Mixture of Experts

CS 4804: Introduction to AI

Fall 2025

https://tuvllms.github.io/ai-fall-2025/

Tu Vu

VIRGINIA TECH.

Logistics

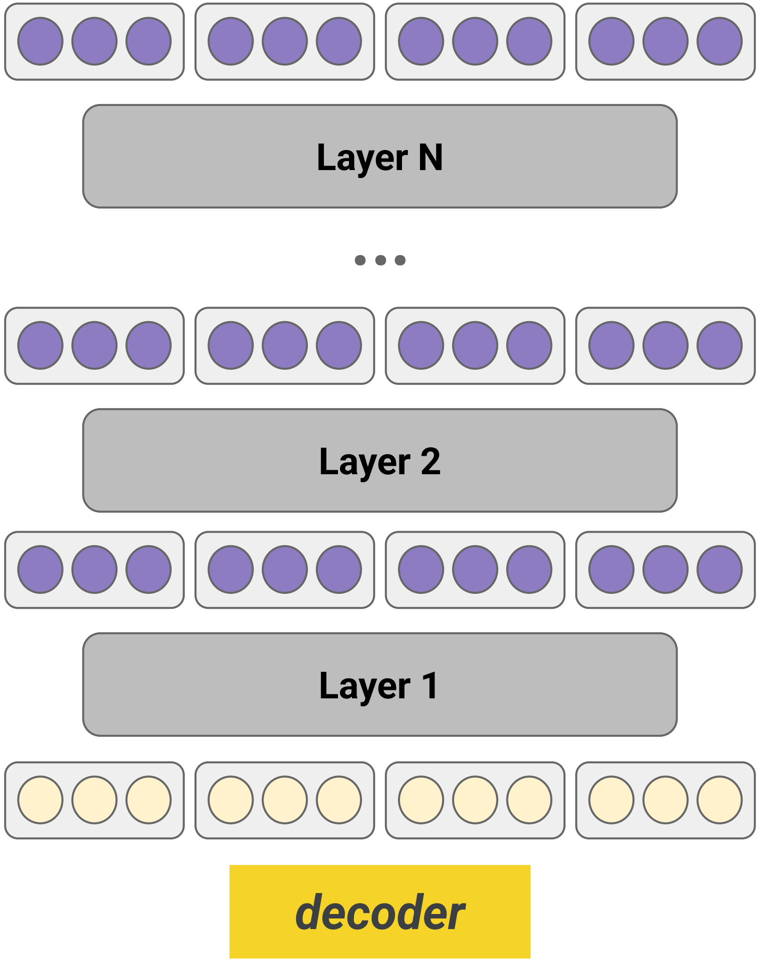
- No class next Tuesday 10/28 (Tu traveling), resume as usual next Thursday
- We are sending feedback for final project proposals
  - Please follow the template
- Quiz 2 released due 10/30
- HW 2 will be released later today due 11/13
- Teaching & learning evaluation: 11/4
- Final presentations: 12/4 & 12/9

AI Browsers

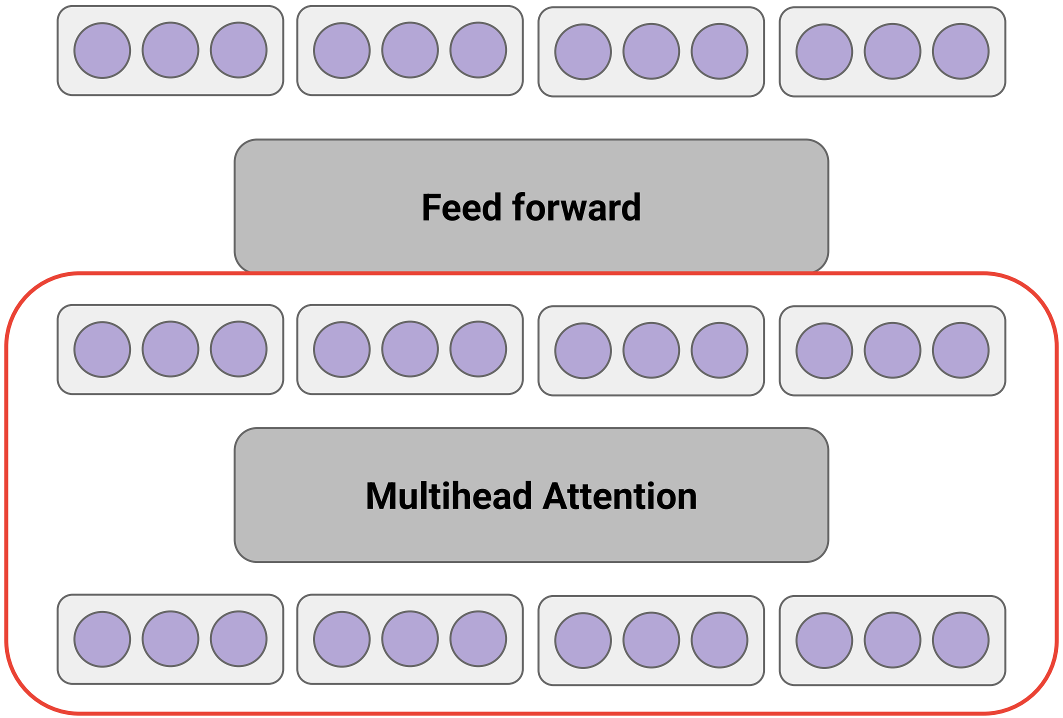
- OpenAI’s Atlas
  - https://openai.com/index/introducing-chatgpt-atlas/
- Perplexity’s Comet
  - https://www.perplexity.ai/comet

Decoder-only Transformer review

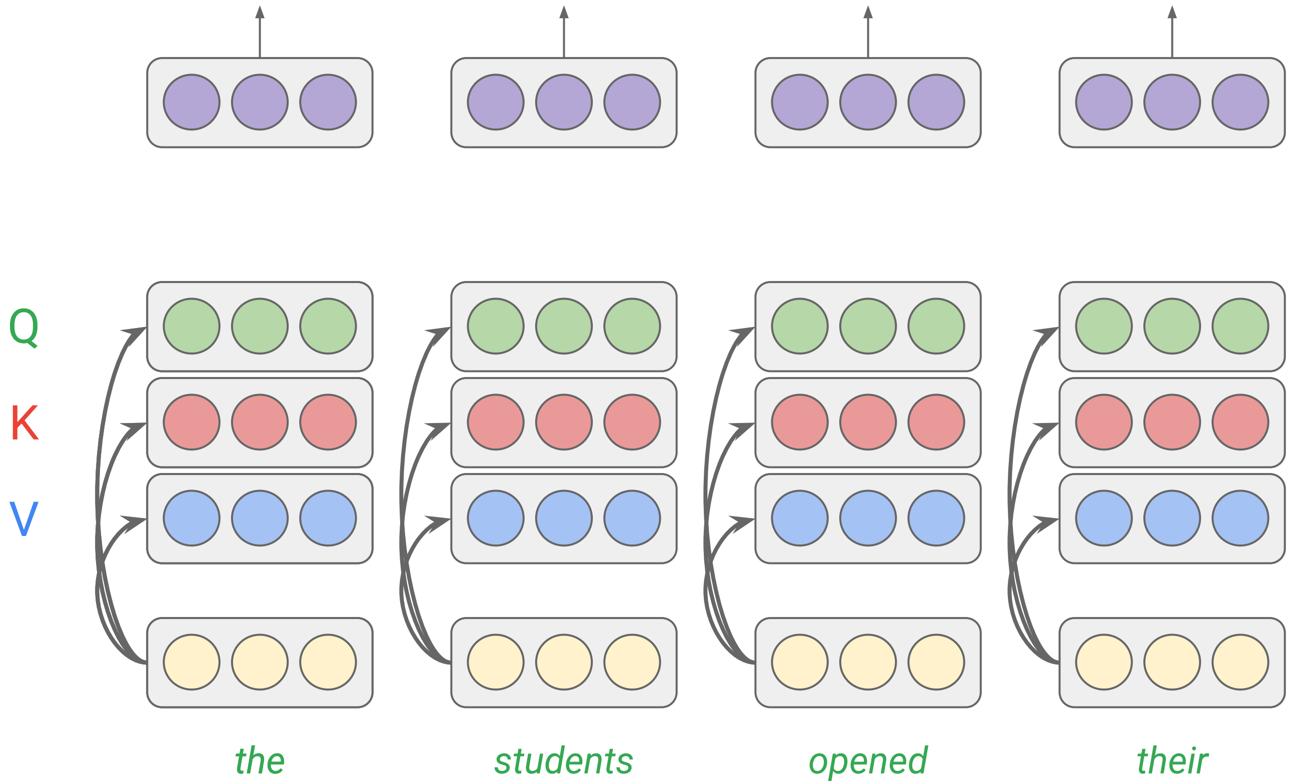
Transformer (N layers)



Transformer decoder



Attention



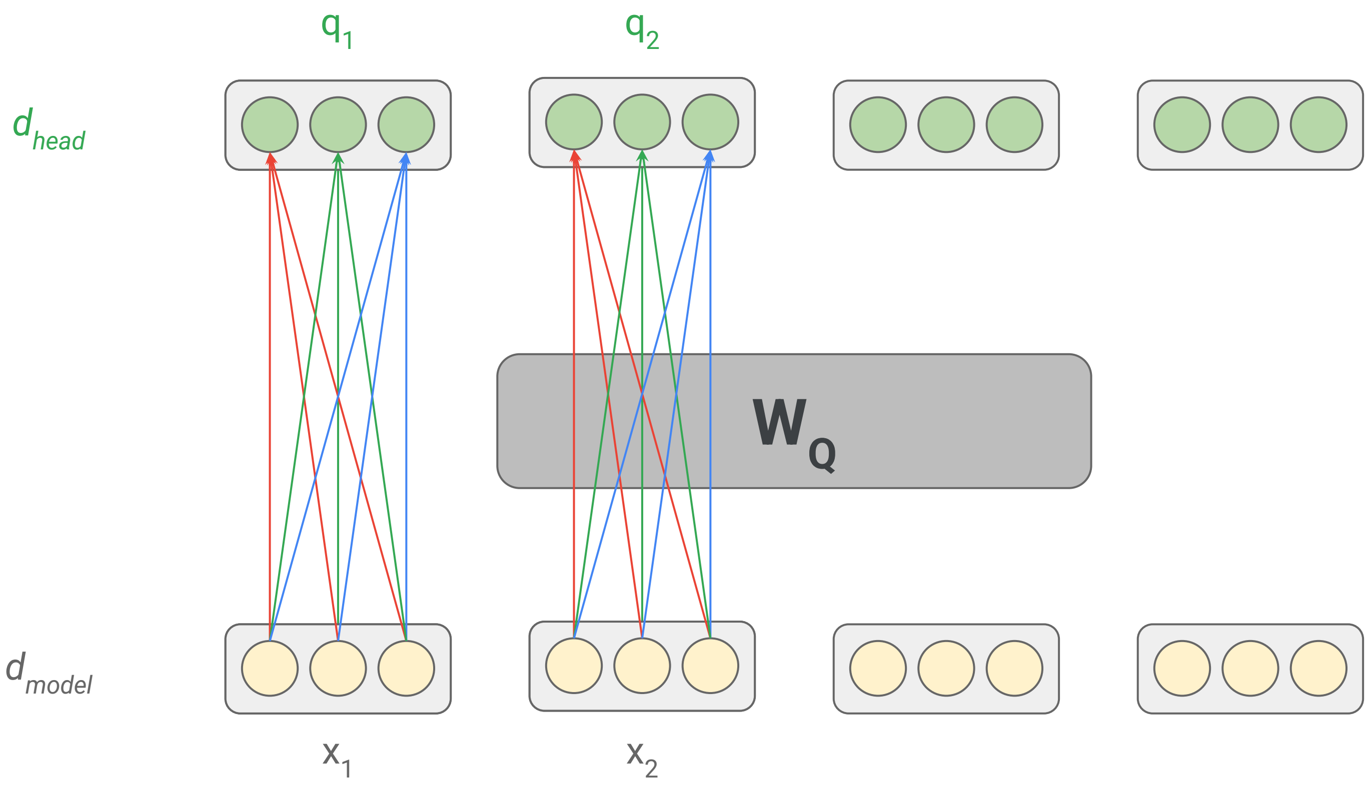
$$Q = X \cdot W_Q$$

$$K = X \cdot W_K$$

$$V = X \cdot W_V$$

linear projections

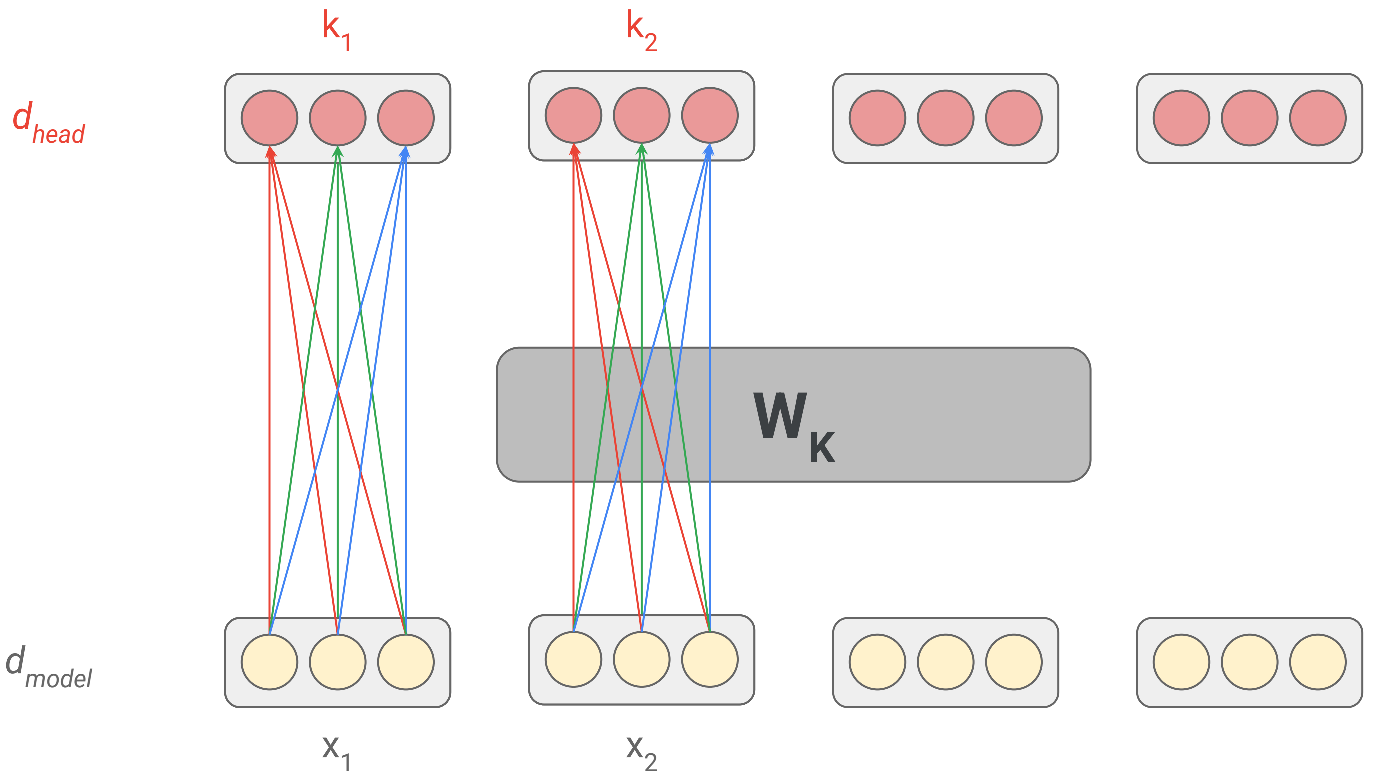
Query vectors



$$Q = X \cdot W_Q$$

linear projections

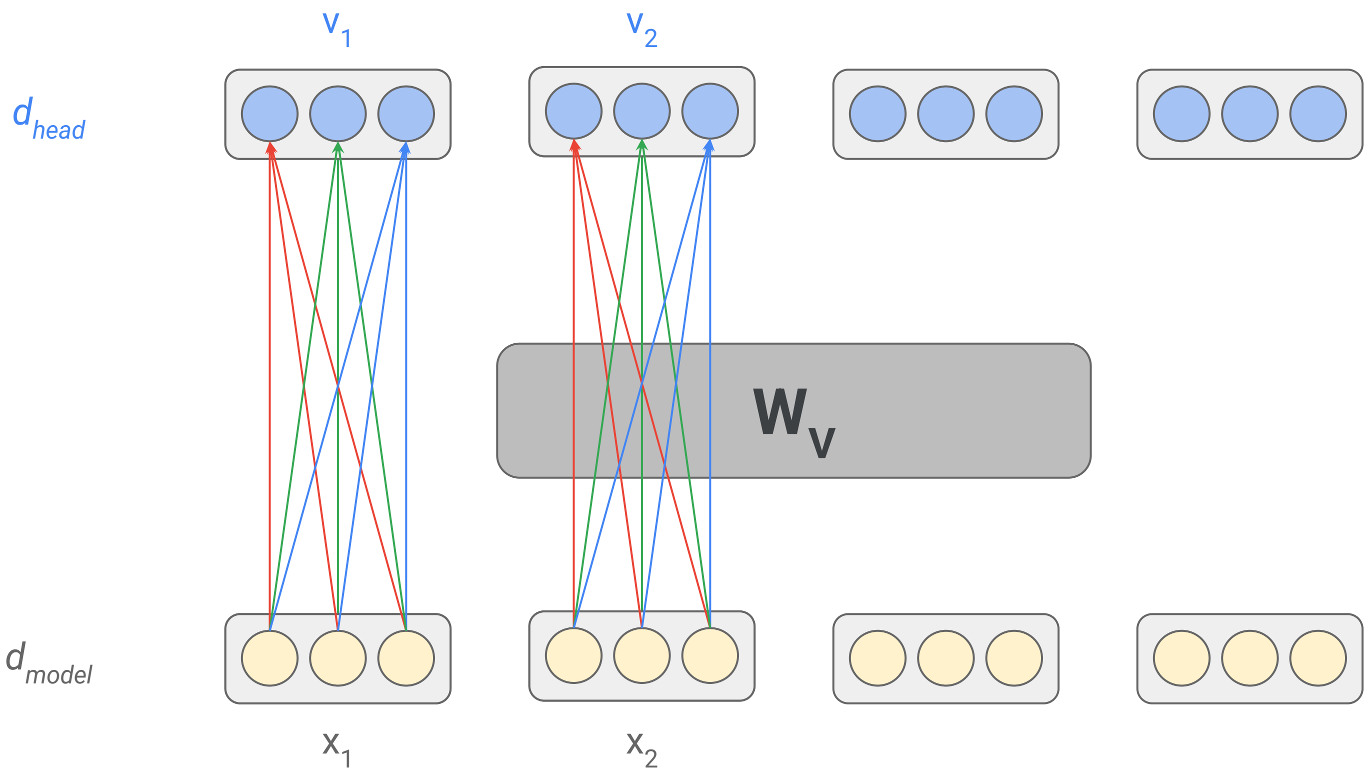
Key vectors



$$K = X \cdot W_K$$

linear projections

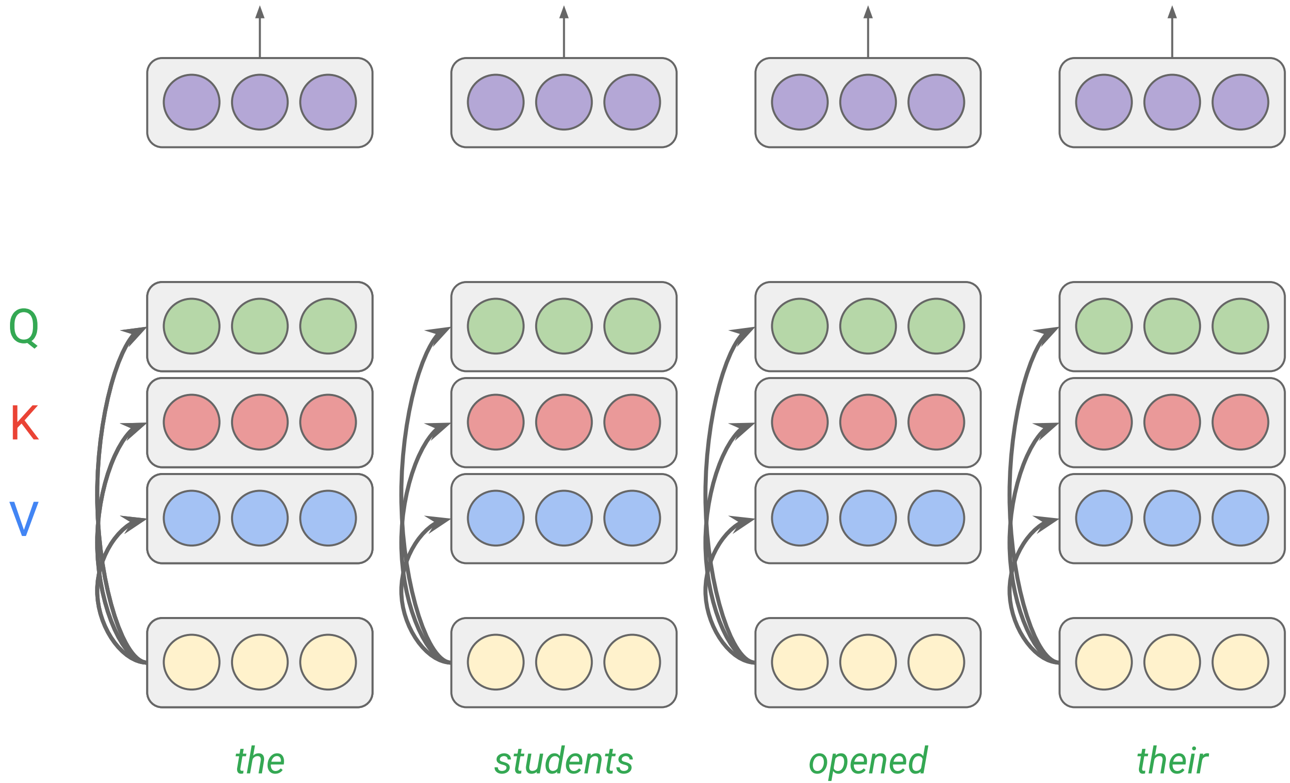
Value vectors



$$V = X \cdot W_V$$

linear projections

Attention (cont'd)



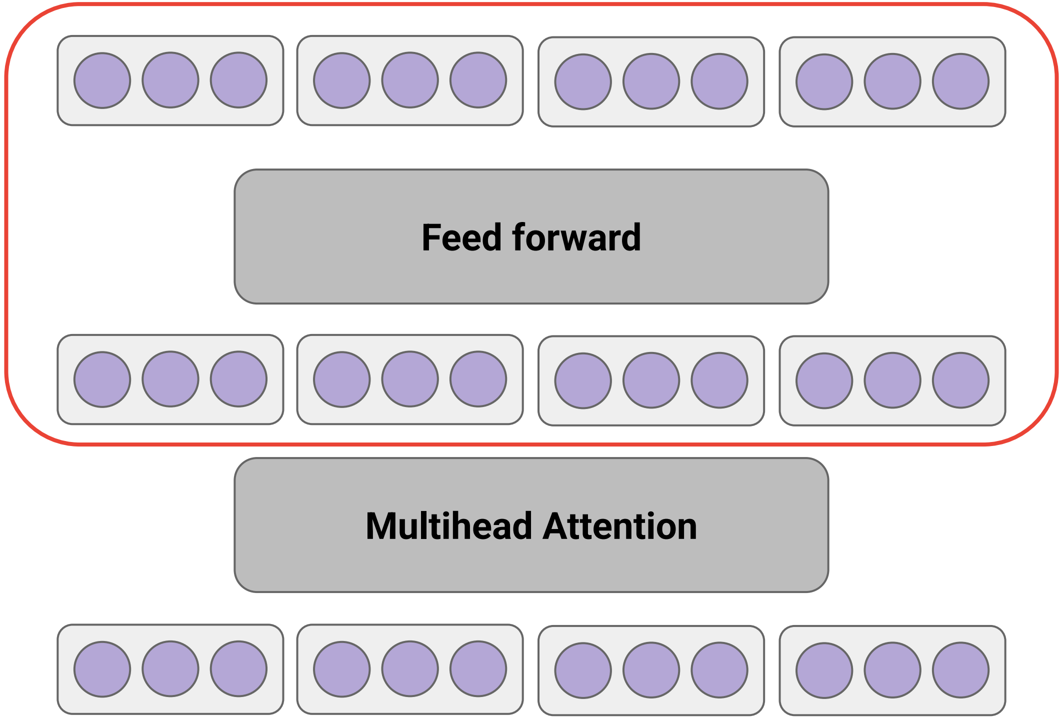
$$Q = X \cdot W_Q$$

$$K = X \cdot W_K$$

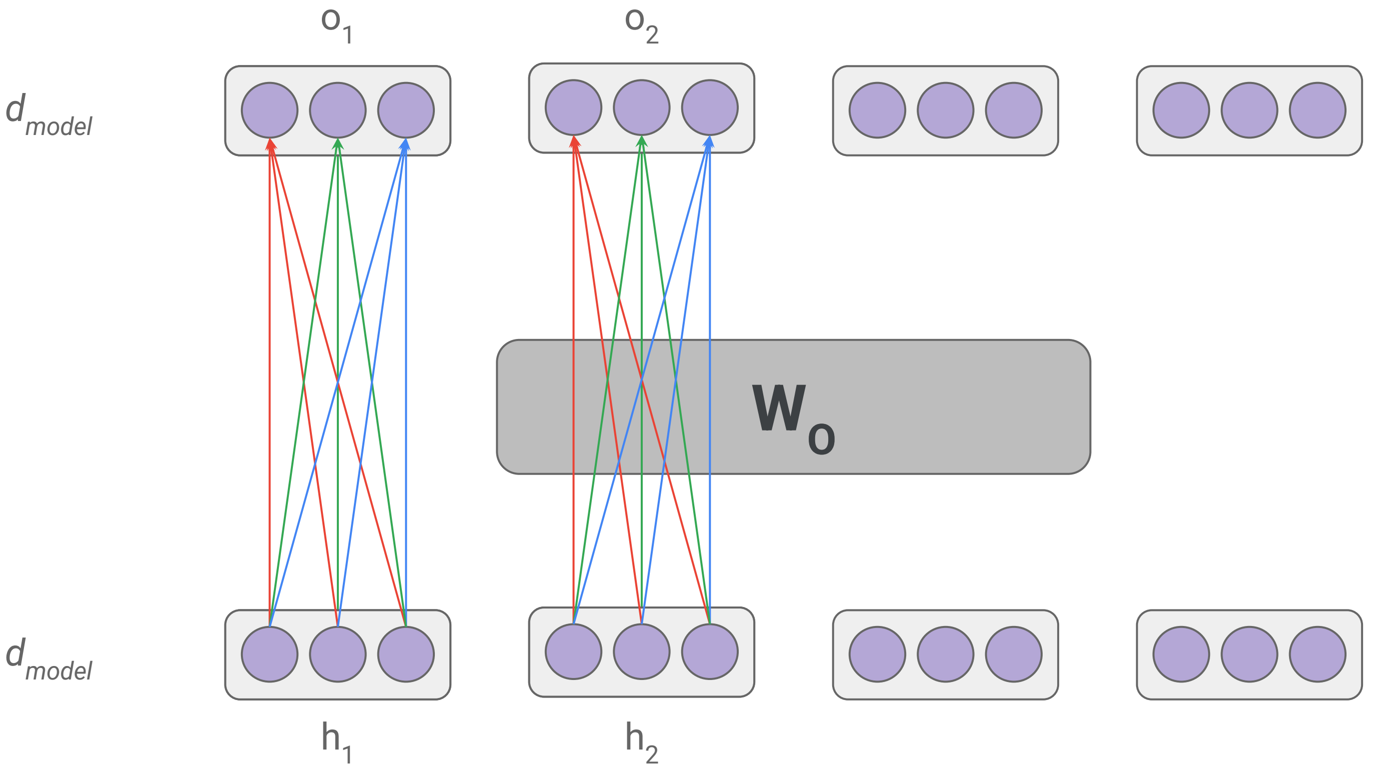
$$V = X \cdot W_V$$

linear projections

Transformer decoder



output vectors



$$O = H \cdot W_O$$

linear projections

What are mixture-of-experts?

- The gpt-oss models are autoregressive Mixture-of-Experts (MoE) transformers
  - **gpt-oss-120b:** 36 layers (116.8B total parameters and 5.1B "active" parameters per token)
  - **gpt-oss-20b:** 24 layers (20.9B total and 3.6B active parameters)
- Llama 4

Llama 4

Llama 4 release

meta-llama/Llama-4-Scout-17B-16E-Instruct
Image-Text-to-Text • 109B • Updated May 22 • 182k • • 1.12k

meta-llama/Llama-4-Scout-17B-16E
Image-Text-to-Text • 109B • Updated Apr 9 • 14.4k • 207

meta-llama/Llama-4-Maverick-17B-128E-Instruct
Image-Text-to-Text • 402B • Updated May 22 • 20.6k • • 417

meta-llama/Llama-4-Maverick-17B-128E-Instruct-FP8
Image-Text-to-Text • 402B • Updated May 22 • 193k • • 137

Llama 4:
Leading Multimodal Intelligence

Newest model suite offering unrivaled speed and efficiency

Llama 4 Behemoth

288B active parameter, **16** experts
2T total parameters

The most intelligent teacher model for distillation

Preview

Llama 4 Maverick

17B active parameters, **128** experts
400B total parameters

Native multimodal with **1M** context length

Available

Llama 4 Scout

17B active parameters, **16** experts
109B total parameters

Industry leading **10M** context length
Optimized inference

Available

The Bitter Lesson

“The biggest lesson that can be read from 70 years of AI research is that general methods that leverage computation are ultimately the most effective, and by a large margin.”

Rich Sutton, 2019

Simple architectures—backed by a generous computational budget, data set size and parameter count—surpass more complicated algorithms

Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer

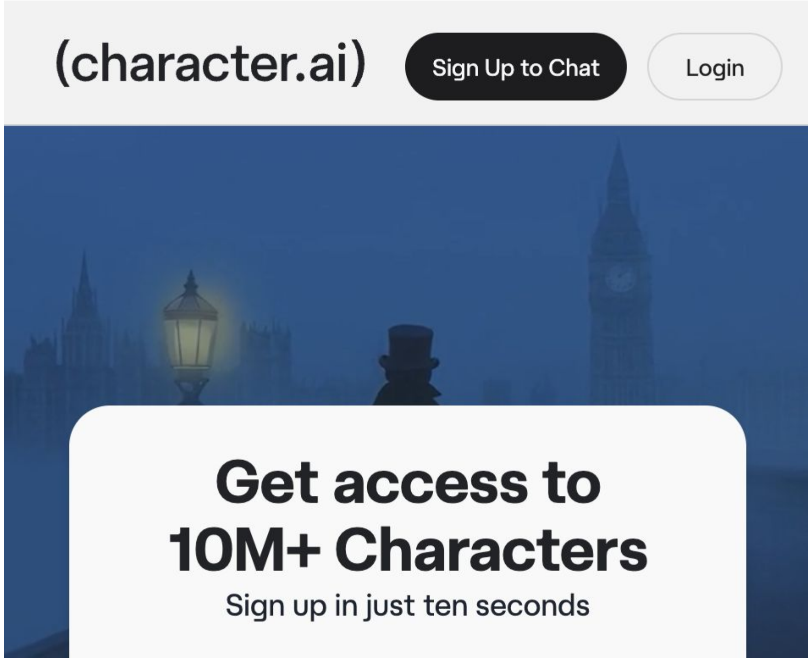
Noam Shazeer[1], Azalia Mirhoseini[*†1], Krzysztof Maziarz[*2], Andy Davis[1], Quoc Le[1], Geoffrey Hinton[1] and Jeff Dean[1]

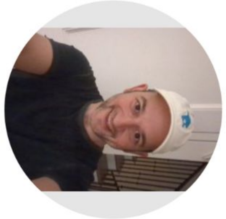
[1]Google Brain, {noam,azalia,andydavis,qvl,geoffhinton,jeff}@google.com
[2]Jagiellonian University, Cracow, krzysztof.maziarz@student.uj.edu.pl

23 Jan 2017

Abstract

The capacity of a neural network to absorb information is limited by its number of parameters. Conditional computation, where parts of the network are active on a per-example basis, has been proposed in theory as a way of dramatically increasing model capacity without a proportional increase in computation. In practice, however, there are significant algorithmic and performance challenges. In this work, we address these challenges and finally realize the promise of conditional





Noam Shazeer

Google

Verified email at google.com

Deep Learning

TITLE	CITED BY	YEAR
Attention is all you need A Vaswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, AN Gomez, ... Advances in neural information processing systems 30	209038	2017
Exploring the limits of transfer learning with a unified text-to-text transformer C Raffel, N Shazeer, A Roberts, K Lee, S Narang, M Matena, Y Zhou, W Li, ... Journal of machine learning research 21 (140), 1-67	27346	2020

Journal of Machine Learning Research 23 (2022) 1-40 Submitted 8/21; Revised 3/22; Published 4/22

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

William Fedus*
LIAMFEDUS@GOOGLE.COM

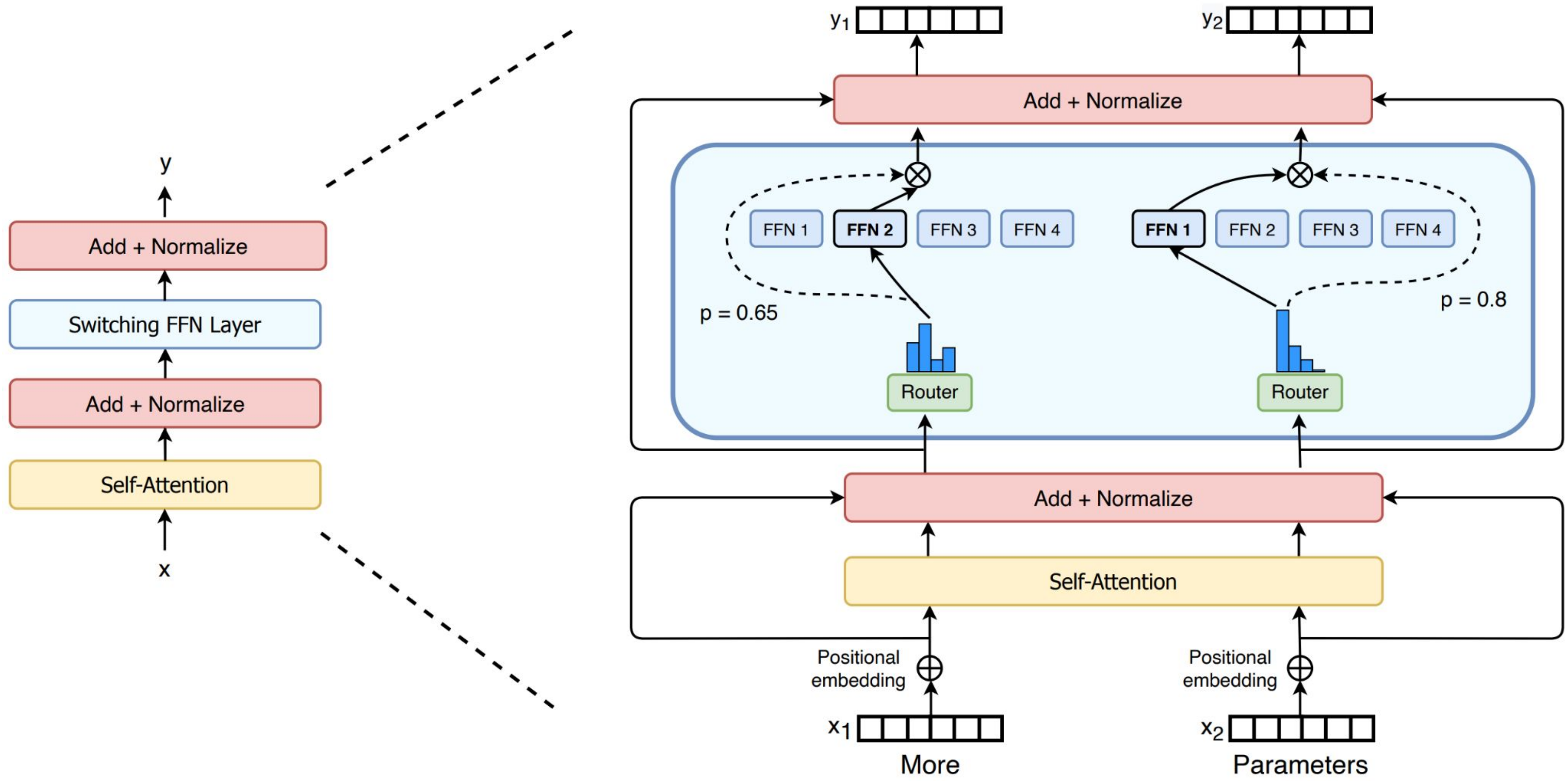
Barret Zoph*
BARRETZOPH@GOOGLE.COM

Noam Shazeer
NOAM@GOOGLE.COM

Google, Mountain View, CA 94043, USA

Switch Transformers

The layer operates independently on the tokens



Switch Transformers

- Vanilla Transformer
  - densely-activated

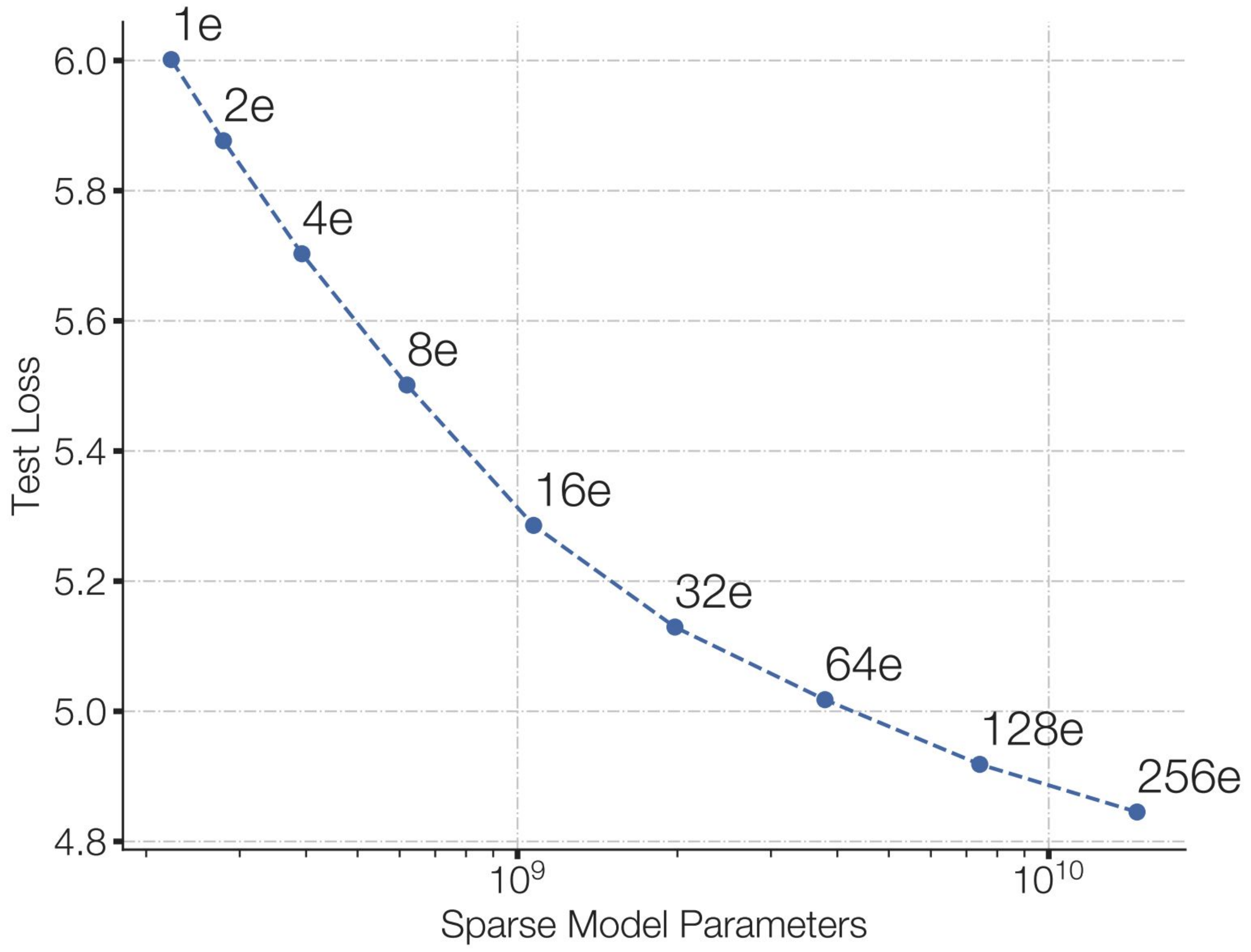
- Switch Transformer
  - sparsely-activated expert model
  - with an outrageous number of parameters—but a constant computational cost (!)
  - pretraining up to *trillion* parameter models and achieving a 4x speedup over the T5-XXL (11B)

Rethinking Mixture-of-Experts

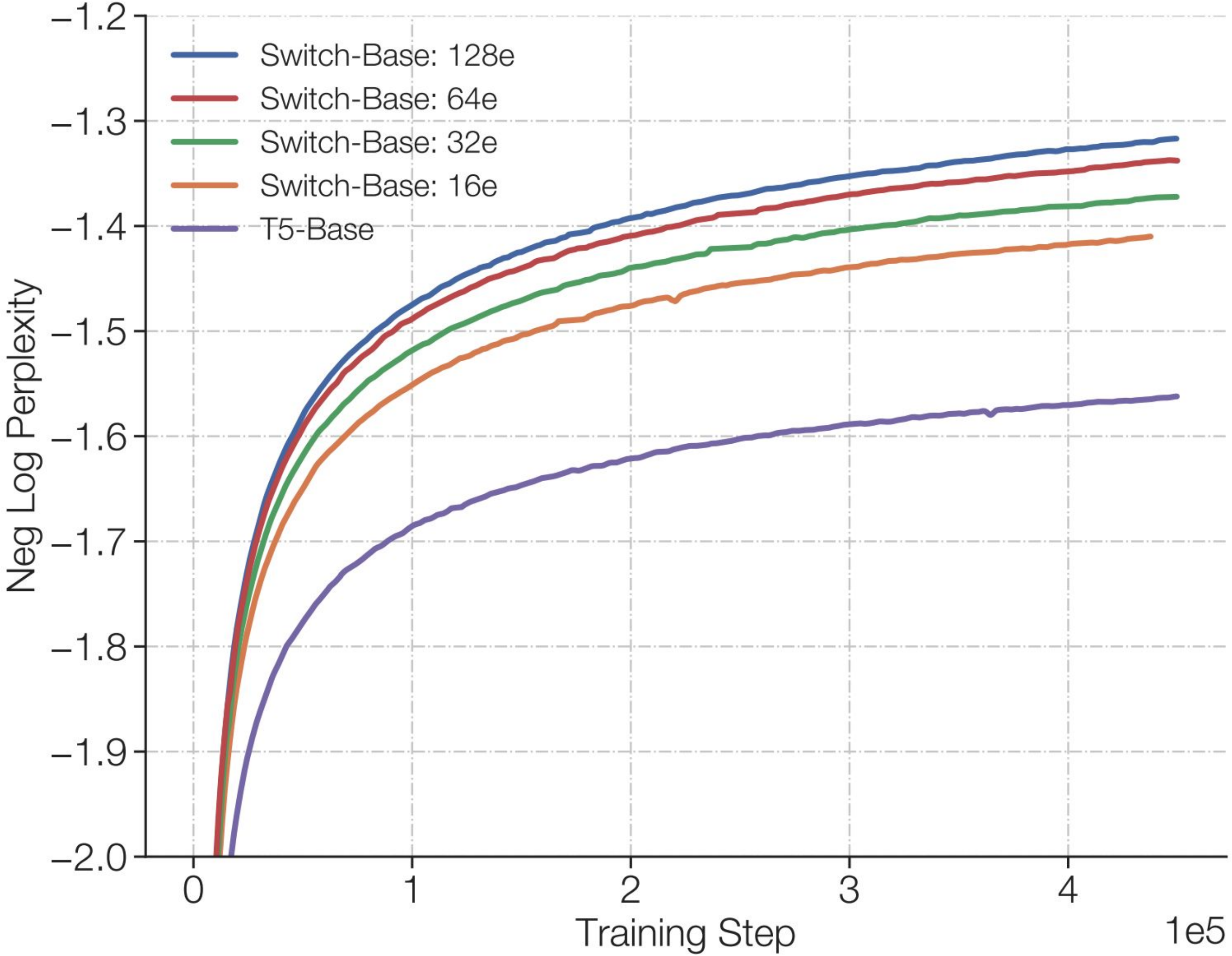
- Shazeer et al. (2017)
  - Each token is processed by every expert, and their outputs are combined
  - This increases computational cost linearly with the number of experts, so adding experts makes training and inference more expensive
- Switch layer
  - Only a small subset of experts are activated per token
  - This allows the model to scale to many more experts while keeping the computational cost per token roughly constant

Rethinking Mixture-of-Experts (con't)

- Shazeer et al. (2017)
  - routing to $k > 1$ experts
  - intuition: learning to route would not work without the ability to compare at least two experts
- Switch layer
  - routes to only a *single* expert
  - preserves model quality
  - reduces routing computation
  - performs better



1e
2e
4e
8e
16e
32e
64e
128e
256e
6.0
5.8
5.6
5.4
5.2
5.0
4.8
Test Loss
10^9
10^{10}
Sparse Model Parameters



Switch-Base: 128e
Switch-Base: 64e
Switch-Base: 32e
Switch-Base: 16e
T5-Base
Neg Log Perplexity
−1.2
−1.3
−1.4
−1.5
−1.6
−1.7
−1.8
−1.9
−2.0
0
1
2
3
4
Training Step
1e5

Mixture of Expert Routing

The MoE layer takes as an input a token representation x and then routes this to the best determined top-k experts, selected from a set $\{E_i(x)\}_{i=1}^N$ of N experts.

The router variable W_r produces logits $h(x) = W_r \cdot x$, which are normalized via a softmax distribution over the available N experts at that layer. The gate value for expert i is given by:

$$p_i(x) = \frac{e^{h(x)_i}}{\sum_j e^{h(x)_j}}$$

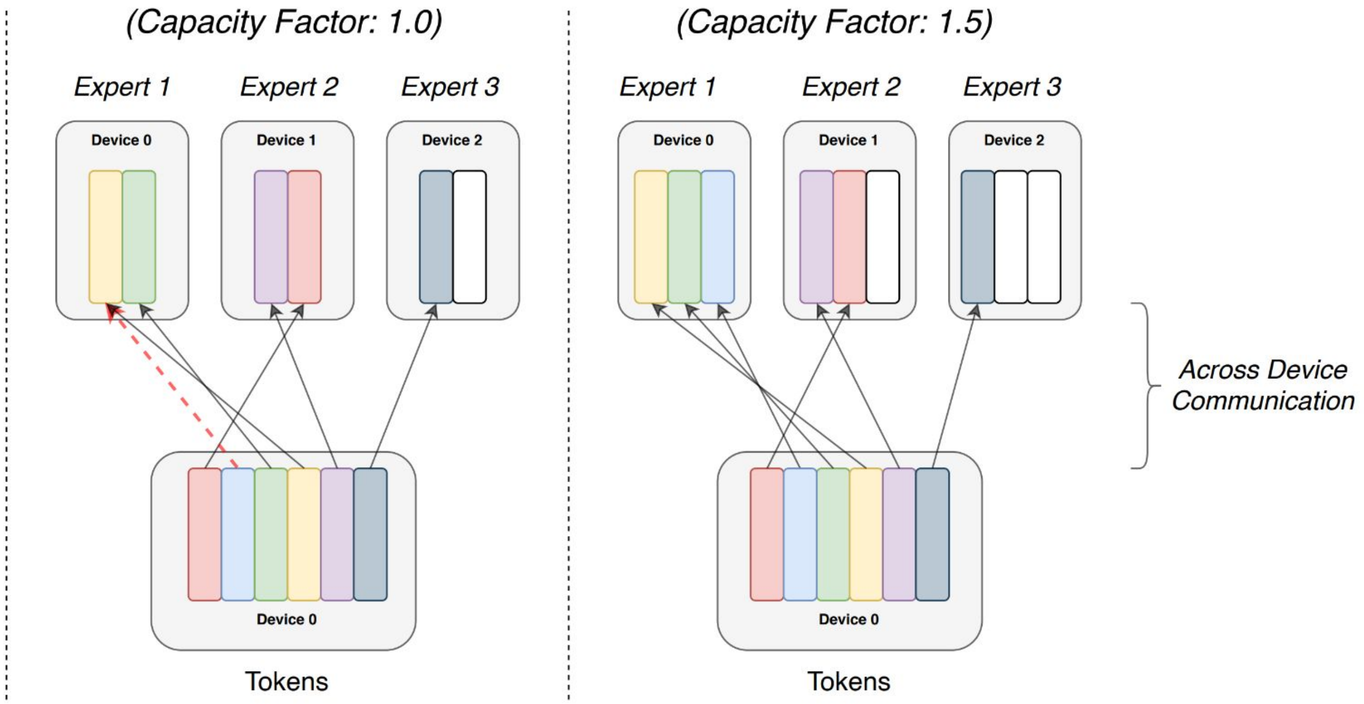
The top-k gate values are selected for routing the token x. If T is the set of selected top-k indices, then the output computation of the layer is the linearly weighted combination of each expert's computation on the token by the gate value:

$$y = \sum_{i \in T} p_i(x) E_i(x)$$

$$\text{expert capacity} = \left(\frac{\text{tokens per batch}}{\text{number of experts}}\right) \times \text{capacity factor}$$

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.

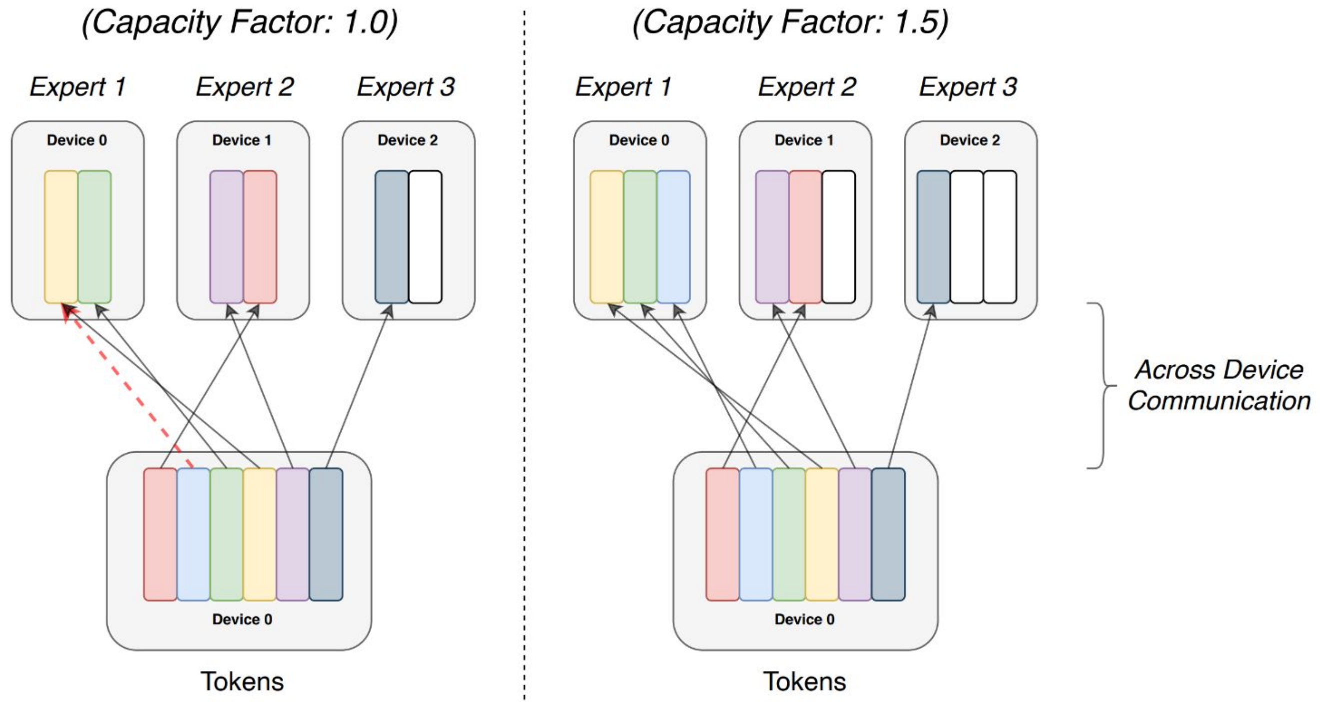


What would happen with the blue (dropped) token?

$$\text{expert capacity} = \left(\frac{\text{tokens per batch}}{\text{number of experts}}\right) \times \text{capacity factor}$$

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.



overflow

wasted computation & memory

An auxiliary load balancing loss

Given N experts indexed by $i = 1$ to N and a batch B with T tokens, the auxiliary loss is computed as the scaled dot-product between vectors f and P:

$$\text{loss} = \alpha \cdot N \sum_{i=1}^{N} f_i \cdot P_i$$

f_i: fraction of tokens to expert i

- ***For each token in the batch, check which expert got chosen.***
- ***Count how many times expert i was picked.***
- ***Divide by the total number of tokens T to get the fraction.***

P_i: router probability for expert i

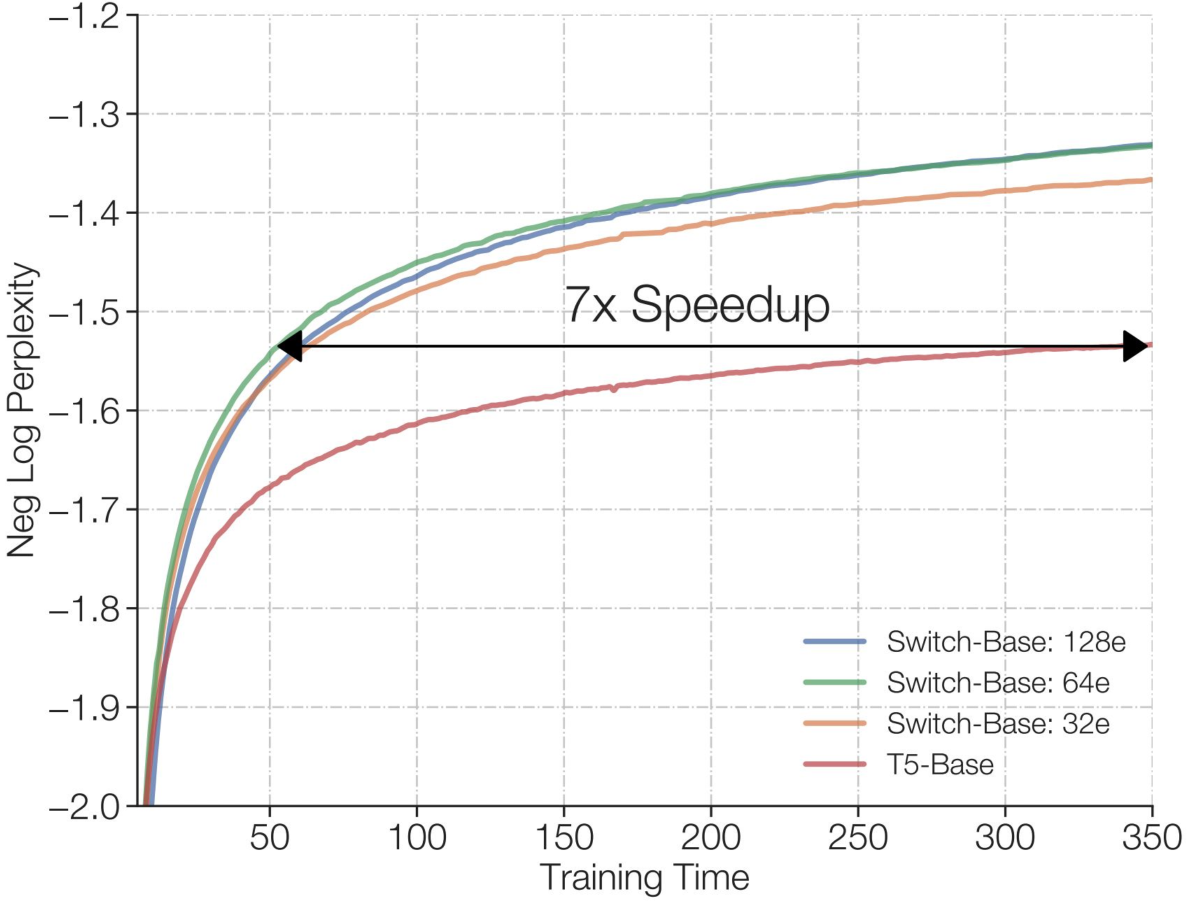
- ***The router assigns probabilities to each expert for every token.***
- ***For each token, take the probability that expert i i was preferred.***
- ***Average over all tokens.***

The auxiliary loss encourages uniform routing since it is minimized under a uniform distribution

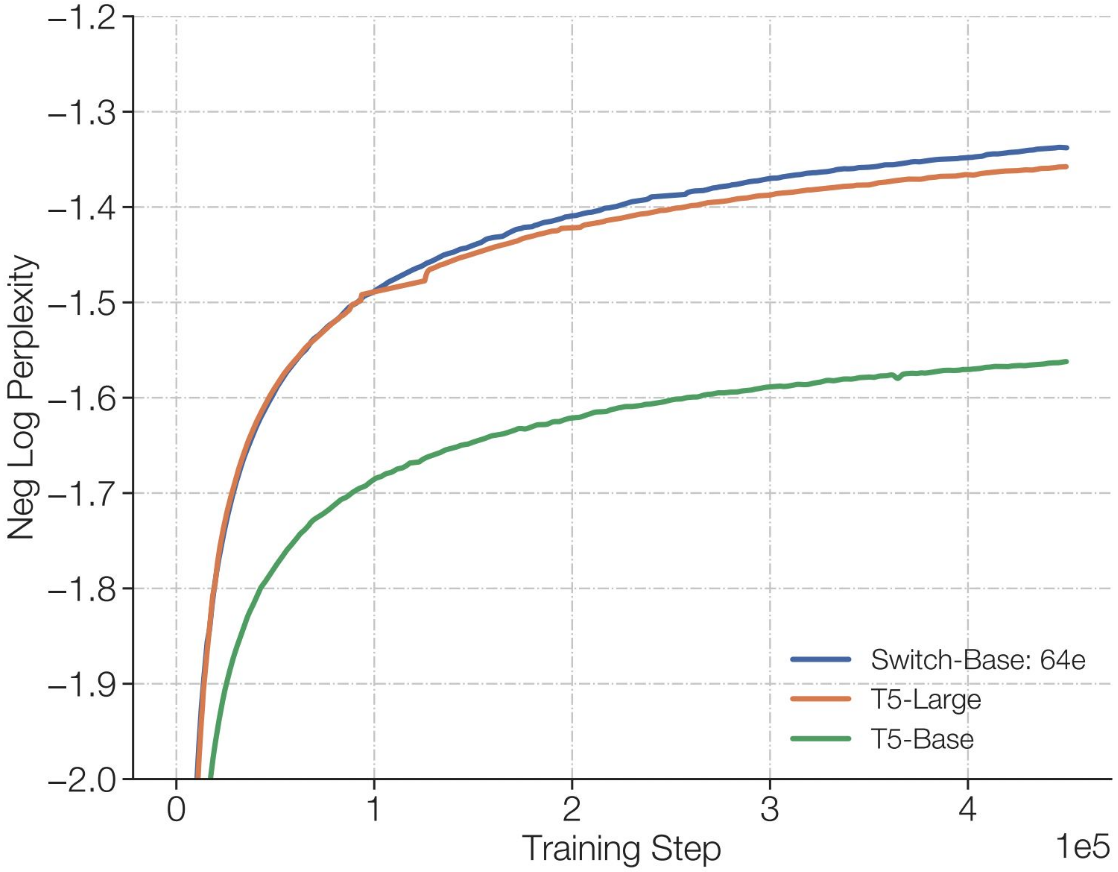
Lower standard dropout rate for non-expert layers, higher for expert feed-forward layers

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**

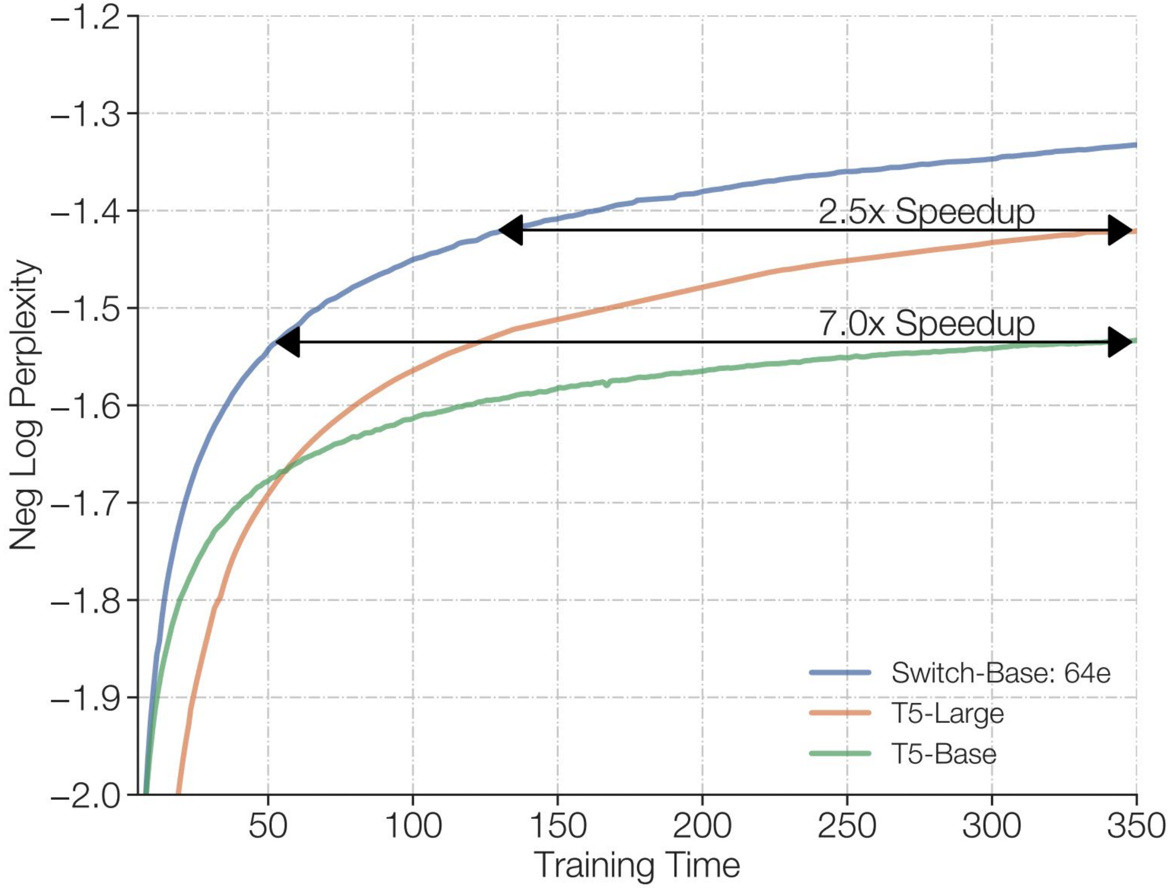
Speed advantage of Switch Transformer



Switch-Base is more sample efficient than T5-Large



Switch-Base is faster than T5-Large (2.5x speedup)



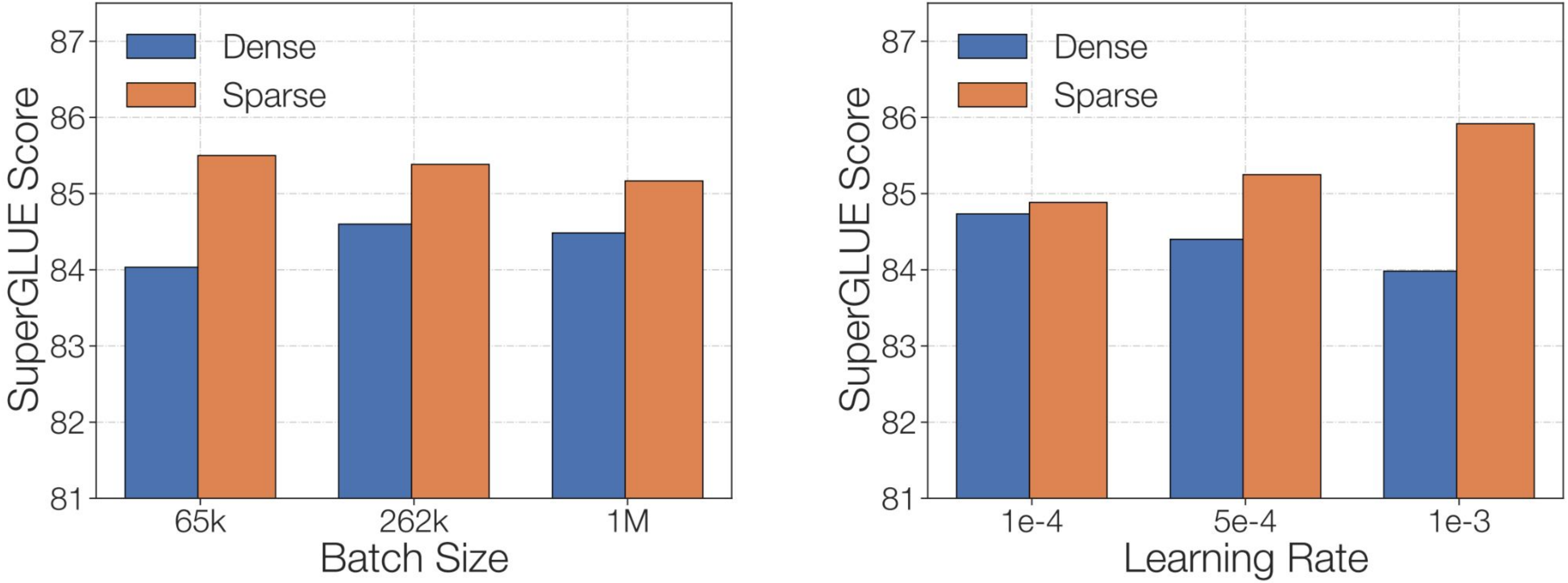
… and significant downstream improvements

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**

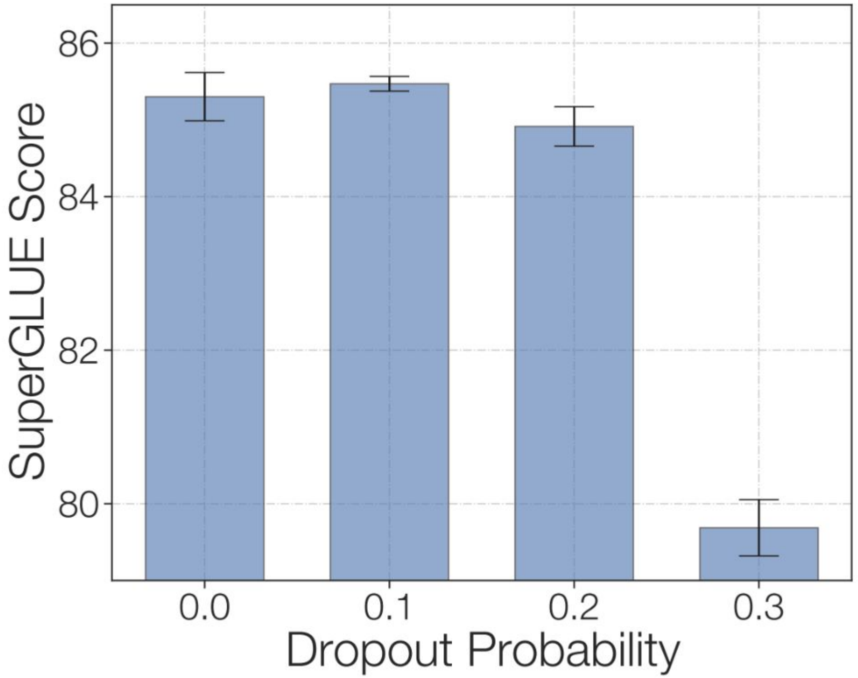
Model	CB Web QA	CB Natural QA	CB Trivia QA
T5-Base	26.6	25.8	24.5
Switch-Base	**27.4**	**26.8**	**30.7**
T5-Large	27.7	27.6	29.5
Switch-Large	**31.3**	**29.5**	**36.9**

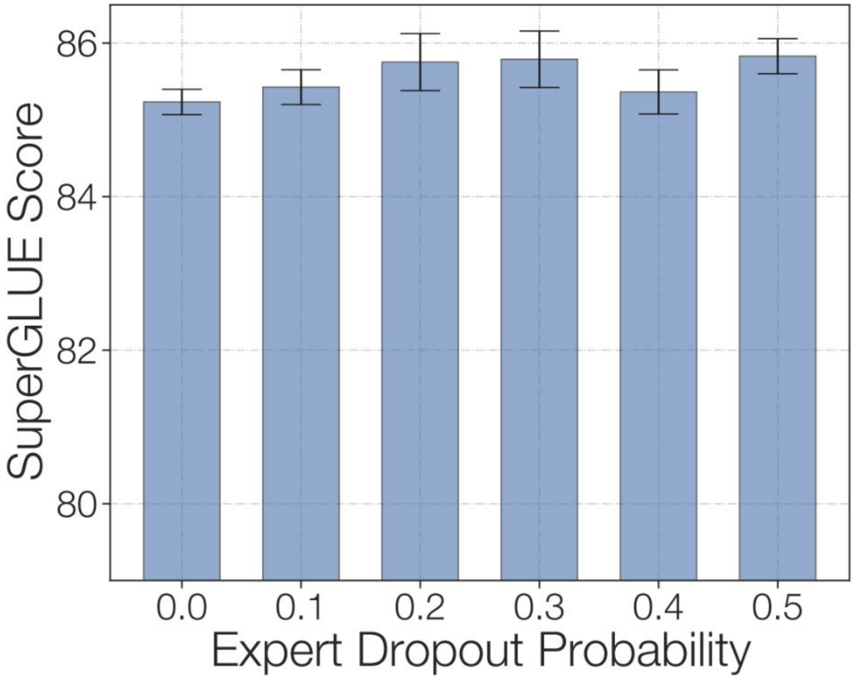
Sparse models benefit from small batch sizes and high learning rates



[“ST-MoE: Designing Stable and Transferable Sparse Expert Models” by Zoph et al. (2022)]

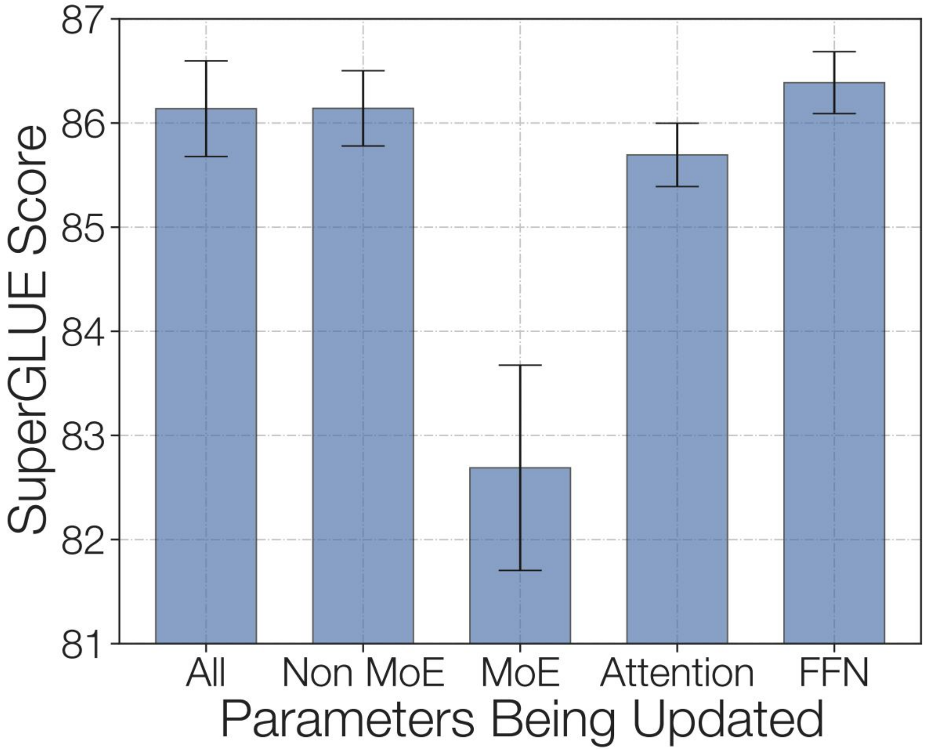
Sparse models benefit from high dropout rates





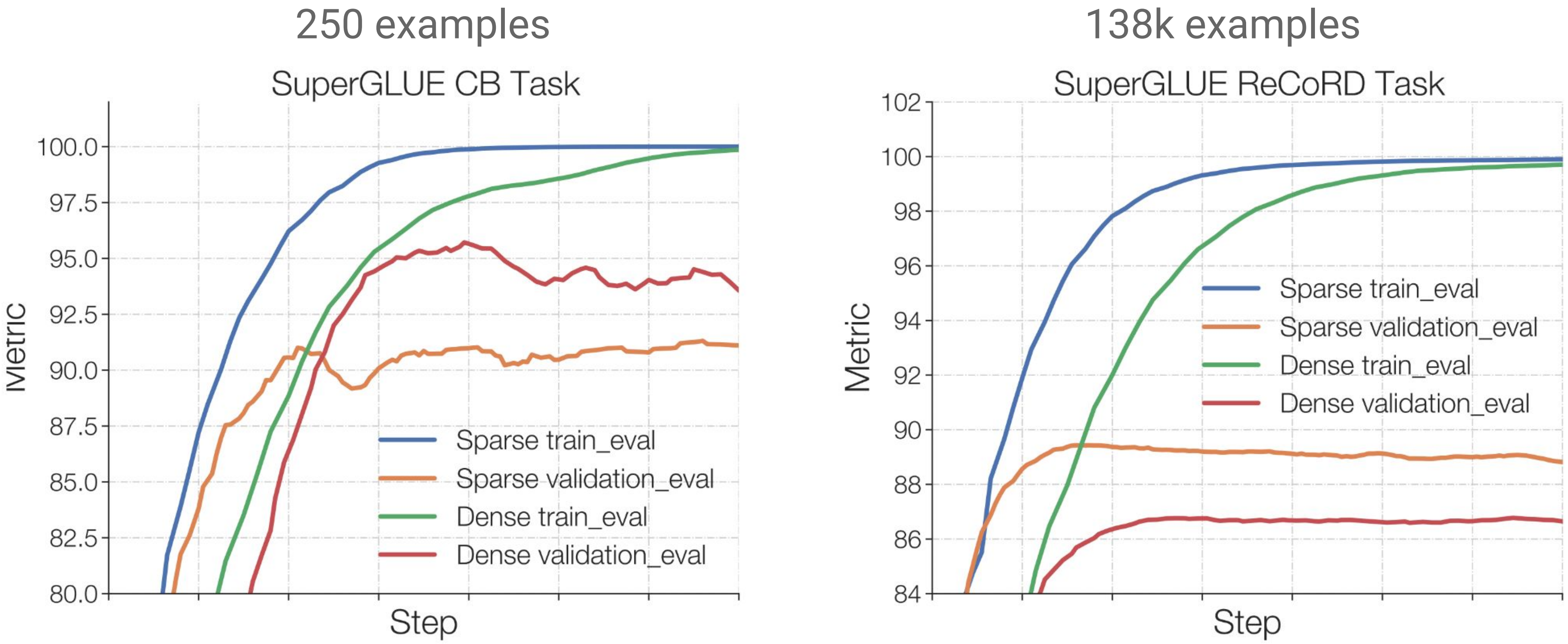
[“ST-MoE: Designing Stable and Transferable Sparse Expert Models” by Zoph et al. (2022)]

For fine-tuning: by freezing the MoE layers, we can speed up the training while preserving the quality



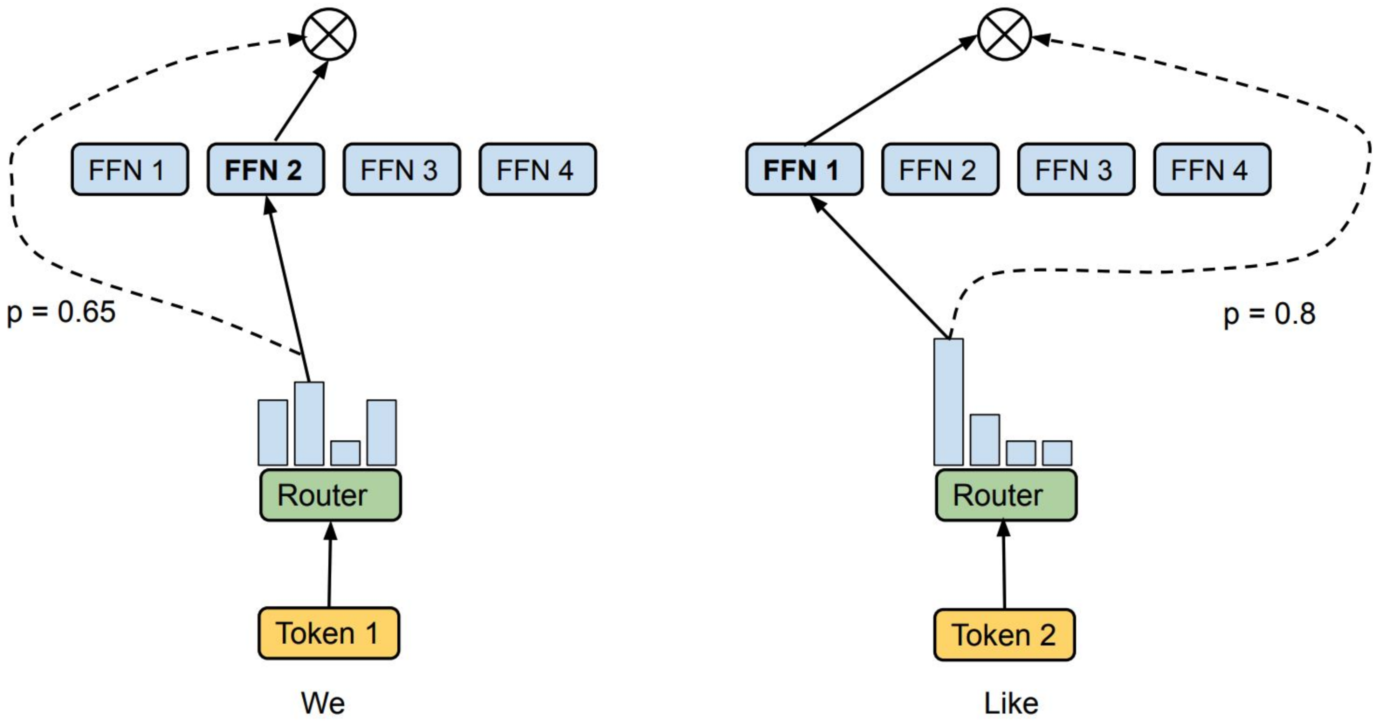
[“ST-MoE: Designing Stable and Transferable Sparse Expert Models” by Zoph et al. (2022)]

Sparse models are prone to overfit



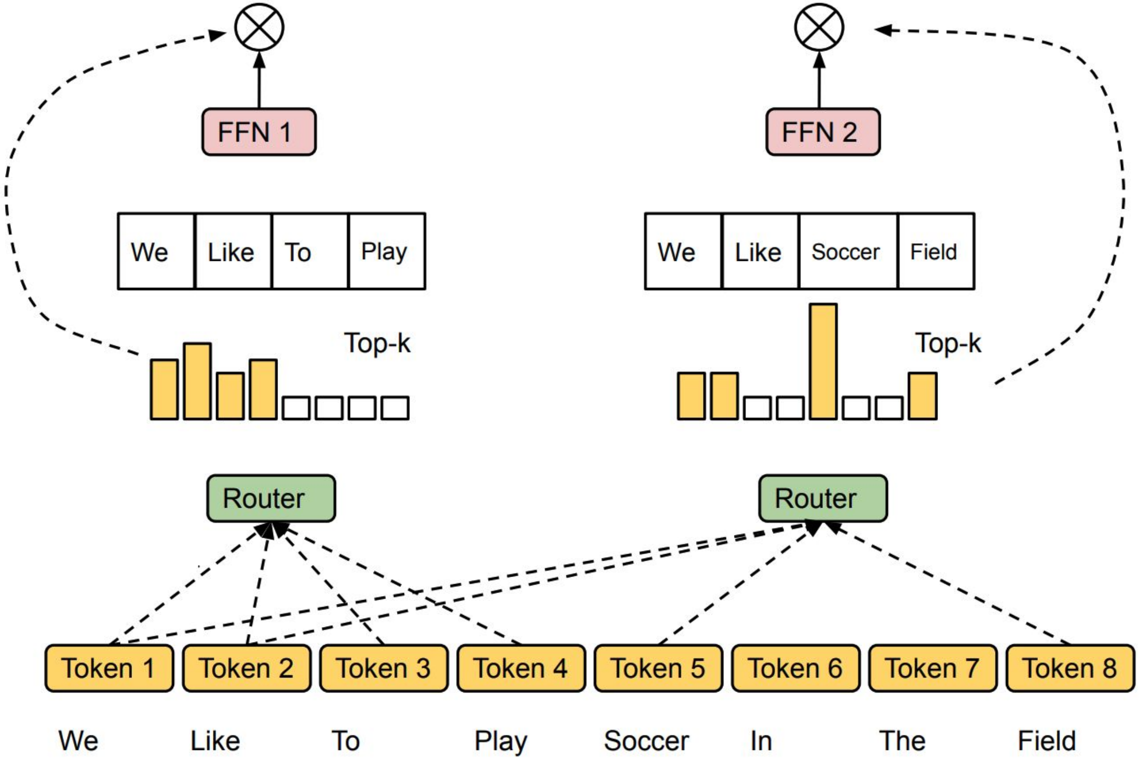
[“ST-MoE: Designing Stable and Transferable Sparse Expert Models” by Zoph et al. (2022)]

Token-choice routing



[“Mixture-of-Experts with Expert Choice Routing” by Zhou et al. (2022)]

Expert-choice routing



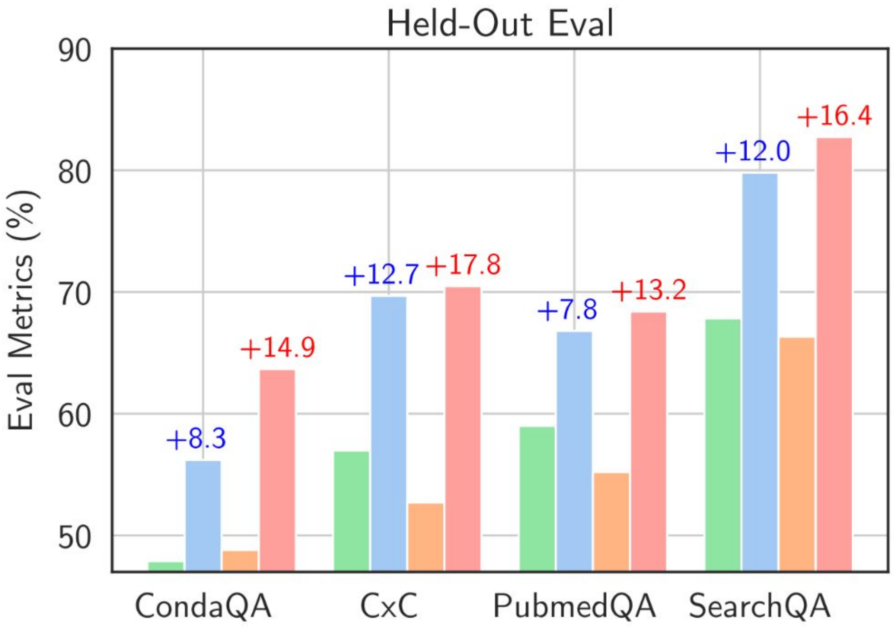
[“Mixture-of-Experts with Expert Choice Routing” by Zhou et al. (2022)]

Mixture-of-Experts Meets Instruction Tuning: A Winning Combination for Large Language Models

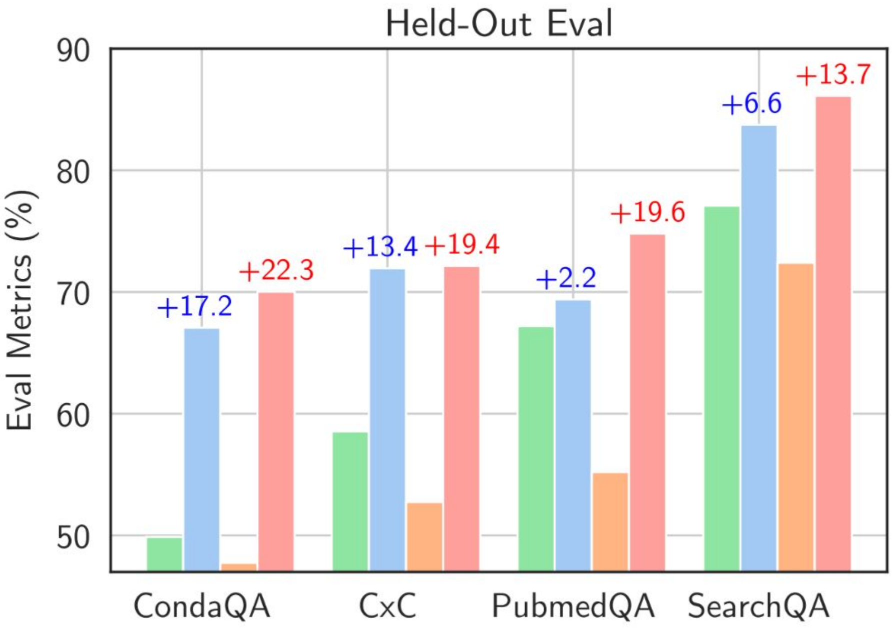
Sheng Shen[♮*] **Le Hou**[†] **Yanqi Zhou**[†] **Nan Du**[†] **Shayne Longpre**[⊤*] **Jason Wei**[†],
Hyung Won Chung[†] **Barret Zoph**[†] **William Fedus**[†] **Xinyun Chen**[†] **Tu Vu**[‡*],
Yuexin Wu[†] **Wuyang Chen**[§*] **Albert Webson**[†] **Yunxuan Li**[†] **Vincent Zhao**[†] **Hongkun Yu**[†]
Kurt Keutzer[♮] **Trevor Darrell**[♮] **Denny Zhou**[†]

[†]Google [♮]University of California, Berkeley [⊤]Massachusetts Institute of Technology
[‡]University of Massachusetts Amherst [§]The University of Texas at Austin

Mixture-of-Experts meets Instruction Tuning



(a) FLAN-EC$_{\text{BASE}}$ v.s. FLAN-T5$_{\text{BASE}}$



(b) FLAN-EC$_{\text{LARGE}}$ v.s. FLAN-T5$_{\text{LARGE}}$

T5 → FT | Flan-T5 → FT | MoE → FT | Flan-MoE → FT

When to use sparse MoEs vs dense models?

Experts are useful for high throughput scenarios with many machines. Given a fixed compute budget for pretraining, a sparse model will be more optimal. For low throughput scenarios with little VRAM, a dense model will be better.

https://huggingface.co/blog/moe

Thank you!