Google

GLaM: Efficient Scaling of Language Models with Mixture-of-Experts

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V Le, Yonghui Wu, Zhifeng Chen and Claire Cui

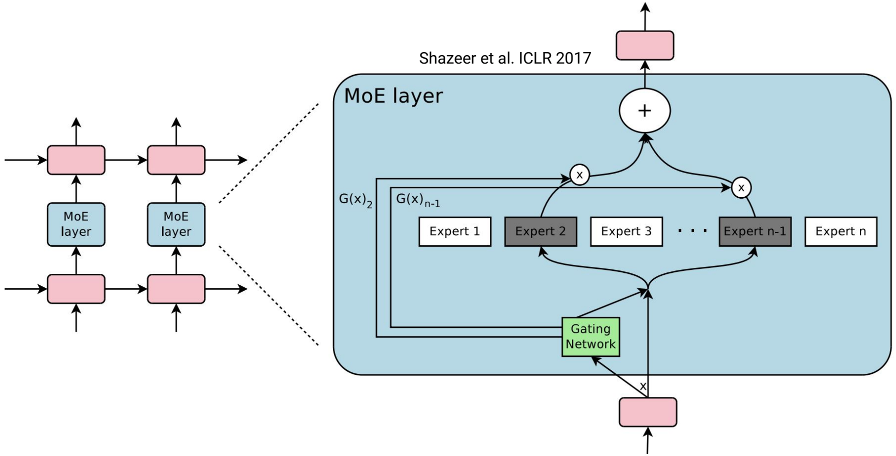
Motivations

- Scaling in both training data and model size has been the pivot to the success of giant large language models.
- Unfortunately, training cost increases ‘quadratically’ w.r.t both training data size and model size.
- We thus seek to solving this problem by training a family of autoregressive language models called GLaM, to strike a balance between *dense* and *conditional computation*.

Mixture of Experts (MoE)

An MoE layer includes

- A number of experts, each of which is a simple feed-forward network
- A trainable gating function mapping a subset of `best' experts for each input
- Final prediction is a weighted combination of the predictions from the select experts



$$G(x) = \mathrm{topK}(\mathrm{Softmax}(x \cdot W_G))$$

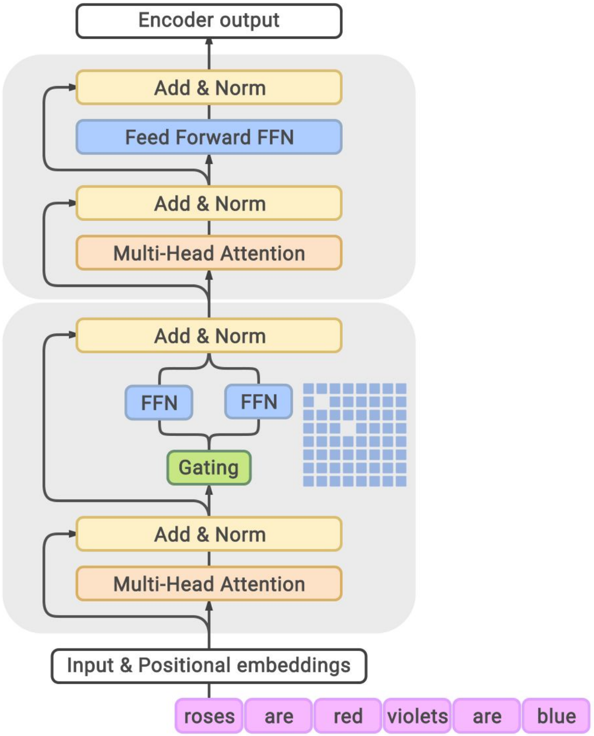
$$y = \sum_{i=1}^{K} \underbrace{G(x)_i}_{\text{expert weight}} \cdot \underbrace{E_i(x)}_{\text{expert output}}$$

expert weight

expert output

Model Architecture

- The FeedForward sub-component of every other Transformer layer in a stack is replaced with the MoE layer.
- Each token is routed to two experts (FFNs) chosen by the gating function.
- Decoupling the computation cost from the model size.
- Achieving almost constant computation cost per input as the model scales up.



Training Corpus

- Our dataset includes web pages, wikipedia, books, social media and news, etc.
- We have trained a linear classifier to remove low-quality web pages of which the languages are much different from to Wikipedia and Books.
- The final corpus has 90% english data and 10% non-english data.

Dataset	Tokens (B)	Weight in mixture
Filtered Webpages	143	0.42
Wikipedia	3	0.06
Conversations	174	0.28
Forums	247	0.02
Books	390	0.20
News	650	0.02

GLaM Models

- Both dense and MoE models are scaled up so that they have comparable activated number of parameters (similar predictive FLOPs) per token.
- The largest **GLaM (64B/64E)** has 1.2T parameters in total but only **96.6B** activated parameters per prediction
  - Nearly half of the 175B parameters of GPT-3
- All trained models share the same learning hyperparameters

GLaM Model	Type	n_{params}	$n_{\text{act-params}}$
0.1B	Dense	130M	130M
0.1B/64E	MoE	1.9B	145M
1.7B	Dense	1.7B	1.700B
1.7B/32E	MoE	20B	1.878B
1.7B/64E	MoE	27B	1.879B
1.7B/128E	MoE	53B	1.881B
1.7B/256E	MoE	105B	1.886B
8B	Dense	8.7B	8.7B
8B/64E	MoE	143B	9.8B
137B	Dense	137B	137B
64B/64E	MoE	1.2T	96.6B

Evaluation Protocol

The 29 benchmarks cover the following categories

- Cloze and Completion tasks
- Open-domain Question Answering
- Winograd-Style tasks
- Common Sense Reasoning
- In-context Reading Comprehension
- SuperGLUE
- Natural Language Inference

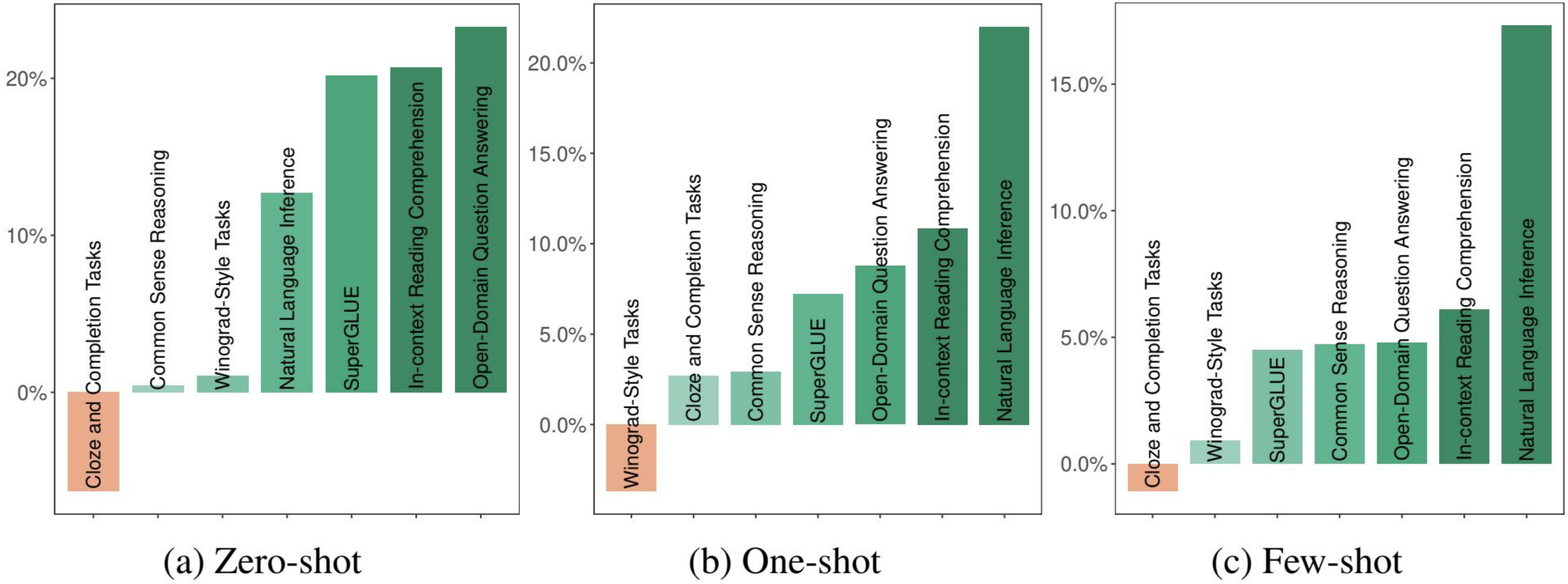
The same zero, one, and few-shot learning setup as GPT-3

Few-shot Performance

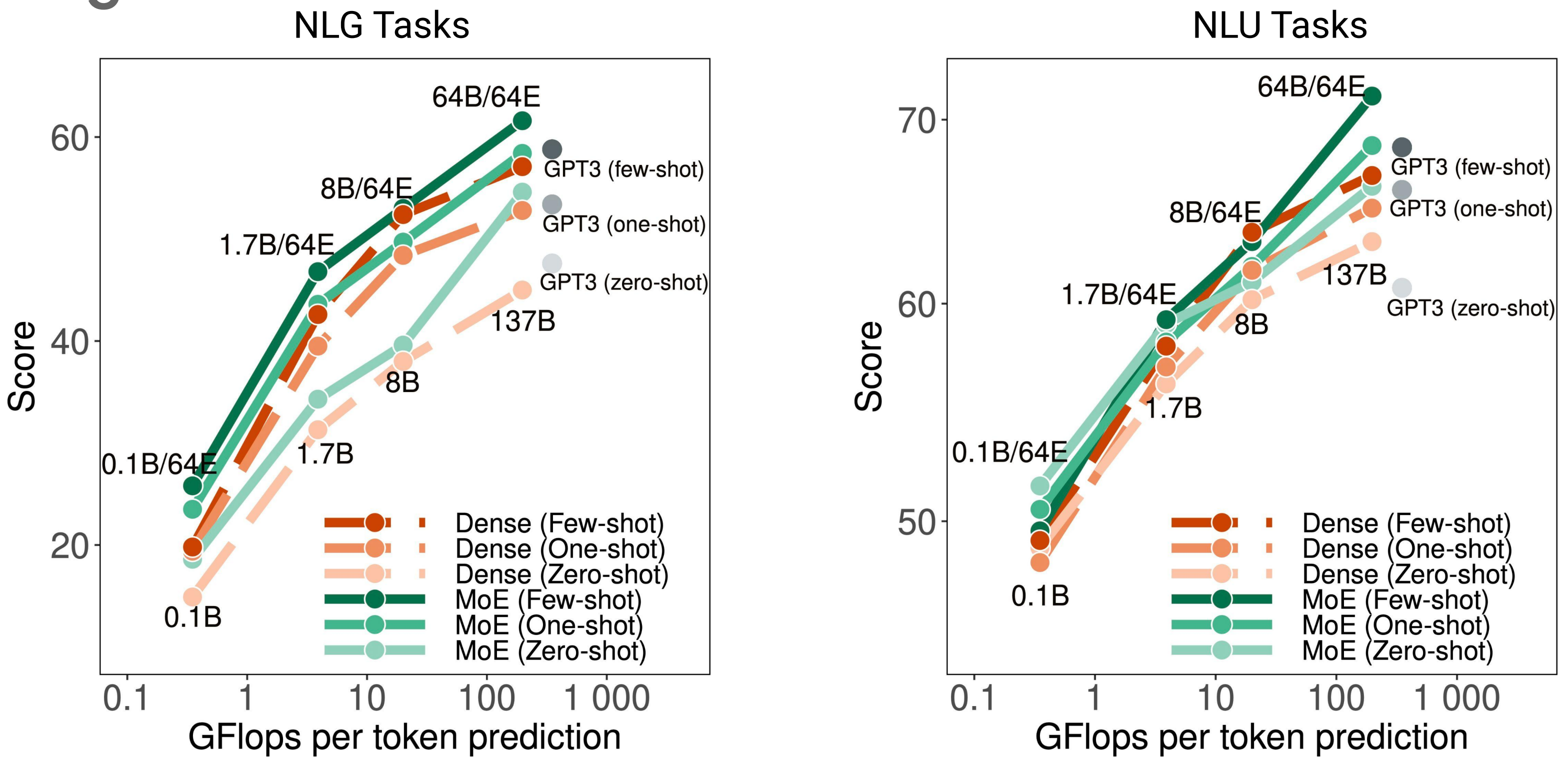
GLaM (64B/64E) has better performance while using ⅓ of the energy and ½ of serving cost of GPT-3.

		GPT-3	**GLaM**	relative
cost	FLOPs / token (G)	350	**180**	–48.6%
	Train energy (MWh)	1287	**456**	–64.6%
accuracy on average	Zero-shot	56.9	**62.7**	+10.2%
	One-shot	61.6	**65.5**	+6.3%
	Few-shot	65.2	**68.1**	+4.4%

Performance Changes by Categories (vs GPT-3)



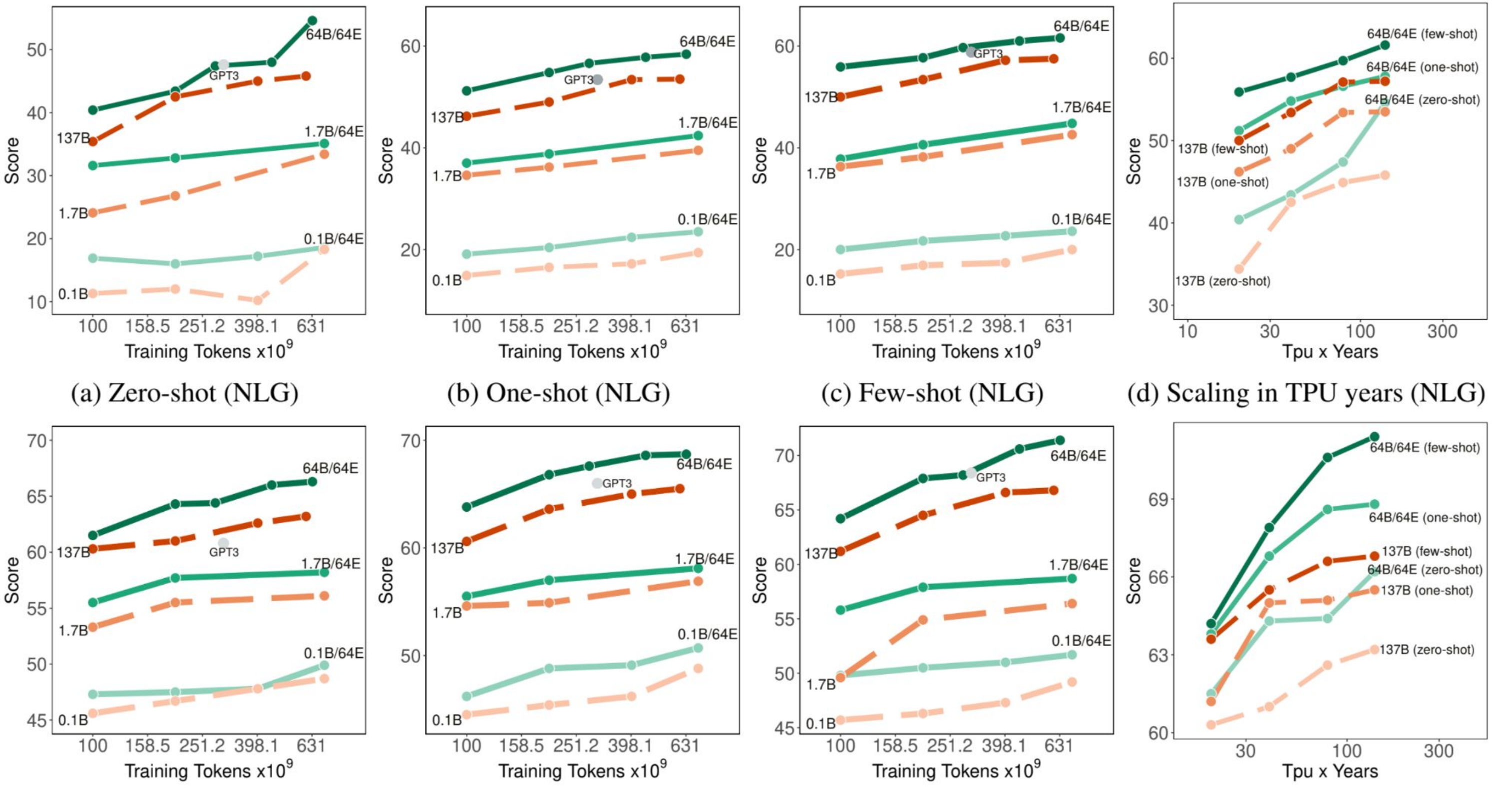
Scaling



Using **similar FLOPs** per token prediction, MoE models have better performance than the dense variants.

Learning Efficiency

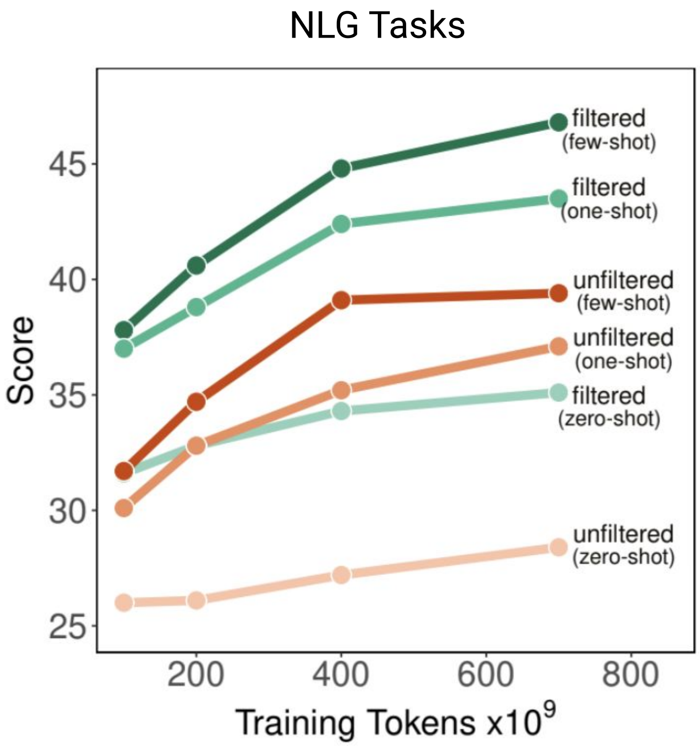
Training with **the same number** of tokens (or TPU time), MoE models have better performance than the dense variants.

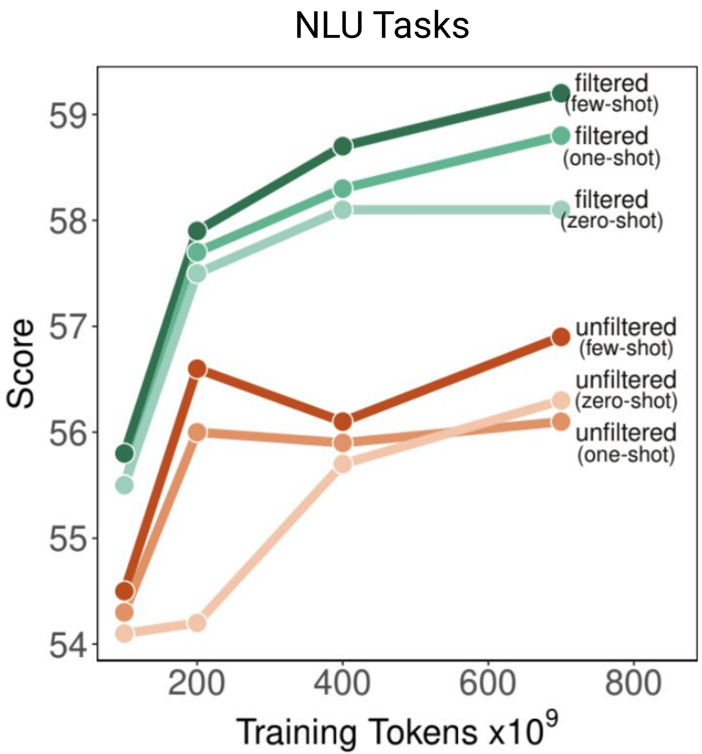


(a) Zero-shot (NLG) (b) One-shot (NLG) (c) Few-shot (NLG) (d) Scaling in TPU years (NLG)

(e) Zero-shot (NLU) (f) One-shot (NLU) (g) Few-shot (NLU) (h) Scaling in TPU years (NLU)

Effects of Data Filtering





High quality data is crucial for general purpose of pre-training even though the raw data can be massive.

Google

Takeaways

- By developing a family of dense and MoE based autoregressive language models, we have shown
  - MoE models have **better predictive performance** when using similar number of FLOPs per token.
  - MoE models have **better learning efficiency** when training with the same number of tokens.
- Given the fast development of more powerful language models, we advocate
  - More research into methods for obtaining high-quality data.
  - Considering using MoE for more efficient scaling.