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

William Fedus, Barret Zoph, Noam Shazeer

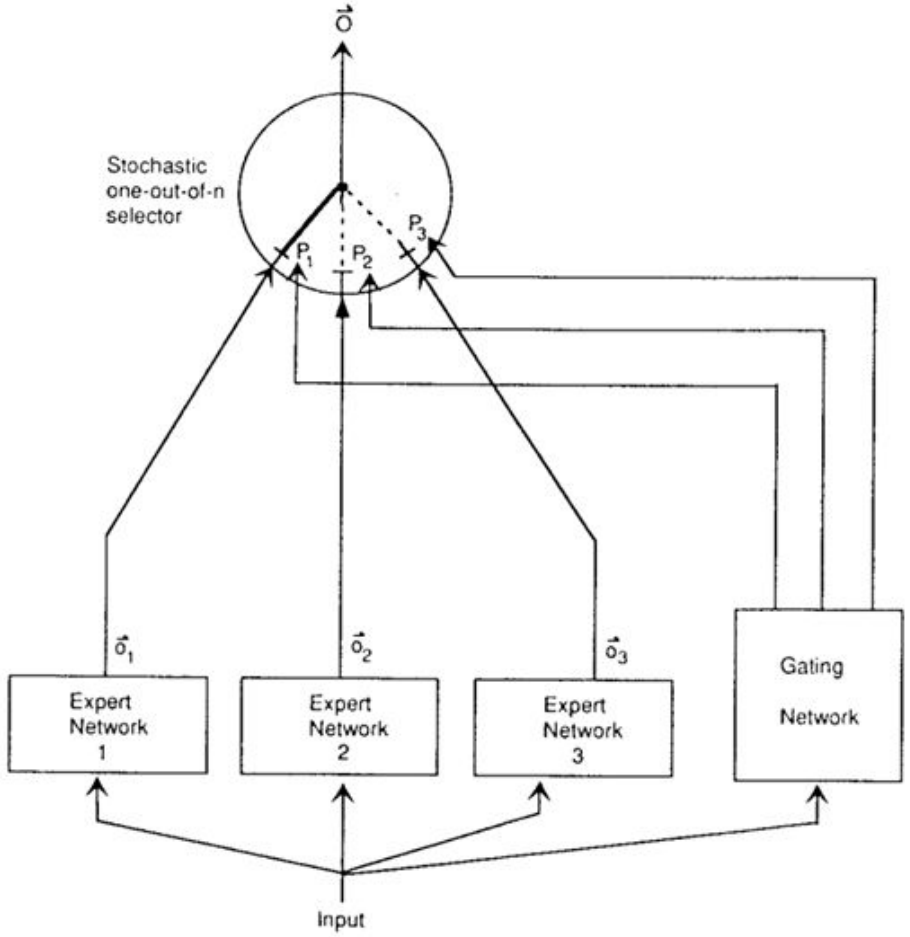
Presented by Aryan Bhardwaj

Contents

Mixture of Experts

Background of MoE (~30 years ago)

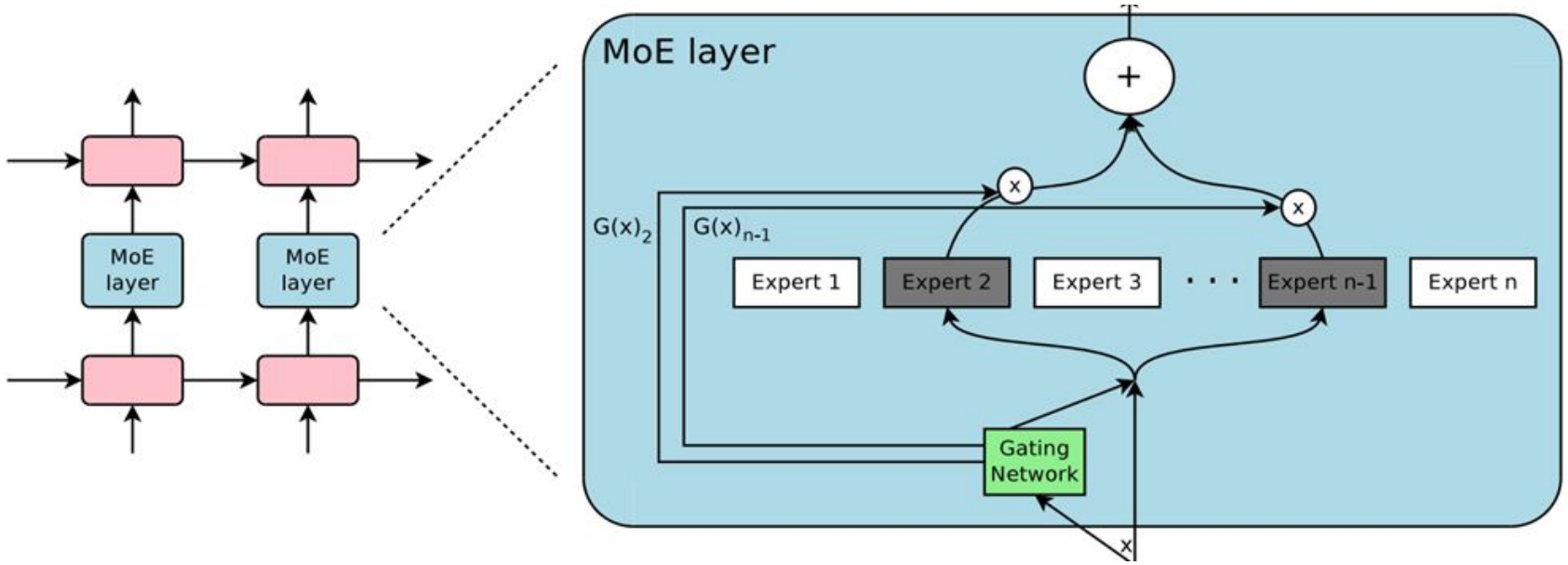
- Dates back at least 30 years to the work *Adaptive Mixtures of Local Experts*[1].
- In early concepts, an expert was defined as an entire neural network and the MoE was similar to ensemble methods[2].



[1] R. A. Jacobs, M. I. Jordan, S. J. Nowlan and G. E. Hinton, "Adaptive Mixtures of Local Experts," , Neural Computation
[2] William Fedus et. al. A Review of Sparse Expert Models in Deep Learning

Background of MoE (RNN Era)[3]

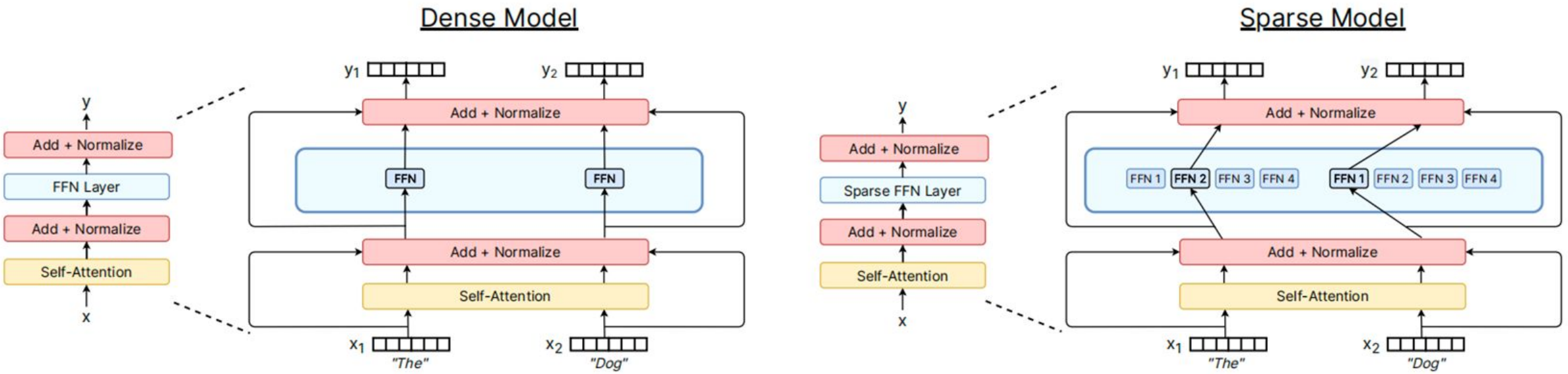
- Backbone: RNN
- MoE layer: multiple FFN experts
- Gating/routing network (also a FFN): assign tokens into different experts



[3] 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

Background of MoE (Transformer Era)

- Replace the original FFN layers with sparse FFN ones (expert layers).



Motivation

Compared to dense models, MoE has two major advantages:

1. **More unique model parameters** can be included given similar computational budgets (FLOPs), which improves model performance empirically;

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

Motivation

2. **Functional specialization** among expert modules, which improves the model interpretability.

Expert	Top-5 preceding tokens
5	year, years, billion, millions, tonnes
9	electronic, local, public, national, outdoor
34	to, will, should it, may
42	two, 50, 1, 80, 000
62	work, started, involved, working, launched
72	is, was, be, been, were
74	going, go, come, back, return
101	B, T, W, H, k

Switch Transformer

Switch Routing: Rethinking Mixture-of-Experts.

Mixture of Experts

Route to k experts, where k > 1

$$h(x) = W_r \cdot x$$

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

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

Switch Transformer

Only route to a single expert (k = 1)

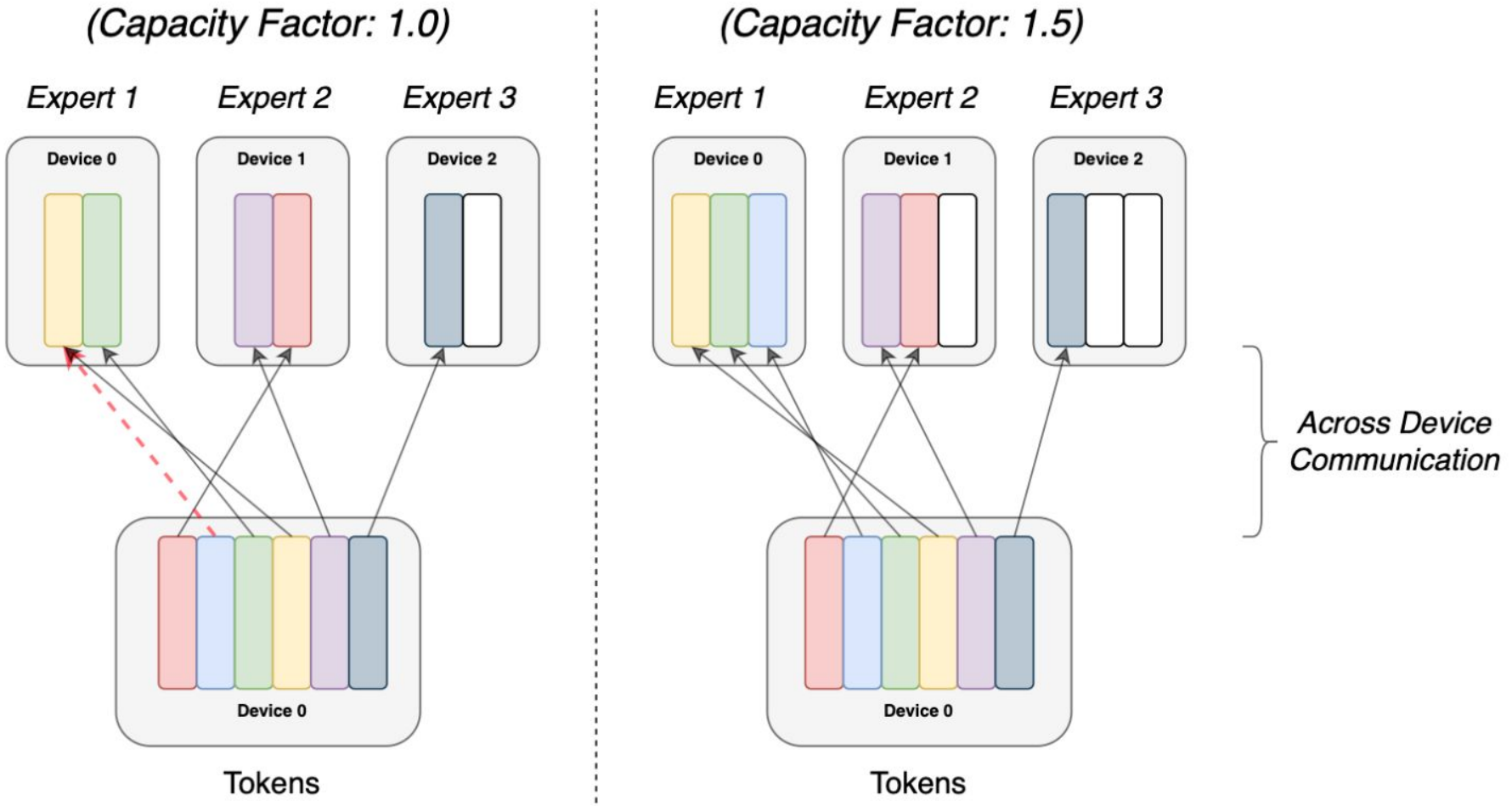
Benefits:

1. Router computation is reduced.
2. Batch size of each expert can be halved.
3. Communication costs are reduced.

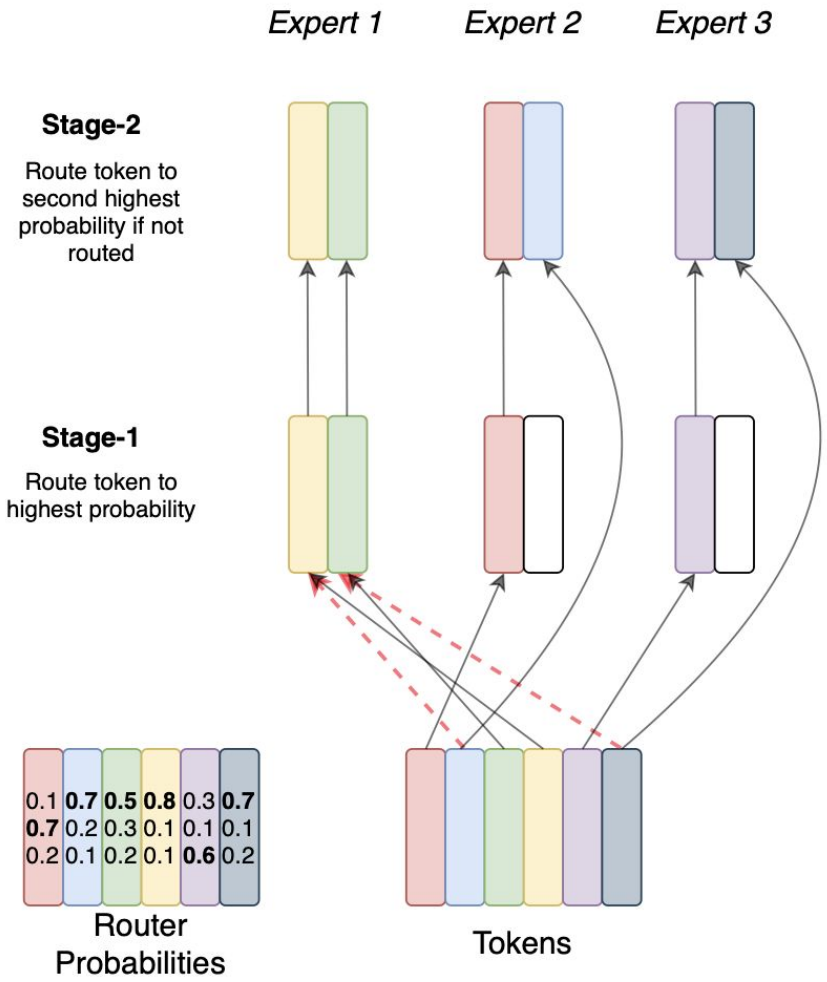
Expert Capacity

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.



Preventing Token Dropping with No-Token-Left-Behind



A Differentiable Load Balancing Loss

$$\text{loss} = \alpha \cdot N \cdot \sum_{i=1}^{N} f_i \cdot P_i \tag{4}$$

where f_i is the fraction of tokens dispatched to expert i,

$$f_i = \frac{1}{T} \sum_{x \in \mathcal{B}} \mathbb{1}\{\text{argmax}\, p(x) = i\} \tag{5}$$

and P_i is the fraction of the router probability allocated for expert i, [2]

$$P_i = \frac{1}{T} \sum_{x \in \mathcal{B}} p_i(x). \tag{6}$$

Improved Training and Fine-Tuning Techniques

Issues with Mixture of Experts (MoE)

- Model complexity
- High communication costs
- Training instabilities

To improve training and fine-tuning of sparse models, Switch Transformers used…

1. Selective precision with large sparse models.
2. Smaller parameter initialization for stability.
3. Regularizing large sparse models.

1. Selective precision with large sparse models

- Model instability hinders the ability to train using efficient bfloat16 precision, so float32 precision was used for MoE.
- **Solution:** Selectively cast to float32 precision
  - Calculations within the router function are done in float32 precision
  - Resulting dispatch and combine tensors are recast to bfloat16

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

2. Smaller parameter initialization for stability

They found that sparse expert models are sensitive to initialization scale. **They reduced the default Transformer initialization scale s = 1.0 by a factor of 10.**

Model (Initialization scale)	Average Quality (Neg. Log Perp.)	Std. Dev. of Quality (Neg. Log Perp.)
Switch-Base (0.1x-init)	**-2.72**	**0.01**
Switch-Base (1.0x-init)	-3.60	0.68

3. Regularizing large sparse models

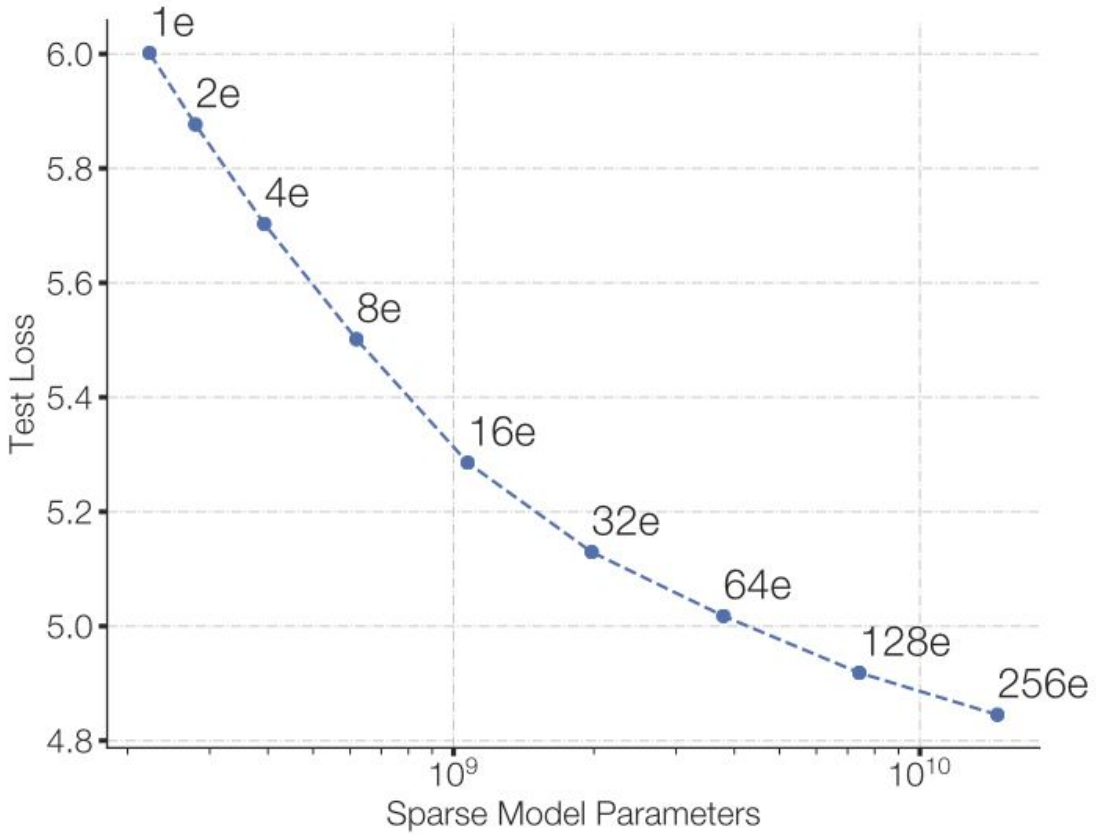
Consider pretraining and then fine-tuning on smaller downstream tasks. Switch transformers have high parameter counts and are prone to overfitting.

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**

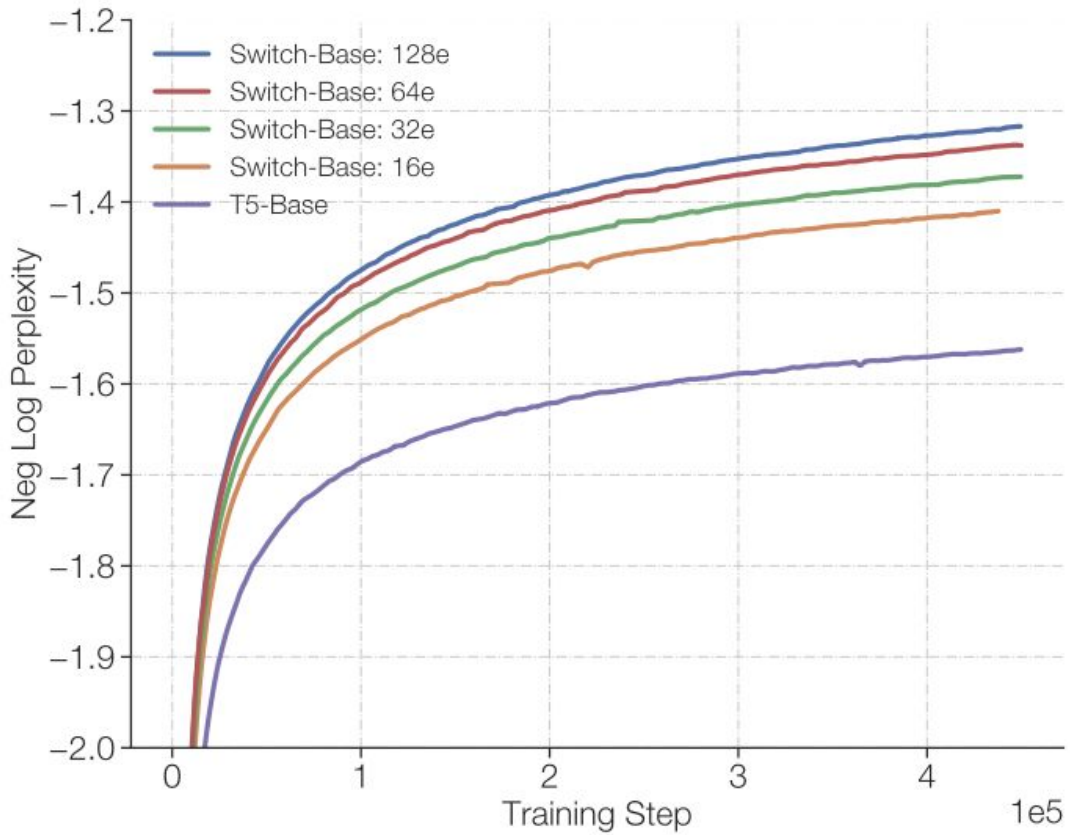
Scaling Properties

Scaling Results on a Step-Basis

Number of experts

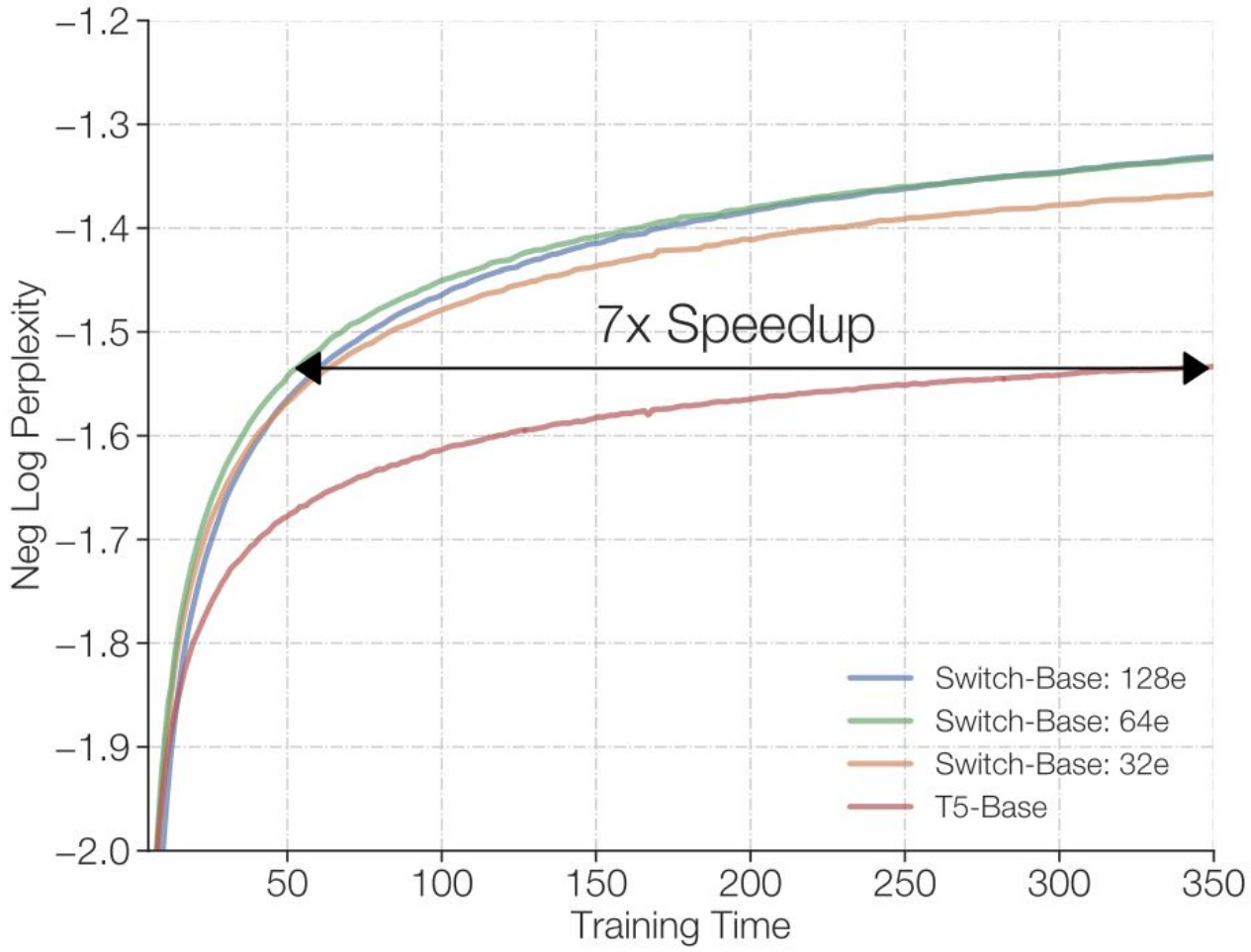


Training steps



Scaling Results on a Time-Basis

Wall-clock time



Fine-Tuning

Fine-tuning results

For most tasks, we found significant improvements of the Switch Transformer variants.

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**

Model Distillation

Distilling Switch Transformers for Language Modeling

- Initialized the student model with weights from the teacher's non-expert layers
- Used a mix of probabilities from the teacher model (25%) and ground truth labels (75%)

Could keep roughly 30% of the the performance improvements even after distilling a model with 100x more parameters back into a small dense model.

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

Distillation compression rates

	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 %

Distilling a fine-tuned model

- Distill a Switch-Base model fine-tuned on the SuperGLUE tasks into a T5-Base model

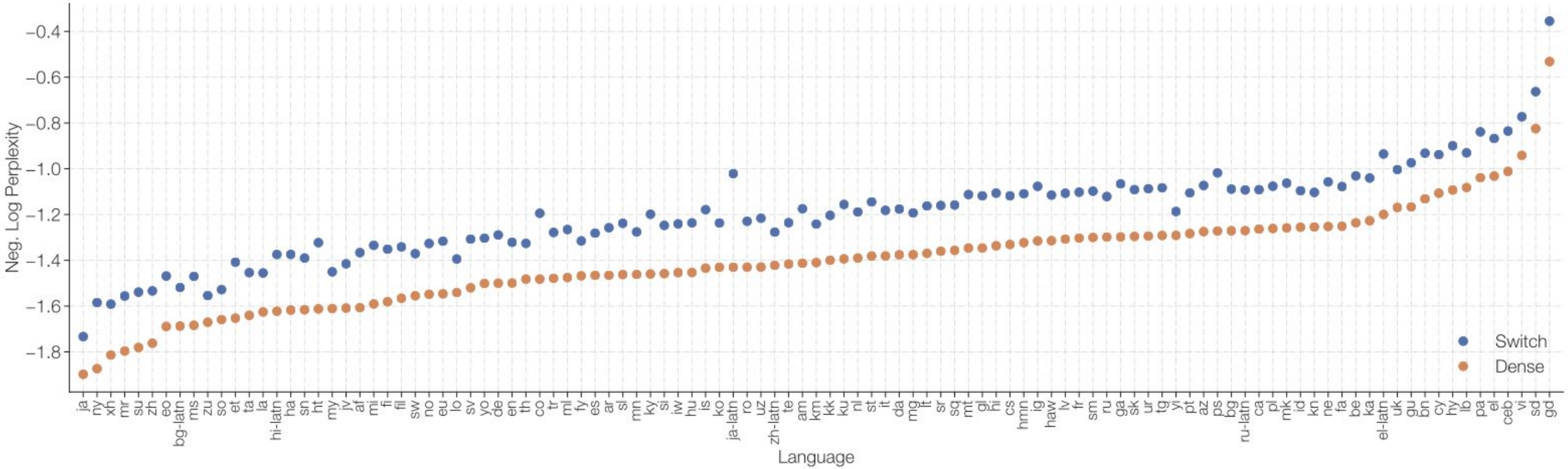
Again, we achieve 30% of the teacher's performance on a 97% compressed model.

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

Multilingual Learning

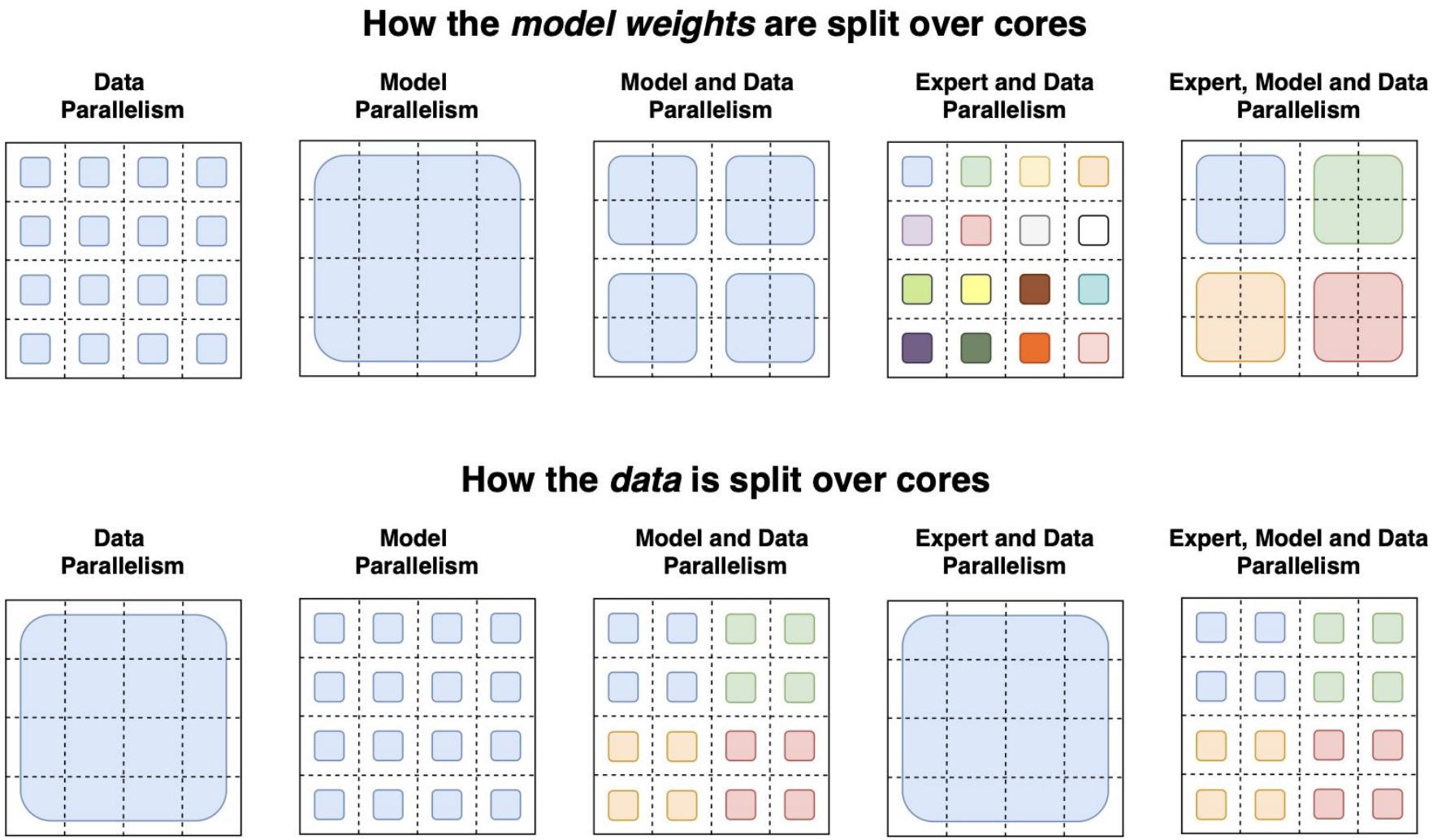
Multilingual Learning

- Pretrain on the multilingual variant of the Common Crawl data set (mC4), spanning 101 languages



Data and Model Partitioning Strategies

Designing Models with Data, Model, and Expert-Parallelism



Towards Trillion Parameter Models

Designing Models with Data, Model, and Expert-Parallelism

- Create Switch-C (395 billion parameters) and Switch-XXL (1.6 trillion parameters) using expert, model, and data parallelism techniques.

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. Experts	Num Layers	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

My Final Thoughts

- Is a novel approach to scaling large models with a lot of parameters without a proportional increase in computational cost.
- Has strong knowledge task performance, but has uneven performance on reasoning tasks
- However, need to further improve training stability for the largest models like Switch-XXL.
- Need to research more into how sparse models scale in relation to different hardware configurations
- Extend Switch Transformers to other modalities, like image or audio data

Thank you!