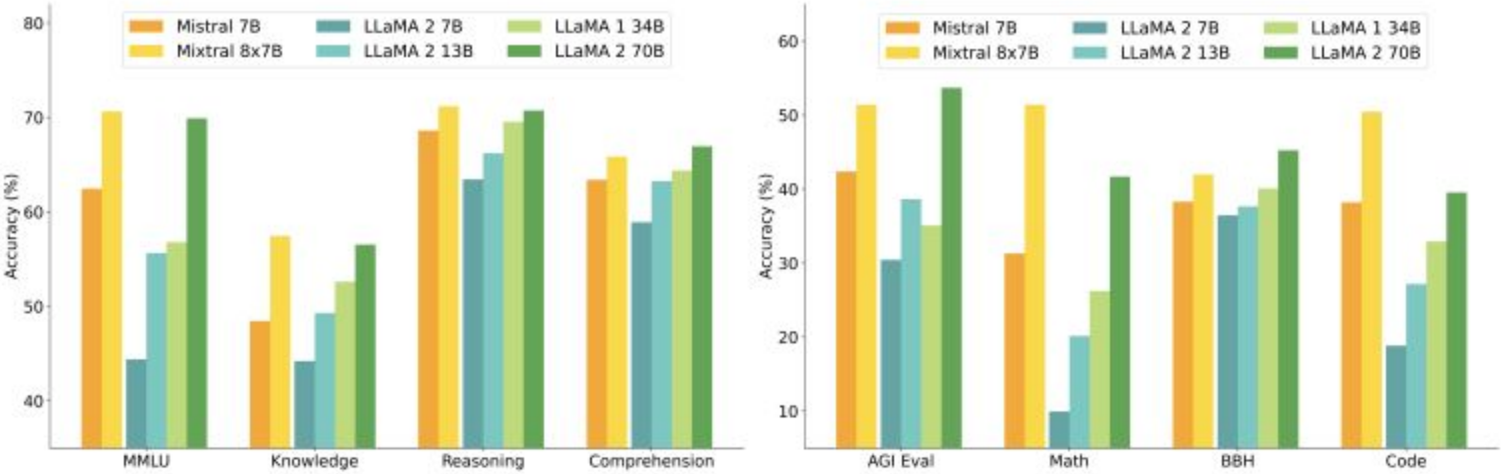


Figure 2: Performance of Mixtral and different Llama models on a wide range of benchmarks. All models were re-evaluated on all metrics with our evaluation pipeline for accurate comparison. Mixtral outperforms or matches Llama 2 70B on all benchmarks. In particular, it is vastly superior in mathematics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

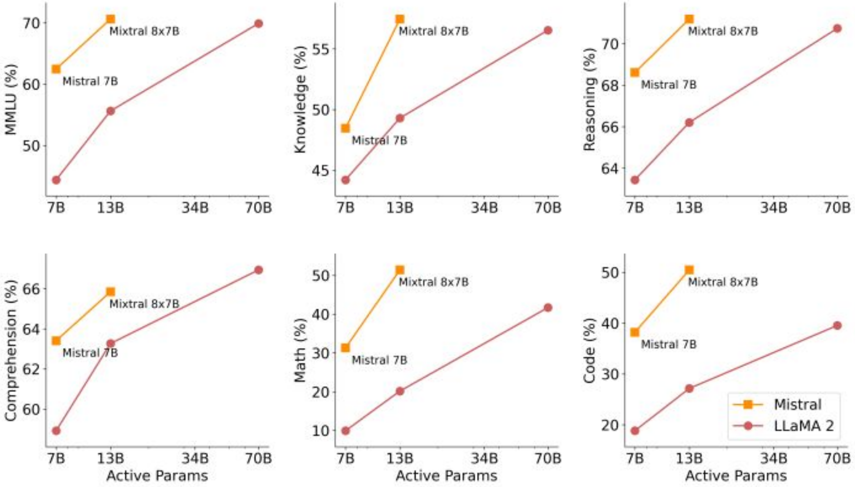
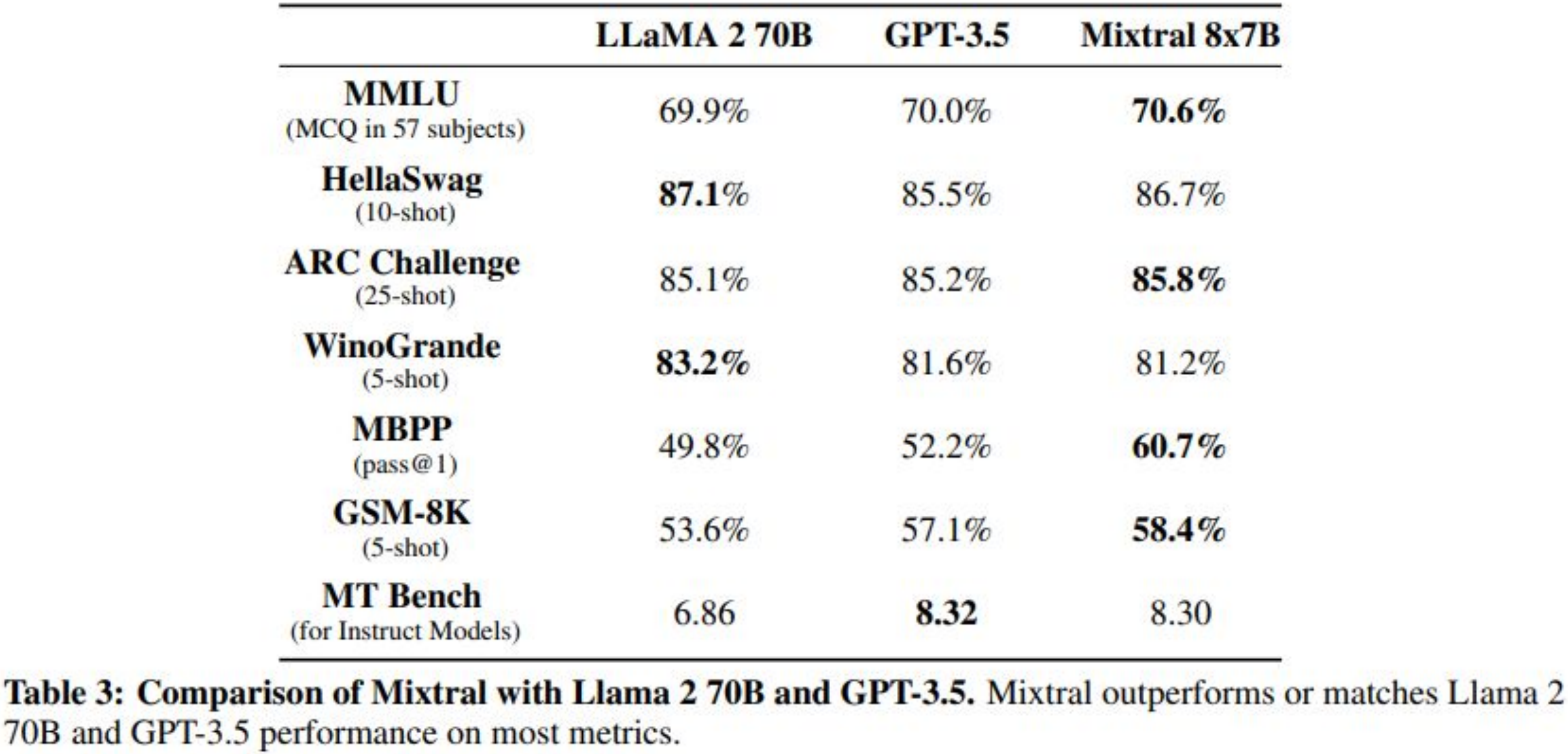
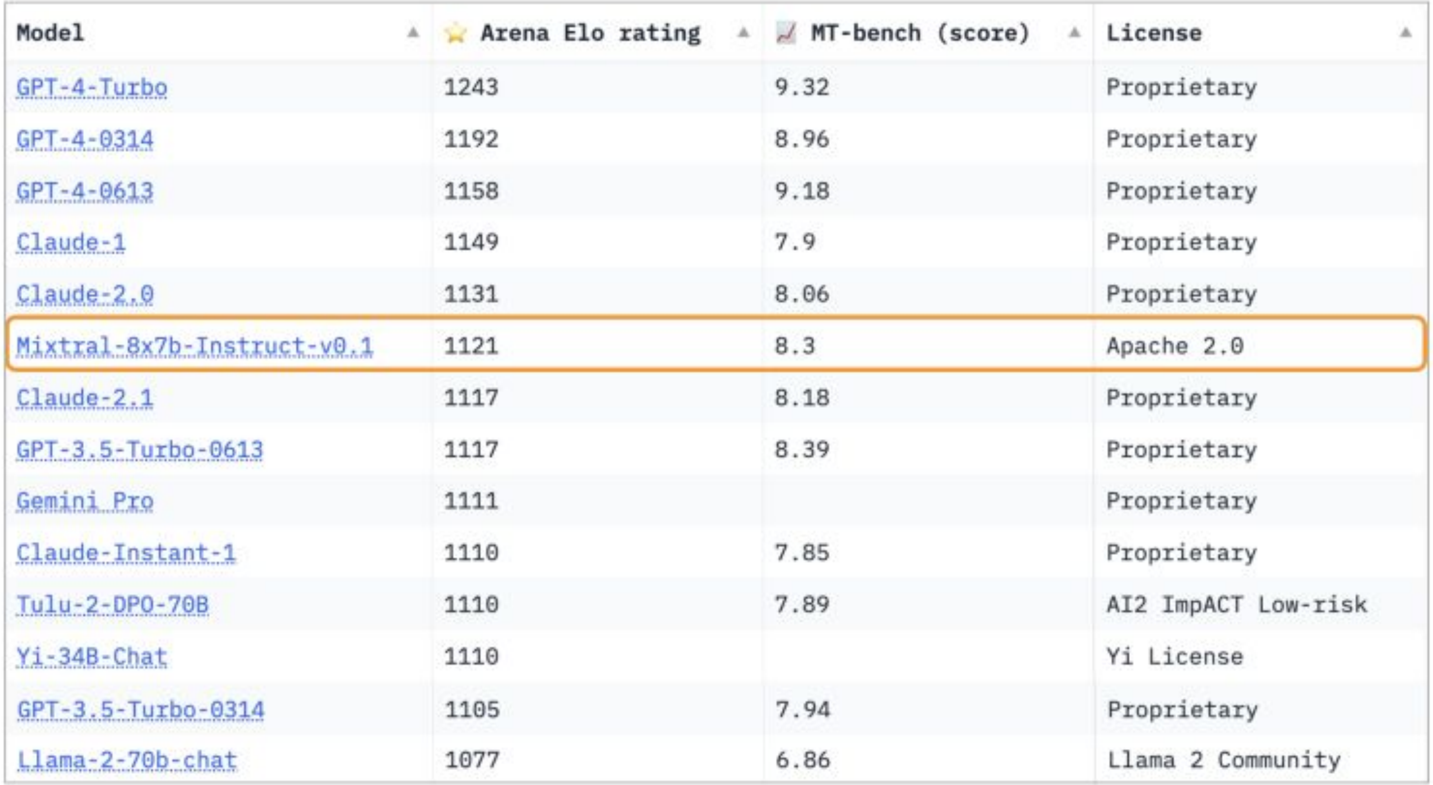


Figure 3: Results on MMLU, commonsense reasoning, world knowledge and reading comprehension, math and code for Mistral (7B/8x7B) vs Llama 2 (7B/13B/70B). Mixtral largely outperforms Llama 2 70B on all benchmarks, except on reading comprehension benchmarks while using 5x lower active parameters. It is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



	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.



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.

Size and Efficiency

Only 13B active parameters per token

Total parameter count: 47B (sparse)

Outperforms Llama 2 70B with significantly fewer active parameters

Active parameter count directly correlates with inference cost

Memory footprint remains smaller than Llama 2 70B

Best suited for batched workloads due to routing overhead

Multilingual Benchmarks

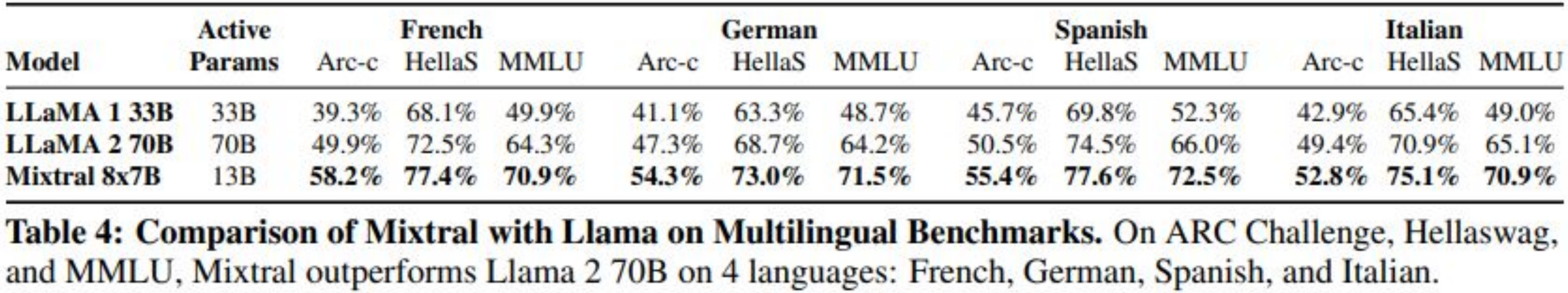
Increased multilingual data during pretraining

Maintains strong English performance

Significant gains in French, German, Spanish, and Italian

Outperforms Llama 2 70B across multilingual benchmarks

Demonstrates effective multilingual understanding



Model	Active	French			German			Spanish			Italian		
	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.

Long-Range Performance

Evaluated using synthetic passkey retrieval task

Achieves 100% retrieval accuracy at all context lengths

Retrieval performance independent of passkey position

Perplexity decreases as context length increases

Confirms effective utilization of long context windows

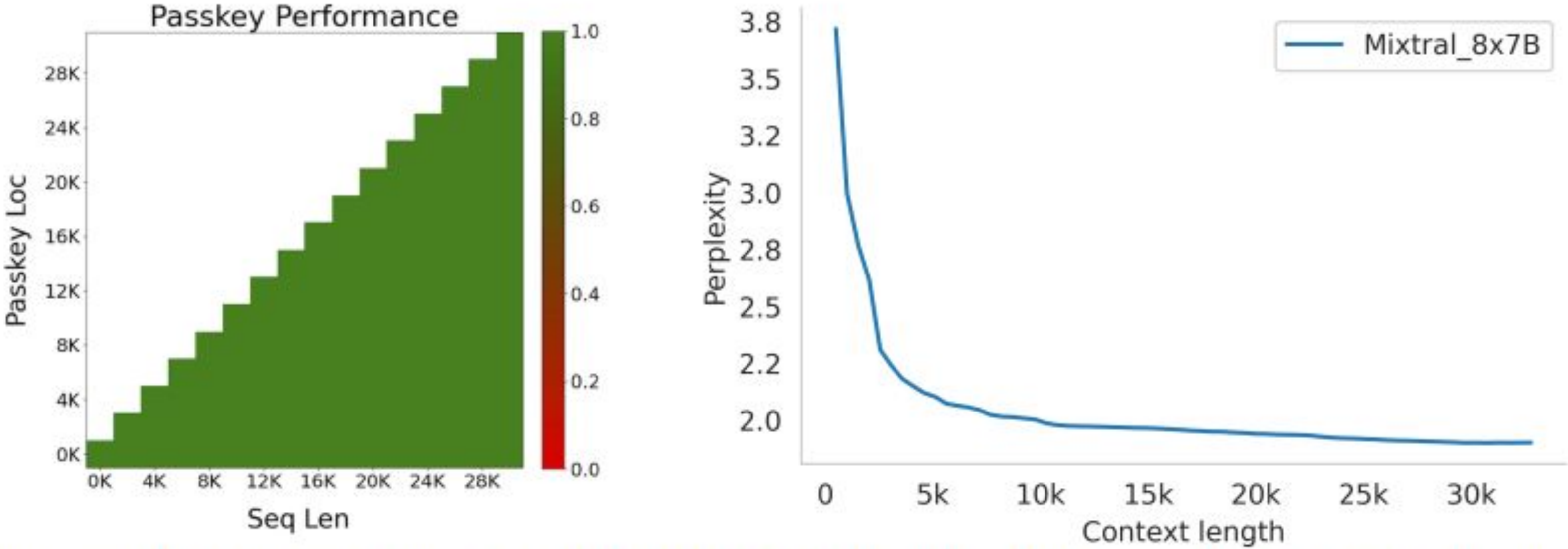


Figure 4: Long range performance of Mixtral. (Left) Mixtral has 100% retrieval accuracy of the Passkey task regardless of the location of the passkey and length of the input sequence. (Right) The perplexity of Mixtral on the proof-pile dataset decreases monotonically as the context length increases.

Bias Benchmarks

Evaluated on BBQ and BOLD datasets

Higher accuracy on bias-related QA tasks

More positive sentiment scores across demographic groups

Lower or comparable variance indicating reduced bias

Demonstrates improved fairness compared to Llama 2 70B

Instruction Fine-Tuning

Instruction-tuned using supervised fine-tuning

Further optimized using Direct Preference Optimization

Achieves high MT-Bench score (8.30)

Best-performing open-weight model as of December 2023

Outperforms leading proprietary chat models in human evaluations

Routing Analysis

Examines expert specialization across domains

No strong expert-topic specialization observed

Slight divergence in mathematics data

Strong temporal locality in expert selection

Consecutive tokens often routed to the same experts

Implications for caching and inference optimization

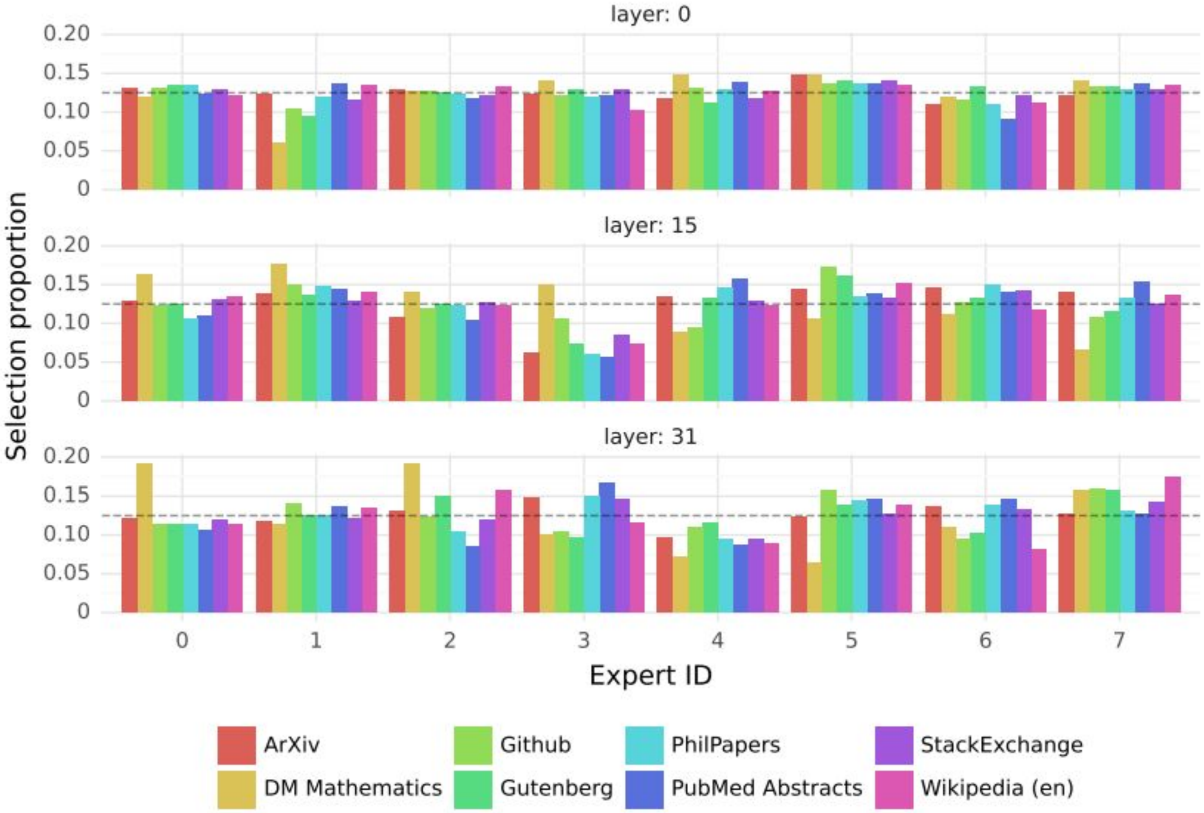


Figure 7: Proportion of tokens assigned to each expert on different domains from The Pile dataset for layers 0, 15, and 31. The gray dashed vertical line marks $1/8$, i.e. the proportion expected with uniform sampling. Here, we consider experts that are either selected as a first or second choice by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

Conclusion

Mixtral 8x7B is a state-of-the-art open-source Mixture-of-Experts model

Achieves strong performance with efficient compute usage

Instruction-tuned variant exceeds leading proprietary models

Only 13B active parameters per token

Released under Apache 2.0 license

Enables broad research and commercial adoption

Q&A