## Mixture of Experts meets Instruction Fine Tuning

## Paper

* <https://arxiv.org/pdf/2305.14705.pdf>

## Why was it made

* Says that llm are computationally expensive
* Mixture of experts reduces computational overhead
* Proposed model Combines instruction fine tuning and Mixture of experts

## Mixture of Experts

* Idea that language models can be **decomposed into smaller, specialized sub-models**, or "experts", that focus on distinct aspects of the input data, thereby enabling more efficient computation and resource allocation
* However typical Moe leads to lower performance
* But its better if they use **both** instruction tuning and MoE

## Model architecture

* Replaces every feedforward layer in transformer with MoE layer
  + Similar to Switch transformer ( <https://arxiv.org/pdf/2101.03961.pdf> )
* Each MoE layer consists of a collection of independent feed-forward networks as the **‘experts’**.
  + Interesting that its a layer with networks
  + Only a subset of those experts is used which reduces computation
* Gating function uses softmax activation over the experts
  + Models a probability distribution
  + Distribution indicates how well experts can handle the input
* The gating network is **trained** to only use the best 2 experts for each **token** of an input sequence
* During inference, the learned gating network dynamically picks the two best experts for each token
* For an MoE layer with E experts, it provides O(E^2) different combinations of feed-forward networks instead of one in the classic Transformer architecture,
* final learned representation of a token will be the weighted combination of the outputs from the selected experts.

## 

## Instruction fine tuning recipe

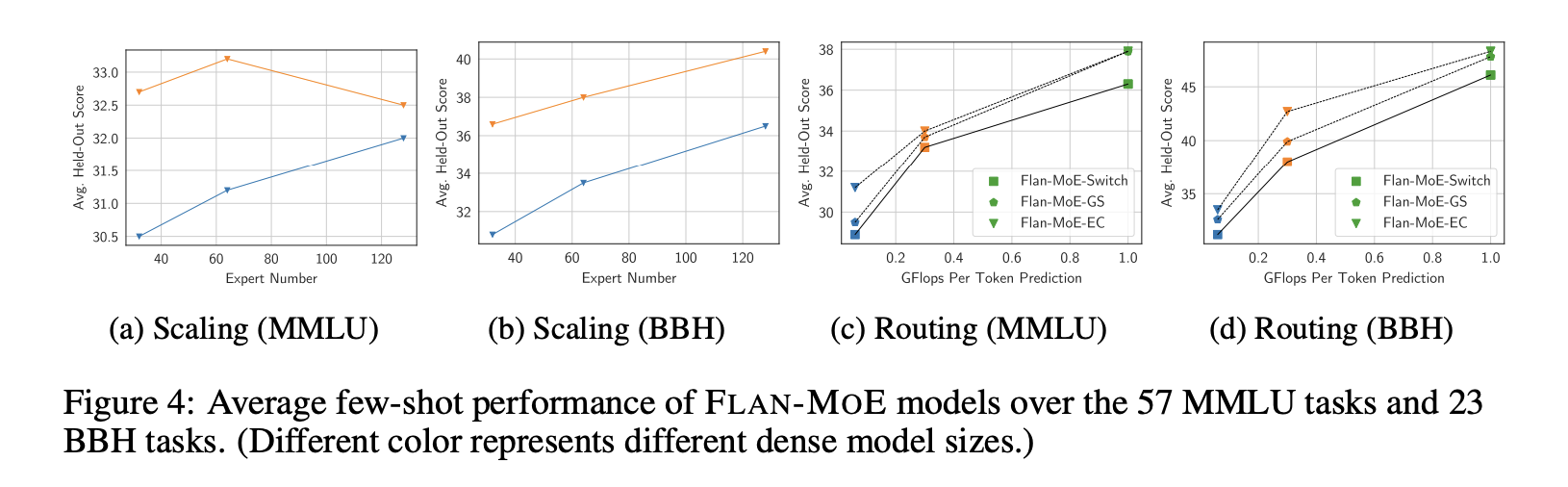
* fine-tune FLAN-MOE using the **prefix language model objective** on the FLAN collective dataset
  + Prefix language model objective is
  + <https://blog.research.google/2021/10/introducing-flan-more-generalizable.html>
* Uses the auxiliary loss during training
  + They tested both router Z loss and balancing loss. Balancing loss was better [ON CERTAIN TASKS]
    - **Mismatched** auxiliary loss and routing strategy could mean **lower performance**
* All model parameters are trained

## Experiment Details

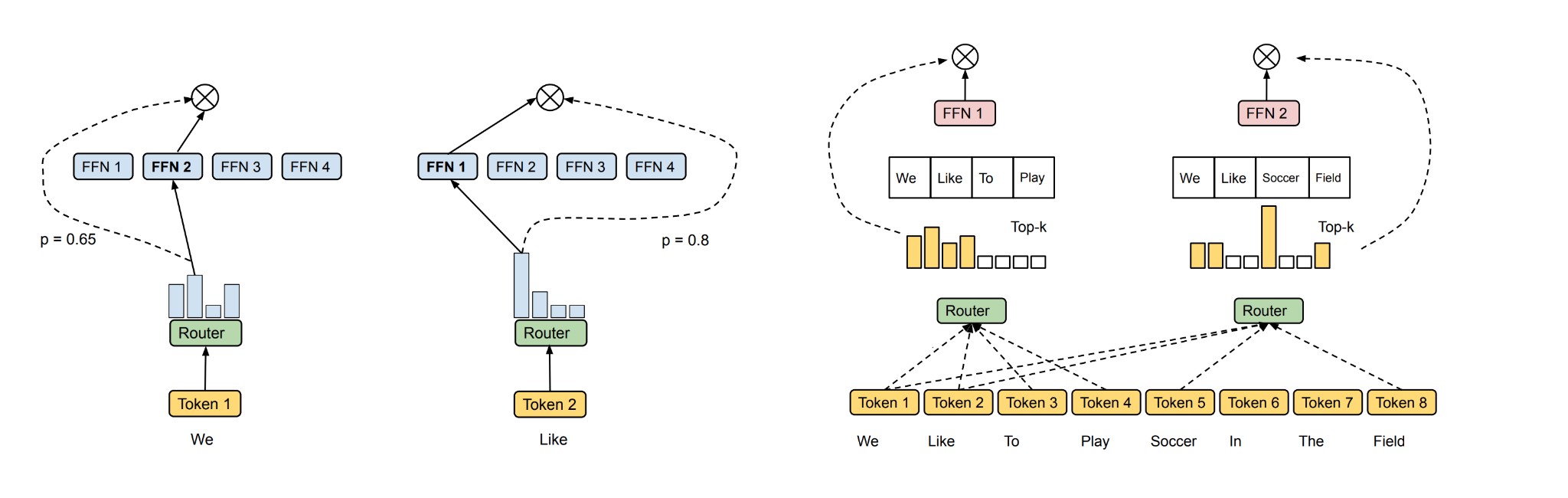
* In the context of instruction fine tuning
* Compares Flan-Moe to T5 models
  + T5 models are encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks and for which each task is converted into a text-to-text format
* Trained on 1,836 finetuning tasks
  + FLAN-MOE had lower flops
* Evaluated on both zero shot and few shot tasks
* Results: **The model FLAN-ST32B, comprising a total of 32 billion parameters, only utilizes 32.1 GFLOPs per token, which amounts to merely one-third of the computational power required by a FLAN-PALM62B model**
* Additionally, **all the routers combined account for less than 4 million parameters**

## Expert Number

* The performance of FLAN-MOE models has been observed to scale with the number of experts included in the architecture to a certain threshold.



## Routing Strategy

* **routing strategy intelligently distributes input data among multiple specialized experts**
* Important for maximizing the utilization of the model’s capacity while minimizing the risk of **overfitting**
* two trending strategies
  + **token-choice** [23] which lets the **token select** the top-K experts
  + **expert-choice** [55] which lets the **experts select** the top-K tokens.
* FLAN-Switch: checkpoints from Switch Transformer top-1 token-choice gating
* FLAN-GS: GShard top-2 token-choice gating
* FLAN-EC: expert choice top-2 gating (FLAN-EC) models pre-trained on the same GLaM dataset
  + Probably based off <https://arxiv.org/abs/2202.09368> ?
  + Instead of letting tokens select the top-k experts, we have experts selecting the top-k tokens
  + 
  + Conventional vs non conventional moe
  + “We propose a very simple yet effective routing method we are calling expert choice. Unlike conventional MoE where tokens select one or two top-scoring experts, our method lets each expert pick the top-k tokens. Our method guarantees perfect load balancing, allows a variable number of experts for each token, and achieves substantial gains in training efficiency and downstream performance as demonstrated in our experiments”
* “It is noteworthy that the performance gap between the token choice and expert-choice models can be bridged when we incorporate advanced auxiliary loss and pre-training strategy as exhibited in ST-MOE “

## Random notes

* Seemingly flan-moe == flan-St

## More Learning

* <https://github.com/XueFuzhao/awesome-mixture-of-experts>

## Reference Papers

* <https://blog.research.google/2021/10/introducing-flan-more-generalizable.html>
* <https://arxiv.org/abs/2202.09368>
* <https://arxiv.org/pdf/2310.06825.pdf>
* <https://arxiv.org/pdf/2101.03961.pdf>
* <https://arxiv.org/pdf/2101.03961.pdf>
* <https://arxiv.org/pdf/2312.07987.pdf>
* <https://huggingface.co/blog/moe#what-is-sparsity>
* <https://github.com/mistralai/mistral-src/blob/main/tutorials/classifier.ipynb>
* <https://www.databricks.com/blog/introducing-mixtral-8x7b-databricks-model-serving>
* <https://arxiv.org/pdf/2210.11416.pdf>
* <https://huggingface.co/blog/moe#what-is-a-mixture-of-experts-moe>
* <https://arxiv.org/abs/2202.09368> mixture of expert with expert route choice
* <https://arxiv.org/abs/2101.03961> Switchhead

## Questions

* How are networks in the MoE layer initialized ?
* Which routing strategy did they introduce in Flan-MOE?