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

Nan Du [*1] **Yanping Huang** [*1] **Andrew M. Dai** [*1] **Simon Tong** [1] **Dmitry Lepikhin** [1] **Yuanzhong Xu** [1] **Maxim Krikun** [1] **Yanqi Zhou** [1] **Adams Wei Yu** [1] **Orhan Firat** [1] **Barret Zoph** [1] **Liam Fedus** [1] **Maarten Bosma** [1] **Zongwei Zhou** [1] **Tao Wang** [1] **Yu Emma Wang** [1] **Kellie Webster** [1] **Marie Pellat** [1] **Kevin Robinson** [1] **Kathleen Meier-Hellstern** [1] **Toju Duke** [1] **Lucas Dixon** [1] **Kun Zhang** [1] **Quoc V Le** [1] **Yonghui Wu** [1] **Zhifeng Chen** [1] **Claire Cui** [1]

Abstract

Scaling language models with more data, compute and parameters has driven significant progress in natural language processing. For example, thanks to scaling, GPT-3 was able to achieve strong results on in-context learning tasks. However, training these large dense models requires significant amounts of computing resources. In this paper, we propose and develop a family of language models named GLaM (**G**eneralist **La**nguage **M**odel), which uses a sparsely activated mixture-of-experts architecture to scale the model capacity while also incurring substantially less training cost compared to dense variants. The largest GLaM has 1.2 trillion parameters, which is approximately 7x larger than GPT-3. It consumes only 1/3 of the energy used to train GPT-3 and requires half of the computation flops for inference, while still achieving better overall zero, one and few-shot performance across 29 NLP tasks.

1. Introduction

Language models have played an important role in the progress of natural language processing (NLP) in the past decade. Variants of language models have been used to produce pretrained word vectors (Mikolov et al., 2013; Pennington et al., 2014), and contextualized word vectors (Peters et al., 2018; Devlin et al., 2019) for many NLP applications. The shift towards scaling with more data and larger models (Shazeer et al., 2017; Huang et al., 2019; Kaplan et al., 2020) has enabled complex natural language tasks to be performed with less labeled data. For example, GPT-3 (Brown et al., 2020) and FLAN (Wei et al., 2021) demonstrated the feasibility of in-context learning for few-shot or even zero-shot generalization, meaning very few labeled examples are needed to achieve good performance on NLP applications. While being effective and performant, scaling further is becoming prohibitively expensive and consumes significant amounts of energy (Patterson et al., 2021).

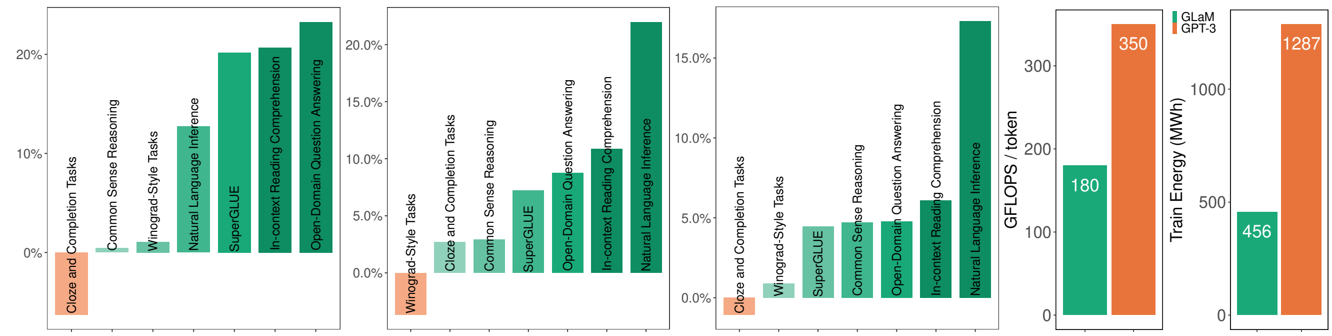
Table 1. Comparison between GPT-3 and GLaM. In a nutshell, GLaM outperforms GPT-3 across 21 natural language understanding (NLU) benchmarks and 8 natural language generative (NLG) benchmarks in average while using about half the FLOPs per token during inference and consuming about one third the energy for training.

		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%

In this work, we show that a large sparsely activated network can achieve competitive results compared to state-of-the-art dense models on few-shot tasks while being more computationally efficient. We present a family of generalist language models called GLaM, that strike a balance between dense and conditional computation. The largest version of GLaM has 1.2T parameters in total with 64 experts per MoE layer (Shazeer et al., 2017; Lepikhin et al., 2021; Fedus et al., 2021) where each token in the input batch only activates a subnetwork of 96.6B (8% of 1.2T) parameters. On zero, one and few-shot learning, this model compares favorably to GPT-3 (175B), with significantly improved learning efficiency across 29 public NLP benchmarks, ranging from language completion tasks, open-domain QA tasks, to natural language inference tasks. Thanks to the sparsely activated architecture and the efficient implementation of the model parallelism algorithm, the total energy consumption during training is only one third of GPT-3's. We highlight the comparison between the largest version of GLaM and GPT-3 in Table 1 and Figure 1.

[*]Equal contribution [1]Google. Correspondence to: Nan Du, Yanping Huang, and Andrew M. Dai <dunan@google.com, huangyp@google.com, adai@google.com>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.



(a) Zero-shot (b) One-shot (c) Few-shot (d) Train and inference cost

Figure 1. An overview of the percentage change in predictive performance (higher is better) of GLaM (64B/64E) versus GPT-3 (175B) in the (a) zero-shot, (b) one-shot, and (c) few-shot setting across 7 benchmark categories with 29 public tasks in total. Each bar in panel (a), (b) and (c) represents one benchmark category. Panel (d) compares the FLOPs needed per token prediction and training energy consumption.

We use GLaM to study the importance of data. Our analysis shows that even for these large models, data quality should not be sacrificed for quantity if the goal is to produce a high-quality auto-regressive language model. More importantly, on social dimensions, our results are also the first, to our knowledge, to close the performance gap between stereotypical and anti-stereotypical examples on the WinoGender benchmark, suggesting that large, sparsely activated models may rely less on superficial statistical correlations.

Finally, although MoE-based sparse models are not yet common in the NLP community, our work shows that sparse decoder-only language models can be more performant than the dense architectures of similar compute FLOPs for the first time within the few-shot in-context learning setting at scale, suggesting that sparsity is one of the most promising directions to achieve high-quality NLP models while saving energy costs (Patterson et al., 2021). MoE should therefore be considered as a strong candidate for future scaling.

2. Related Work

Language models. Neural language models (Mikolov et al., 2010; Sutskever et al., 2011) have been shown to be useful for many natural language processing tasks. Word embedding models and extensions such as word2vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014) and paragraph vectors (Le & Mikolov, 2014) have shown good generalization to many tasks simply by transferring the embeddings.

Pre-training and Fine-tuning. The abundance of compute and data enables training increasingly large models via unsupervised pre-training. This is a natural fit for training neural networks as they exhibit remarkable scalability. Work on using recurrent models such as RNNs and LSTMs for language representation (Dai & Le, 2015; Kiros et al., 2015) showed that general language models could be fine-tuned to improve various language understanding tasks. More recently, models that used Transformers (Vaswani et al., 2017) showed that larger models with self-supervision on unlabeled data could yield significant improvements on NLP tasks (Devlin et al., 2019; Yang et al., 2019; Liu et al., 2019; Clark et al., 2020). Transfer learning based on pre-training and finetuning (Raffel et al., 2020; Houlsby et al., 2019) has been extensively studied and demonstrated good performance on downstream tasks. However, a major limitation to this method is that it requires a task-specific fine-tuning.

In-Context Few-shot Learning. GPT-3 (Brown et al., 2020) and related work (Shoeybi et al., 2019; Lieber et al., 2021; Wei et al., 2021) demonstrated that scaling up language models greatly improves task-agnostic, few-shot performance. These language models are applied without any gradient updates, and only few-shot demonstrations specified purely via text interactions with the model are needed.

Sparsely Gated Networks. Mixture-of-Experts based models have also shown significant advantages. For language modeling and machine translation, Shazeer et al. (2017) showed that they could effectively use a very large number of weights while only needing to compute a small subset of the computation graph at inference time. There has also been work on scaling sparsely activated MoE architectures (Hestness et al., 2017; Shazeer et al., 2018; Lepikhin et al., 2021; Kudugunta et al., 2021). Recently, Fedus et al. (2021) showed results with even larger 1 trillion parameter sparsely activated models (Switch-C). Although both Switch-C and the largest GLaM model have one trillion number of trainable parameters, GLaM is a family of decoder-only language models, and Switch-C is an encoder-decoder based sequence to sequence model. Furthermore, Switch-C is mainly evaluated on fine-tuning benchmarks, *e.g.*, SuperGlue, while GLaM performs well without any

Table 2. A sample of related models (Devlin et al., 2019; Raffel et al., 2020; Brown et al., 2020; Lieber et al., 2021; Rae et al., 2021; Shoeybi et al., 2019; Lepikhin et al., 2021; Fedus et al., 2021) pre-trained on text corpora. n_{params} is the total number of trainable model parameters, $n_{\text{act-params}}$ is the number of activated model parameters per input token.

Model Name	Model Type	n_{params}	$n_{\text{act-params}}$
BERT	Dense Encoder-only	340M	340M
T5	Dense Encoder-decoder	13B	13B
GPT-3	Dense Decoder-only	175B	175B
Jurassic-1	Dense Decoder-only	178B	178B
Gopher	Dense Decoder-only	280B	280B
Megatron-530B	Dense Decoder-only	530B	530B
GShard-M4	MoE Encoder-decoder	600B	1.5B
Switch-C	MoE Encoder-decoder	1.5T	1.5B
GLaM (64B/64E)	MoE Decoder-only	1.2T	96.6B

need for fine-tuning in the few-shot setting shared by GPT-3 where SuperGlue is a subset. Table 2 summarizes the key differences between GLaM and related models pre-trained on text corpora.

3. Training Dataset

To train our model, we build a high-quality dataset of 1.6 trillion tokens that are representative of a wide range of natural language use cases. Web pages constitute the vast quantity of data in our unlabeled dataset. However, their quality ranges from professional writing to low-quality comment and forum pages. Similarly to Brown et al. (2020), we develop our own text quality classifier to produce a high-quality web corpus out of an original larger raw corpus. We use a feature hash based linear classifier for inference speed. This classifier is trained to classify between a collection of curated text (Wikipedia, books and a few selected websites) and other webpages. We use this classifier to estimate the content quality of a webpage. We then apply this classifier by using a Pareto distribution to sample webpages according to their score. This allows some lower-quality webpages to be included to prevent systematic biases in the classifier (Brown et al., 2020).

Table 3. Data and mixture weights in GLaM training set.

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

We use this process to generate a high-quality filtered subset of webpages and combine this with books, Wikipedia pages, forums and news pages and other data sources to create the final GLaM dataset. We also incorporate the data from public domain social media conversations used by Adiwardana et al. (2020). We set the mixture weights based on the performance of each component in a smaller model and to prevent small sources such as Wikipedia from being over-sampled. Table 3 shows the details of our data component sizes and mixture weights. The mixture weights were chosen based on the performance of the component in a small model and to prevent small datasets such as Wikipedia from being over-sampled. To check data contamination, in Section D we conduct an overlap analysis between our training set and the evaluation data and find that it roughly matches that of previous work (Brown et al., 2020).

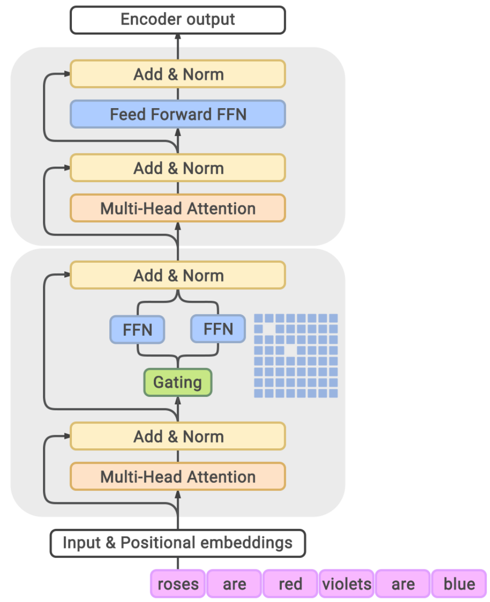


Figure 2. GLaM model architecture. Each MoE layer (the bottom block) is interleaved with a Transformer layer (the upper block). For each input token, *e.g.*, 'roses', the *Gating* module dynamically selects two most relevant experts out of 64, which is represented by the blue grid in the MoE layer. The weighted average of the outputs from these two experts will then be passed to the upper Transformer layer. For the next token in the input sequence, two different experts will be selected.

4. Model Architecture

We leverage sparsely activated Mixture-of-Experts (MoE) (Shazeer et al., 2017; Fedus et al., 2021) in GLaM models. Similar to the GShard MoE Transformer (Lepikhin et al., 2021), we replace the feed-forward component of every other Transformer layer with an MoE layer, as shown in Figure 2. Each MoE layer consists of a collection of independent feed-forward networks as the 'experts'. A gating function then uses a softmax activation function to model a probability distribution over these experts. This distribution indicates how well each expert is able to process the incoming input.

Even though each MoE layer has many more parameters, the experts are sparsely activated. This means that for a given input token, only a limited subset of experts is used, giving the model more capacity while limiting computation. In our architecture, the subset size is two[1]. Each MoE layer's learnable gating network is trained to use its input to activate the best two 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, this essentially provides a collection of $O(E^2)$ different combinations of feed-forward networks instead of one in the classic Transformer architecture, leading to much more computational flexibility. The final learned representation of a token will be the weighted combination of the outputs from the selected experts.

We also make additional modifications to the original Transformer architecture. We replace the standard positional embedding with per-layer relative positional bias from Dai et al. (2019). In the non-MoE Transformer feed-forward sub-layers, we replace the first linear projection and the activation function with the Gated Linear Unit (Dauphin et al., 2017; Shazeer, 2020), which computes the component-wise product of two linear transformation of the input, followed by a Gaussian Error Linear Unit (Hendrycks & Gimpel, 2016) activation function. We partition the weights and computation of large GLaM models using the 2D sharding algorithm as described in Xu et al. (2021), which is described in more details in the Section C of the appendix.

5. Experiment Setup

GLaM is a family of dense and sparse decoder-only language models, so we first elaborate our training settings, hyperparameters, and evaluation protocol in this section.

5.1. Training Setting

We train several variants of GLaM to study the behavior of MoE and dense models on the same training data. Table 4 shows the hyperparameter settings of different scale GLaM models ranging from 130 million parameters to 1.2 trillion parameters. Here, E is the number of experts in the MoE layer, B is the mini-batch size, S is the input sequence length, M is the model and embedding dimension, H is the hidden dimension of the feed-forward network, L is the number of layers and N is the number of total devices. Additionally, n_{params} is the total number of trainable model parameters, $n_{\text{act-params}}$ is the number of **activated** model parameters per input token, n_{heads} is the number of self-attention heads, and d_{head} is the hidden dimension of each attention head. We also include the respective dense models with comparable numbers of activated parameters per-token during inference (and thus similar numbers of per-token FLOPs) as references. We adopt the notation of

$$\text{GLaM (Base Dense Size}/E) \quad e.g., \ \text{GLaM (8B}/\text{64E)}$$

to describe different variants in the GLaM models. For example, GLaM (8B/64E) represents the architecture of an approximate 8B parameter dense model with every other layer replaced by a 64 expert MoE layer. GLaM reduces to a dense Transformer-based language model architecture when each MoE layer only has one expert. We use the notation

$$\text{GLaM (Dense Size)} \quad e.g., \ \text{GLaM (137B)}$$

refers to a dense 137B parameter model trained with the same dataset.

5.2. Hyperparameters and Training Procedure

We use the same learning hyperparameters for all GLaM models. More specifically, We use a maximum sequence length of 1024 tokens, and pack each input example to have up to 1 million tokens per batch. The dropout rate is set to 0 since the number of available tokens in the training corpus is much greater than the number of processed tokens during training. Our optimizer is Adafactor (Shazeer & Stern, 2018) with first-moment decay $\beta_1 = 0$, second-moment decay $\beta_2 = 0.99$ with a $1 - t^{-0.8}$ decay schedule, update clipping threshold of 1.0, and factored second-moment estimation. We keep the initial learning rate of 0.01 for the first 10K training steps, and then decay it with inverse square root schedule $\text{lr}\langle \text{t} \rangle \propto \frac{1}{\sqrt{\text{t}}}$. On top of the standard cross-entropy loss, we add the MoE auxiliary loss as described in GShard (Lepikhin et al., 2021) with a 0.01 coefficient to encourage expert load balancing so that the gating function will distribute tokens more evenly across all experts. We use the SentencePiece (Kudo & Richardson, 2018) subword tokenizer with a vocabulary of size of 256K. During training, we use *float32* for model weights and *bfloat16* for activations. The largest GLaM 64B/64E model was trained on 1,024 Cloud TPU-V4 chips.

Training models at the trillion parameter scale is extremely expensive even for sparsely activated models. There is little room for hyperparameter tuning. Here we share our training recipes and some implementation tricks for the GLaM models.

[1]Using more experts will cost more compute FLOPs per prediction, pushing the network to be 'denser'. Setting the number of selected experts to be two is based on the trade-off between predictive performance and the training/serving efficiency of the model.

Table 4. Sizes and architectures of both MoE and dense models that we have trained in our experiments. Models are grouped by the number of activated parameters per token. All trained models share the same learning hyperparameters described in Session 5.1.

GLaM Model	Type	n_{params}	$n_{\text{act-params}}$	L	M	H	n_{heads}	d_{head}	E
0.1B	Dense	130M	130M	12	768	3,072	12	64	–
0.1B/64E	MoE	1.9B	145M						64
1.7B	Dense	1.7B	1.700B	24	2,048	8,192	16	128	–
1.7B/32E	MoE	20B	1.878B						32
1.7B/64E	MoE	27B	1.879B						64
1.7B/128E	MoE	53B	1.881B						128
1.7B/256E	MoE	105B	1.886B						256
8B	Dense	8.7B	8.7B	32	4,096	16,384	32	128	–
8B/64E	MoE	143B	9.8B						64
137B	Dense	137B	137B	64	8,192	65,536	128	128	–
64B/64E	MoE	1.2T	96.6B	64	8,192	32,768	128	128	64

- We train smaller-scale models to convergence first. This allows us to expose potential issues in the dataset and infrastructure as early as possible.
- We skip weight updates for a batch if there are any *NaN*s or *Inf*s in the gradients (Shen et al., 2019). Note *NaN/Inf* could still occur during the applying gradient step, in which case we restart from an earlier checkpoint as described below. For example, even if there is no *Inf* in the existing variable or the gradient, the updated variable could still lead to *Inf*.
- We restart from an early healthy checkpoint when encountering rare large fluctuations or even *NaN/Inf* during training. Randomness of the sequentially loaded batches might help escape from previous failed states in the training after restart.

5.3. Evaluation Setting

Protocol. To clearly demonstrate the effectiveness of GLaM models, we mainly focus on evaluating the zero, one and few-shot learning protocols suggested by Radford et al. (2018); Brown et al. (2020). For the zero-shot learning setting, in most cases, we evaluate each example in the development set directly. For one/few-shot learning, we mainly draw random one/few examples from that task's training set as the only demonstration and context. Such a demonstration is concatenated with the evaluation example with two newlines in between, and then fed into the model.

Benchmarks. To allow for an apples-to-apples comparison between GPT-3 and GLaM, we choose the same suite of evaluation tasks as Brown et al. (2020). But for simplicity, we exclude 7 synthetic tasks (arithmetic and word unscramble) and 6 machine translation datasets. With this exclusion, we end up with 29 datasets, which includes 8 natural language generative (NLG) tasks and 21 natural language understanding (NLU) tasks. These datasets can be further grouped into 7 categories and are listed in section A.

Natural Language Generative tasks. We compare the language sequences decoded by the models to the ground truth in generative tasks. These tasks are TriviaQA, NQS, WebQS, SQuADv2, LAMBADA, DROP, QuAC and CoQA. The performance is measured by the accuracy of exact match (EM) and F1 score, following the standard for each task in Brown et al. (2020). We use beam search with a width of 4 to generate the sequences.

Natural Language Understanding tasks. Most language understanding tasks require the model to select one correct answer from multiple options. All binary classification tasks are formulated into the form of selecting among two options ('Yes' or 'No'). The prediction is based on the maximum log-likelihood of each option given the context $\log P(\text{option}|\text{context})$ normalized by the token length of each option. On a few tasks, such as ReCoRD (Zhang et al., 2018) and COPA (Gordon et al., 2012), the non-normalized loss can yield better results and thus is adopted. Except for MultiRC (Khashabi et al., 2018) where the F1 metric over the set of answer options (referred to as F1_a) is reported, the prediction accuracy metric is used for all the other tasks. We use the average of the scores reported in all datasets to report the overall few-shot performance of models on both NLG and NLU tasks. Both Accuracy (EM) and F1 scores have been normalized to lie between 0 and 100. On TriviaQA, we also report the testing server score of our one-shot submission.

6. Results

We conduct extensive evaluation on the whole family of GLaM models, to show the advantages of sparsely activated models in language modeling and their scaling trends. We also quantitatively inspect the effectiveness of data quality for language model training.

6.1. Comparison between MoE and Dense Models

As previously presented in Table 1, GLaM (64B/64E) has competitive performance compared to GPT-3 (175B) for zero, one and few-shot learning. Figure 1 compares the performance for each category of tasks. In total, GLaM (64B/64E) outperforms GPT-3 in 6 out of 7 categories on average, indicating the performance gain is consistent. For more details on each individual task, see Table 11. We include results on the much larger and computationally demanding Megatron-NLG and Gopher for reference. More importantly, as shown in Table 4, GLaM (64B/64E) activates roughly 96.6B parameters per token during inference, which requires only half of the compute FLOPs needed by GPT-3 given the same input.

We highlight one particular challenging open-domain question answer task: *TriviaQA*. In open-domain question answer tasks, the model is required to directly answer a given query without access to any additional context. Brown et al. (2020) show that the few-shot performance of TriviaQA is able to grow smoothly with model size, indicating a language model is able to absorb knowledge using its model capacity. As shown in Table 5, GLaM (64B/64E) is better than the dense model and outperforms the previous finetuned state-of-the-art (SOTA) on this dataset in the open-domain setting. Our one-shot result exceeds the previous finetuned SOTA (Yu et al., 2022) where additional knowledge graph information is infused by 8.6%, and outperforms the few-shot GPT-3 on the testing server by 5.3%. This suggests that the additional capacity of GLaM plays a crucial role in the performance gain even though the $n_{\text{act-params}}$ of GLaM (64B/64E) is only half of that in GPT-3. Comparing to Switch-C, even though both models have similar total number of parameters, GLaM (64B/64E) uses much larger experts (beyond one TPU core) than Switch-C. Therefore, GLaM's one-shot performance on TriviaQA is also better than the fine-tuned results of Switch-C in the open-domain setting. Finally, we report zero, one and few-shot evaluation mainly on the development set for all tasks in Tables 11, 12, 13 and 14 of the appendix.

6.2. Effect of Data Quality

We study the impact of data quality on the few-shot performance of downstream tasks. We use a modest-size GLaM model (1.7B/64E) to show the effectiveness of filtering text on model quality. We train models with the same hyperparameters on two datasets. One is the original dataset described in Section 3 and the second consists of the dataset with the filtered webpages replaced with the unfiltered webpages. The mixing proportions are fixed as given in Table 3.

Table 5. GLaM (64B/64E) one-shot performance significantly outperforms prior SOTAs for open domain settings in the wiki split.

Model	TriviaQA (Open-Domain)
KG-FiD (large) (Yu et al., 2022) (finetuned, test)	69.8
Switch-C (finetuned, dev)	47.5
GPT-3 One-shot (dev)	68.0
GPT-3 64-shot (test)	71.2
GLaM One-shot (test)	75.0
GLaM One-shot (dev)	**75.8**

The filtered webpages consist of 143B tokens whereas the unfiltered webpages consist of around 7T tokens.

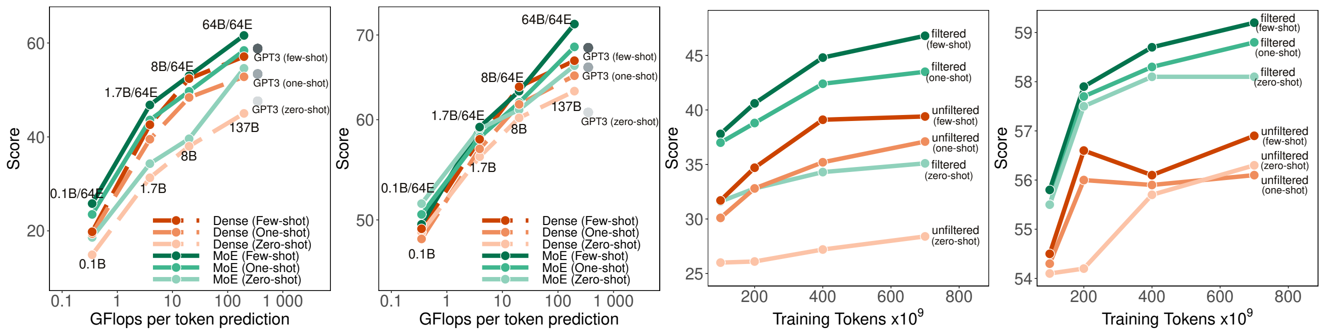
Figure 3 (c) and (d) show that the model trained on filtered data performs consistently better on both NLG and NLU tasks. In particular, the effect of filtering is bigger on NLG than that on NLU. Perhaps this is because NLG often requires generating high-quality language and filtered pretraining corpora is crucial to the generation capability of language models. Our study highlights the fact that the quality of the pretrained data also plays a critical role, specifically, in the performance of downstream tasks.

6.3. Scaling Studies

Scaling up dense language models generally involves making the models deeper by adding more layers, and wider by increasing the embedding dimension of token representations. This process increases the total number of parameters n_{params} of the model. For each prediction on a given input example, these models are 'dense' in that all n_{params} parameters will be activated, i.e., $n_{\text{params}} = n_{\text{act-params}}$ in Table 4. Therefore, the effective FLOPs per prediction increases linearly with the model size n_{params}. While the increased FLOPs may lead to boosted predictive performance, it also raises the overall cost per prediction.

In contrast, GLaM MoE models are **sparsely activated** in that only a small fraction of the total n_{params} parameters will be activated for each prediction where $n_{\text{params}} \gg n_{\text{act-params}}$. Therefore, GLaM MoE models can scale by also growing the size or number of experts in the MoE layer.

As shown in Figure 3(a), the average zero, one and few-shot performance across the generative tasks scales well with the effective FLOPs per prediction which is in turn determined by $n_{\text{act-params}}$. We also find that GLaM MoE models perform consistently better than GLaM dense models for similar effective FLOPs per token. For language understanding tasks shown in Figure 3(b), the performance gain of GLaM MoE models has a similar scaling trend to that of the generative tasks. We observe that both MoE and dense models perform similarly at smaller scales but MoE models outperform at



(a) Scaling (NLG) (b) Scaling (NLU) (c) Data filtering (NLG) (d) Data filtering (NLU)

Figure 3. Average zero, one and few-shot performance of GLaM MoE models versus GLaM dense models for similar effective FLOPs per token over the 8 NLG tasks (a) and 21 NLU tasks (b). Comparison of model performance with filtered and unfiltered training data using GLaM (1.7B/64E). Filtered data improves results significantly over unfiltered data for both (c) NLG and (d) NLU tasks across zero, one and few-shot settings.

larger scales. We also show experiments with scaling the number of experts in Section B where we observe that, for a fixed budget of computation per prediction, adding more experts generally leads to better predictive performance.

6.4. Efficiency of GLaM

Existing large dense language models usually require tremendous amounts of computation resources for training and serving (Patterson et al., 2021). They also need to consume massive amounts of pretraining data. We investigate the data and compute efficiency of the proposed GLaM models.

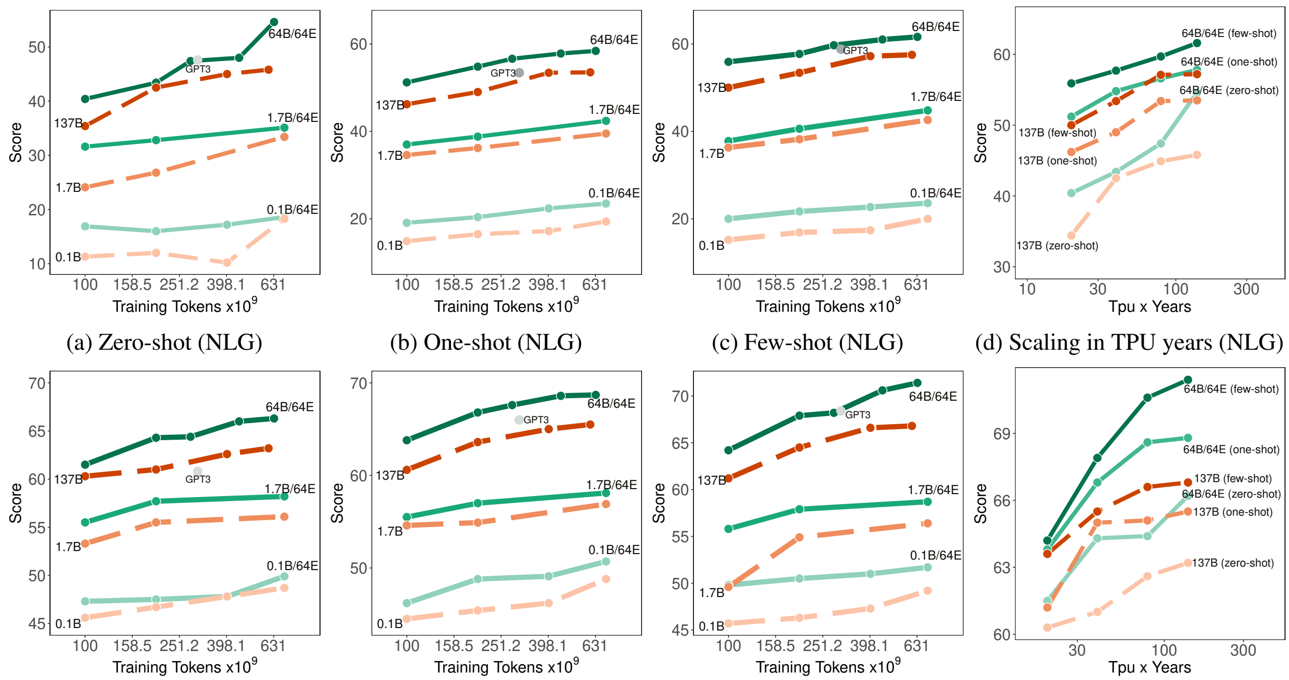
Data Efficiency. Figure 4 (a-c) and Figure 4(e-g) show the learning curves of our models compared to the dense baselines of similar effective FLOPs in both NLG and NLU tasks. The x-axis is the number of tokens used in training where we explicitly include GPT-3's results when it is around 300B tokens. We first observe that GLaM MoE models require significantly less data than dense models of comparable FLOPs to achieve similar zero, one, and few-shot performance. In other words, when the same amount of data is used for training, MoE models perform much better, and the difference in performance becomes larger when training up to 630B. Moreover, GLaM (64B/64E) model trained with 280B tokens outperforms GPT-3 trained with 300B tokens by large margins on 4 out of the 6 learning settings (zero-shot/one-shot NLU and one-shot/few-shot NLG), and matches GPT-3 scores for the remaining setting, i.e., zero-shot NLG tasks.

Computation Efficiency & Energy Consumption. Figure 4 (d) and Figure 4 (h) show how the average zero, one and few-shot performance scales with the number of TPU years spent training MoE and dense models. We find that to achieve similar performance on downstream tasks, training sparsely activated models takes much less computational resources than training dense models.

As previously presented in Table 1, the GLaM (64B/64E) training after 600B tokens consumes 456 MWh, about 1/3 of the energy cost of 1287 MWh used by GPT-3. Moreover, to reach similar (and slightly exceeded) scores as GPT-3, we train using 1,024 TPU-v4 chips for 574 hours (with 280B tokens). This consumes 213 MWh or 1/6 of the GPT-3 energy cost. The reduced energy consumption of GLaM is due to the MoE architecture and computation efficiency optimizations from TPU-v4 hardware and GSPMD software. Energy calculations can be found in Section F.

7. Ethics and Unintended Biases

Large language models' zero-and few-shot inference is an exciting capability: being able to control model behaviour intuitively with natural language and small datasets significantly lowers the barrier to prototyping and the development of new applications; it has the potential to help democratise using AI by dramatically decreasing the need for specialist knowledge. However, such opportunities also serve to highlight the importance of the many ethical challenges (Leidner & Plachouras, 2017; Bender et al., 2021; Bommasani et al., 2021) including representation bias (Blodgett et al., 2020), proper selection and handling of training data (Rogers, 2021) and its documentation (Bender & Friedman, 2018), privacy (Abadi et al., 2016b; Carlini et al., 2020), and environmental concerns (Strubell et al., 2019; Patterson et al., 2021). An important strand of this research focuses on unintended biases learnt by language models, including correlations between gender and profession (Bolukbasi et al., 2016; Rudinger et al., 2018; Zhao et al., 2018), negative sentiment about racial and religious groups (Li et al., 2020; Nadeem et al., 2021), and about people with disabilities (Hutchinson et al., 2020), as well as other social biases



(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)

Figure 4. Learning efficiency comparison. Average zero-shot , one-shot and few-shot performance of GLaM MoE models versus GLaM dense models as more tokens are processed during training for 9 NLG tasks (a-c) and 21 NLU tasks (e-g). Panel (d) and (h) also display the learning curves against the number of TPU years, respectively.

(Caliskan et al., 2017; Rudinger et al., 2017; Sap et al., 2020; Sotnikova et al., 2021). While measuring and mitigating the potential harm of language models is a very active area of research, as recognized by Blodgett et al. (2021); Jacobs & Wallach (2021) there is still a significant need for more rigorous evaluation methods to assess the degree to which language models encode harmful stereotypes (May et al., 2019; Webster et al., 2021).

While there is not yet consensus on measurement methods or criteria for such general purpose large language models, the versatility and power of these models make it important to assess them on a range of metrics. We take inspiration from GPT-3 (Brown et al., 2020) and examine the co-occurrence in generated text referencing identity terms as well as report on the WinoGender benchmark (Rudinger et al., 2018). We also analyse toxicity degeneration similarly to Gopher (Rae et al., 2021), and extend the analysis to consider the human-behavioral baseline.

7.1. Co-occurrence prompts

Following the procedure described in Brown et al. (2020), we analyze commonly co-occurring words in the continuations when given prompts like "{term} was very..." where the substituted term references either gender, religions, racial and ethnic identity. For each prompt (Table 7 of the appendix), 800 outputs are generated using top-k sampling ($k = 40$) with a temperature of 1. An off-the-shelf POS tagger (Bird & Loper, 2004) is used to remove stop words and select only descriptive words (i.e., adjectives and adverbs). Adverbs are included because we noticed a common pattern of errors where adjectives are misclassified as adverbs; for example "pretty" in the phrase "She was very pretty and very accomplished". Like Brown et al. (2020), to make the analysis transparent and easily reproducible, we omit any manual human labeling.

Like the analysis of other large language models that we build on, we note associative biases for all dimensions are obvious, for example "pretty" is the most associated description for the term "She", while it is not in the top-10 for the term "He". Table 8 shows the most frequently occurring descriptive words in response to prompt-templates for gendered pronouns, and Tables 9 and 10 of the appendix show the same for race and religion prompts.

7.2. WinoGender

Coreference resolution is a capability that many applications require to perform well, including machine translation (Stanovsky et al., 2019; Webster & Pitler, 2020) and question answering (Lamm et al., 2020). To assess whether gendered correlations in GLaM cause it to make corefer-

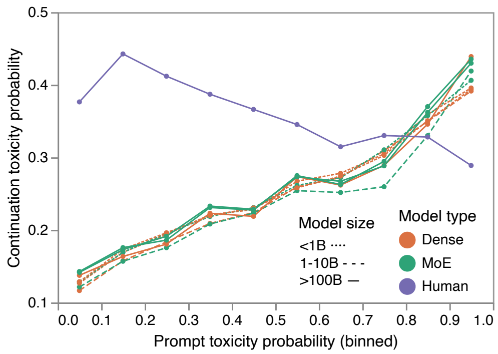


Figure 5. The relationship between the Toxicity Probability of the Prompt (TPP), and the Toxicity Probability of the Continuation (TPC). Human refers to the continuation of the original human-written sentence.

ence errors in the one-shot setting, we measure WinoGender (Rudinger et al., 2018). GLaM (64B/64E) achieves a new state-of-the-art of 71.7% on the full dataset (compared to 64.2% for GPT-3 (Brown et al., 2020)). Promisingly, accuracy is remarkably close between 'he' examples (70.8%) and 'she' examples (72.5%), as well as between stereotypical examples (where the intended distribution is assumed to be close to the US occupation statistics, (Rudinger et al., 2018)) and anti-stereotypical (or 'gotcha') examples (both 71.7%).

7.3. Toxicity Degeneration

Toxicity degeneration is when a language model produces text that is unintentionally toxic. To evaluate toxicity degeneration, we adapt the methodology used in (Welbl et al., 2021; Rae et al., 2021). We use the RealToxicityPrompts dataset (Gehman et al., 2020) which consists of sentences that have been split into two parts: a *prompt* prefix, and a *continuation* postfix. Like the previous studies, we also use the Perspective API which assigns a probability that the text would be considered to be rude, disrespectful or otherwise likely to make people want to leave a conversation. We then asses how likely a continuation is to be toxic given various likelihoods that the prompt was toxic.

For each of 10K randomly sampled prompts, we generate 25 continuations, with up to 100 tokens per continuations using top-k sampling ($k = 40$) with a temperature of 1. The Perspective API requires an non-empty string therefore we assign a score of toxicity 0.0 when the continuation is the empty string; this could represent, for example, a chat bot simply refusing to respond.

Figure 5 shows the relationship between the Toxicity Probability of the Prompt (TPP), and the Toxicity Probability of the Continuation (TPC). Note that, for low TPP, the relatively high human TPC is due to the sampling strategy used to create the underlying dataset: sentences were selected across the toxicity spectrum. Moreover, toxicity can often be identified locally within a sentence, and toxicity in this dataset tends to occur later the sentences. This causes the human-TPC to slightly drop as the TPP increases. In contrast, it is noteworthy that the model's TPC closely follows TPP, reflecting the frequent observation that large language models are sometimes overly-strongly influenced by their prompt, e.g. repeating phrases from the prompt.

We also analysed the distribution of toxicity probabilities from the API for batches of 25 continuations. This highlighted that, even for low toxicity prompts, it is very likely that some generated continuation will be judged as toxic by most people reviewing it, according to the Perspective API's predicted probability; further details can be found in Figure 8. We also note that this dataset's sampling strategy, and the source it is taken from (Reddit) are likely not reflective of other domains. Moreover, even for very low TPP, applications are likely to want a much lower TPC: even generating 1 in 100 toxic suggestions is likely to be very problematic for applications.

8. Discussion

As observed in previous work on sparsely-activated models (Fedus et al., 2021), MoE models are more performant in knowledge-oriented tasks. Open-domain tasks are one way of measuring the amount of knowledge stored in a model. The performance of the MoE model in open-domain QA benchmarks such as TriviaQA demonstrate the significantly increased information capacity of these models compared to dense models of similar effective FLOPs. Despite the in-context learning and training efficiency advantages, the sparsely activated models consist of a higher number of parameters and thus require a larger number of devices. This limits the resource accessibility and increases the serving cost especially when the serving traffic is low.

9. Conclusions

We propose and develop a family of generalist language models called GLaM, which use a sparsely activated mixture-of-experts architecture to achieve better average scores than not only their dense counterparts of similar effective FLOPs, but also the GPT-3 models on 29 representative NLP tasks in zero, one and few-shot learning. In particular, GLaM (64B/64E), our largest 1.2 trillion parameter MoE language model, achieves better average performance with only one third of energy consumption compared to training GPT-3. We hope that our work will encourage more research into methods for obtaining high-quality data, and using MoE for more efficient scaling of giant language models.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X. TensorFlow: A system for Large-Scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pp. 265–283, Savannah, GA, November 2016a. USENIX Association. ISBN 978-1-931971-33-1. URL https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi.

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Oct 2016b. doi: 10.1145/2976749.2978318. URL http://dx.doi.org/10.1145/2976749.2978318.

Adiwardana, D., Luong, M., So, D. R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., and Le, Q. V. Towards a human-like open-domain chatbot. *CoRR*, abs/2001.09977, 2020. URL https://arxiv.org/abs/2001.09977.

Bender, E. M. and Friedman, B. Data statements for natural language processing: Toward mitigating system bias and enabling better science. *Transactions of the Association for Computational Linguistics*, 6:587–604, 2018. doi: 10.1162/tacl_a_00041. URL https://aclanthology.org/Q18-1041.

Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, pp. 610–623, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL https://doi.org/10.1145/3442188.3445922.

Berant, J., Chou, A., Frostig, R., and Liang, P. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1533–1544, Seattle, Washington, USA, October 2013. Association for Computational Linguistics. URL https://aclanthology.org/D13-1160.

Bird, S. and Loper, E. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pp. 214–217, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://aclanthology.org/P04-3031.

Bisk, Y., Zellers, R., Bras, R. L., Gao, J., and Choi, Y. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Blodgett, S. L., Barocas, S., Daumé III, H., and Wallach, H. Language (technology) is power: A critical survey of "bias" in NLP. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5454–5476, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.485. URL https://aclanthology.org/2020.acl-main.485.

Blodgett, S. L., Lopez, G., Olteanu, A., Sim, R., and Wallach, H. Stereotyping Norwegian salmon: An inventory of pitfalls in fairness benchmark datasets. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1004–1015, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.81. URL https://aclanthology.org/2021.acl-long.81.

Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/a486cd07e4ac3d270571622f4f316ec5-Paper.pdf.

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., Brynjolfsson, E., Buch, S., Card, D., Castellon, R., Chatterji, N. S., Chen, A. S., Creel, K., Davis, J. Q., Demszky, D., Donahue, C., Doumbouya, M., Durmus, E., Ermon, S., Etchemendy, J., Ethayarajh, K., Fei-Fei, L., Finn, C., Gale, T., Gillespie, L., Goel, K., Goodman, N. D., Grossman, S., Guha, N., Hashimoto, T., Henderson, P., Hewitt, J., Ho, D. E., Hong, J., Hsu, K., Huang, J., Icard, T., Jain, S., Jurafsky, D., Kalluri, P., Karamcheti, S., Keeling, G., Khani, F., Khattab, O., Koh, P. W., Krass, M. S., Krishna, R., Kuditipudi, R., and et al. On the opportunities and risks of foundation models. *CoRR*, abs/2108.07258, 2021. URL https://arxiv.org/abs/2108.07258.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Caliskan, A., Bryson, J. J., and Narayanan, A. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, Apr 2017. ISSN 1095-9203. doi: 10.1126/science.aal4230. URL http://dx.doi.org/10.1126/science.aal4230.

Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T. B., Song, D., Erlingsson, Ú., Oprea, A., and Raffel, C. Extracting training data from large language models. *CoRR*, abs/2012.07805, 2020.

Choi, E., He, H., Iyyer, M., Yatskar, M., Yih, W.-t., Choi, Y., Liang, P., and Zettlemoyer, L. QuAC: Question answering in context. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2174–2184, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1241. URL https://aclanthology.org/D18-1241.

Clark, C., Lee, K., Chang, M.-W., Kwiatkowski, T., Collins, M., and Toutanova, K. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1300. URL https://aclanthology.org/N19-1300.

Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555*, 2020.

Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

Dagan, I., Glickman, O., and Magnini, B. The pascal recognising textual entailment challenge. In Quiñonero-Candela, J., Dagan, I., Magnini, B., and d'Alché Buc, F. (eds.), *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pp. 177–190, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-33428-6.

Dai, A. M. and Le, Q. V. Semi-supervised sequence learning. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/7137debd45ae4d0ab9aa953017286b20-Paper.pdf.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J., Le, Q., and Salakhutdinov, R. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1285. URL https://aclanthology.org/P19-1285.

Dauphin, Y. N., Fan, A., Auli, M., and Grangier, D. Language modeling with gated convolutional networks. In *International conference on machine learning*, pp. 933–941. PMLR, 2017.

de Marneffe, M.-C., Simons, M., and Tonhauser, J. The commitmentbank: Investigating projection in naturally occurring discourse. *Proceedings of Sinn und Bedeutung*, 23(2):107–124, Jul. 2019. doi: 10.18148/sub/2019.v23i2.601. URL https://ojs.ub.uni-konstanz.de/sub/index.php/sub/article/view/601.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

Dua, D., Wang, Y., Dasigi, P., Stanovsky, G., Singh, S., and Gardner, M. DROP: A reading comprehension benchmark requiring discrete reasoning over paragraphs. In Burstein, J., Doran, C., and Solorio, T. (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 2368–2378. Association for Computational Linguistics, 2019. doi:

10.18653/v1/n19-1246. URL https://doi.org/10.18653/v1/n19-1246.

Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *CoRR*, abs/2101.03961, 2021. URL https://arxiv.org/abs/2101.03961.

Fyodorov, Y., Winter, Y., and Francez, N. A natural logic inference system. In *Inference in Computational Semantics*, 2000.

Gehman, S., Gururangan, S., Sap, M., Choi, Y., and Smith, N. A. Realtoxicityprompts: Evaluating neural toxic degeneration in language models, 2020.

Gordon, A., Kozareva, Z., and Roemmele, M. SemEval-2012 task 7: Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pp. 394–398, Montréal, Canada, 7-8 June 2012. Association for Computational Linguistics. URL https://aclanthology.org/S12-1052.

Hendrycks, D. and Gimpel, K. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016. URL http://arxiv.org/abs/1606.08415.

Hestness, J., Narang, S., Ardalani, N., Diamos, G. F., Jun, H., Kianinejad, H., Patwary, M. M. A., Yang, Y., and Zhou, Y. Deep learning scaling is predictable, empirically. *CoRR*, abs/1712.00409, 2017. URL http://arxiv.org/abs/1712.00409.

Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., De Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. Parameter-efficient transfer learning for NLP. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2790–2799. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/houlsby19a.html.

Huang, Y., Cheng, Y., Bapna, A., Firat, O., Chen, D., Chen, M. X., Lee, H., Ngiam, J., Le, Q. V., Wu, Y., and Chen, Z. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 103–112, 2019.

Hutchinson, B., Prabhakaran, V., Denton, E., Webster, K., Zhong, Y., and Denuyl, S. Social biases in NLP models as barriers for persons with disabilities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5491–5501, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.487. URL https://aclanthology.org/2020.acl-main.487.

Jacobs, A. Z. and Wallach, H. Measurement and fairness. *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, Mar 2021. doi: 10.1145/3442188.3445901. URL http://dx.doi.org/10.1145/3442188.3445901.

Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, July 2017. Association for Computational Linguistics.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Khashabi, D., Chaturvedi, S., Roth, M., Upadhyay, S., and Roth, D. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 252–262, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1023. URL https://aclanthology.org/N18-1023.

Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. Skip-thought vectors. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper/2015/file/f442d33fa06832082290ad8544a8da27-Paper.pdf.

Kudo, T. and Richardson, J. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP*, 2018.

Kudugunta, S., Huang, Y., Bapna, A., Krikun, M., Lepikhin, D., Luong, M.-T., and Firat, O. Beyond distillation: Task-level mixture-of-experts for efficient inference. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 3577–3599, 2021.

Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Kelcey, M., Devlin, J., Lee, K., Toutanova, K. N., Jones, L., Chang, M.-W., Dai, A., Uszkoreit, J., Le, Q., and Petrov, S. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

Lai, G., Xie, Q., Liu, H., Yang, Y., and Hovy, E. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082.

Lamm, M., Palomaki, J., Alberti, C., Andor, D., Choi, E., Soares, L. B., and Collins, M. QED: A framework and dataset for explanations in question answering. *CoRR*, abs/2009.06354, 2020. URL https://arxiv.org/abs/2009.06354.

Le, Q. and Mikolov, T. Distributed representations of sentences and documents. In *International conference on machine learning*, 2014.

Leidner, J. L. and Plachouras, V. Ethical by design: Ethics best practices for natural language processing. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pp. 30–40, Valencia, Spain, April 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-1604. URL https://aclanthology.org/W17-1604.

Lepikhin, D., Lee, H., Xu, Y., Chen, D., Firat, O., Huang, Y., Krikun, M., Shazeer, N., and Chen, Z. GShard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=qrwe7XHTmYb.

Levesque, H., Davis, E., and Morgenstern, L. The winograd schema challenge. In *13th International Conference on the Principles of Knowledge Representation and Reasoning, KR 2012*, Proceedings of the International Conference on Knowledge Representation and Reasoning, pp. 552–561. Institute of Electrical and Electronics Engineers Inc., 2012. ISBN 9781577355601. 13th International Conference on the Principles of Knowledge Representation and Reasoning, KR 2012 ; Conference date: 10-06-2012 Through 14-06-2012.

Li, T., Khashabi, D., Khot, T., Sabharwal, A., and Srikumar, V. UNQOVERing stereotyping biases via underspecified questions. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 3475–3489, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.311. URL https://aclanthology.org/2020.findings-emnlp.311.

Lieber, O., Sharir, O., Lenz, B., and Shoham, Y. Jurassic-1: Technical details and evaluation. *White Paper. AI21 Labs*, 2021.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

May, C., Wang, A., Bordia, S., Bowman, S. R., and Rudinger, R. On measuring social biases in sentence encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 622–628, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1063. URL https://aclanthology.org/N19-1063.

Mihaylov, T., Clark, P., Khot, T., and Sabharwal, A. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

Mikolov, T., Karafiát, M., Burget, L., Cernocký, J. H., and Khudanpur, S. Recurrent neural network based language model. In *INTERSPEECH*, 2010.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. In Bengio, Y. and LeCun, Y. (eds.), *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013. URL http://arxiv.org/abs/1301.3781.

Mostafazadeh, N., Chambers, N., He, X., Parikh, D., Batra, D., Vanderwende, L., Kohli, P., and Allen, J. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 839–849, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1098. URL https://aclanthology.org/N16-1098.

Nadeem, M., Bethke, A., and Reddy, S. StereoSet: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5356–5371, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.416. URL https://aclanthology.org/2021.acl-long.416.

Paperno, D., Kruszewski, G., Lazaridou, A., Pham, N. Q., Bernardi, R., Pezzelle, S., Baroni, M., Boleda, G., and Fernández, R. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1144. URL https://aclanthology.org/P16-1144.

Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M., and Dean, J. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

Pennington, J., Socher, R., and Manning, C. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://aclanthology.org/D14-1162.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

Pilehvar, M. T. and Camacho-Collados, J. Wic: 10, 000 example pairs for evaluating context-sensitive representations. *ArXiv*, abs/1808.09121, 2018.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2018. URL https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

Rae, J. W., Borgeaud, S., Cai, T., Millican, K., Hoffmann, J., Song, H. F., Aslanides, J., Henderson, S., Ring, R., Young, S., Rutherford, E., Hennigan, T., Menick, J., Cassirer, A., Powell, R., van den Driessche, G., Hendricks, L. A., Rauh, M., Huang, P., Glaese, A., Welbl, J., Dathathri, S., Huang, S., Uesato, J., Mellor, J., Higgins, I., Creswell, A., McAleese, N., Wu, A., Elsen, E., Jayakumar, S. M., Buchatskaya, E., Budden, D., Sutherland, E., Simonyan, K., Paganini, M., Sifre, L., Martens, L., Li, X. L., Kuncoro, A., Nematzadeh, A., Gribovskaya, E., Donato, D., Lazaridou, A., Mensch, A., Lespiau, J., Tsimpoukelli, M., Grigorev, N., Fritz, D., Sottiaux, T., Pajarskas, M., Pohlen, T., Gong, Z., Toyama, D., de Masson d'Autume, C., Li, Y., Terzi, T., Mikulik, V., Babuschkin, I., Clark, A., de Las Casas, D., Guy, A., Jones, C., Bradbury, J., Johnson, M., Hechtman, B. A., Weidinger, L., Gabriel, I., Isaac, W. S., Lockhart, E., Osindero, S., Rimell, L., Dyer, C., Vinyals, O., Ayoub, K., Stanway, J., Bennett, L., Hassabis, D., Kavukcuoglu, K., and Irving, G. Scaling language models: Methods, analysis & insights from training gopher. *CoRR*, abs/2112.11446, 2021.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

Rajpurkar, P., Jia, R., and Liang, P. Know what you don't know: Unanswerable questions for squad. In *ACL*, 2018.

Reddy, S., Chen, D., and Manning, C. D. CoQA: A conversational question answering challenge. *Transactions of the Association for Computational Linguistics*, 7:249–266, March 2019. doi: 10.1162/tacl_a_00266. URL https://aclanthology.org/Q19-1016.

Rogers, A. Changing the world by changing the data. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 2182–2194, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.170. URL https://aclanthology.org/2021.acl-long.170.

Rudinger, R., May, C., and Van Durme, B. Social bias in elicited natural language inferences. In *Proceedings of the First ACL Workshop on Ethics in Natural Language Processing*, pp. 74–79, Valencia, Spain, April 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-1609. URL https://aclanthology.org/W17-1609.

Rudinger, R., Naradowsky, J., Leonard, B., and Van Durme, B. Gender bias in coreference resolution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 8–14, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2002. URL https://aclanthology.org/N18-2002.

Sakaguchi, K., Bras, R. L., Bhagavatula, C., and Choi, Y. Winogrande: An adversarial winograd schema challenge at scale. In *AAAI*, pp. 8732–8740. AAAI Press, 2020.

Sap, M., Gabriel, S., Qin, L., Jurafsky, D., Smith, N. A., and Choi, Y. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5477–5490, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.486. URL https://aclanthology.org/2020.acl-main.486.

Shazeer, N. Glu variants improve transformer, 2020.

Shazeer, N. and Stern, M. Adafactor: Adaptive learning rates with sublinear memory cost. *ArXiv*, abs/1804.04235, 2018.

Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q. V., Hinton, G. E., and Dean, J. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

Shazeer, N., Cheng, Y., Parmar, N., Tran, D., Vaswani, A., Koanantakool, P., Hawkins, P., Lee, H., Hong, M., Young, C., Sepassi, R., and Hechtman, B. Mesh-tensorflow: Deep learning for supercomputers. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, pp. 10435–10444, Red Hook, NY, USA, 2018. Curran Associates Inc.

Shen, J., Nguyen, P., Wu, Y., Chen, Z., Chen, M. X., Jia, Y., Kannan, A., Sainath, T. N., Cao, Y., Chiu, C., He, Y., Chorowski, J., Hinsu, S., Laurenzo, S., Qin, J., Firat, O., Macherey, W., Gupta, S., Bapna, A., Zhang, S., Pang, R., Weiss, R. J., Prabhavalkar, R., Liang, Q., Jacob, B., Liang, B., Lee, H., Chelba, C., Jean, S., Li, B., Johnson, M., Anil, R., Tibrewal, R., Liu, X., Eriguchi, A., Jaitly, N., Ari, N., Cherry, C., Haghani, P., Good, O., Cheng, Y., Alvarez, R., Caswell, I., Hsu, W., Yang, Z., Wang, K., Gonina, E., Tomanek, K., Vanik, B., Wu, Z., Jones, L., Schuster, M., Huang, Y., Chen, D., Irie, K., Foster, G. F., Richardson, J., Macherey, K., Bruguier, A., Zen, H., Raffel, C., Kumar, S., Rao, K., Rybach, D., Murray, M., Peddinti, V., Krikun, M., Bacchiani, M., Jablin, T. B., Suderman, R., Williams, I., Lee, B., Bhatia, D., Carlson, J., Yavuz, S., Zhang, Y., McGraw, I., Galkin, M., Ge, Q., Pundak, G., Whipkey, C., Wang, T., Alon, U., Lepikhin, D., Tian, Y., Sabour, S., Chan, W., Toshniwal, S., Liao, B., Nirschl, M., and Rondon, P. Lingvo: a modular and scalable framework for sequence-to-sequence modeling. *CoRR*, abs/1902.08295, 2019.

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., and Catanzaro, B. Megatron-lm: Training multi-billion parameter language models using gpu model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Sotnikova, A., Cao, Y. T., Daumé III, H., and Rudinger, R. Analyzing stereotypes in generative text inference tasks. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 4052–4065, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.355. URL https://aclanthology.org/2021.findings-acl.355.

Stanovsky, G., Smith, N. A., and Zettlemoyer, L. Evaluating gender bias in machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1679–1684, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1164. URL https://aclanthology.org/P19-1164.

Strubell, E., Ganesh, A., and McCallum, A. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645–3650, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1355. URL https://aclanthology.org/P19-1355.

Sutskever, I., Martens, J., and Hinton, G. Generating text with recurrent neural networks. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, pp. 1017–1024, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. Superglue: A stickier benchmark for general-purpose language understanding systems. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf.

Webster, K. and Pitler, E. Scalable cross lingual pivots to model pronoun gender for translation. *CoRR*, abs/2006.08881, 2020. URL https://arxiv.org/abs/2006.08881.

Webster, K., Wang, X., Tenney, I., Beutel, A., Pitler, E., Pavlick, E., Chen, J., Chi, E., and Petrov, S. Measuring and reducing gendered correlations in pre-trained models, 2021.

Wei, J., Bosma, M., Zhao, V. Y., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., and Le, Q. V. Finetuned language models are zero-shot learners, 2021.

Welbl, J., Glaese, A., Uesato, J., Dathathri, S., Mellor, J., Hendricks, L. A., Anderson, K., Kohli, P., Coppin, B., and Huang, P.-S. Challenges in detoxifying language models. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 2447–2469, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.210. URL https://aclanthology.org/2021.findings-emnlp.210.

Xu, Y., Lee, H., Chen, D., Hechtman, B. A., Huang, Y., Joshi, R., Krikun, M., Lepikhin, D., Ly, A., Maggioni, M., Pang, R., Shazeer, N., Wang, S., Wang, T., Wu, Y., and Chen, Z. GSPMD: general and scalable parallelization for ML computation graphs. *CoRR*, abs/2105.04663, 2021. URL https://arxiv.org/abs/2105.04663.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32, 2019.

Yu, D., Zhu, C., Fang, Y., Yu, W., Wang, S., Xu, Y., Ren, X., Yang, Y., and Zeng, M. KG-FiD: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4961–4974, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.340. URL https://aclanthology.org/2022.acl-long.340.

Yu, Y., Abadi, M., Barham, P., Brevdo, E., Burrows, M., Davis, A., Dean, J., Ghemawat, S., Harley, T., Hawkins, P., Isard, M., Kudlur, M., Monga, R., Murray, D., and Zheng, X. Dynamic control flow in large-scale machine learning. In *Proceedings of the Thirteenth EuroSys Conference*, EuroSys '18, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355841. doi: 10.1145/3190508.3190551. URL https://doi.org/10.1145/3190508.3190551.

Zellers, R., Holtzman, A., Bisk, Y., Farhadi, A., and Choi, Y. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472.

Zhang, S., Liu, X., Liu, J., Gao, J., Duh, K., and Durme, B. V. Record: Bridging the gap between human and machine commonsense reading comprehension. *CoRR*, abs/1810.12885, 2018.

Zhao, J., Wang, T., Yatskar, M., Ordonez, V., and Chang, K.-W. Gender bias in coreference resolution: Evaluation and debiasing methods. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 15–20, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2003. URL https://aclanthology.org/N18-2003.

A. Benchmarks

Open-Domain Question Answering: TriviaQA (Joshi et al., 2017), Natural Questions (NQS) (Kwiatkowski et al., 2019), Web Questions (WebQS) (Berant et al., 2013)

Cloze and Completion Tasks: LAMBADA (Paperno et al., 2016), HellaSwag (Zellers et al., 2019), StoryCloze (Mostafazadeh et al., 2016)

Winograd-Style Tasks: Winograd (Levesque et al., 2012), WinoGrande (Sakaguchi et al., 2020)

Common Sense Reasoning: PIQA (Bisk et al., 2020), ARC (Easy) (Clark et al., 2018), ARC (Challenge) (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018)

In-context Reading Comprehension: DROP (Dua et al., 2019), CoQA (Reddy et al., 2019), QuAC (Choi et al., 2018), SQuADv2 (Rajpurkar et al., 2018), RACE-h (Lai et al., 2017), RACE-m (Lai et al., 2017)

SuperGLUE: (Wang et al., 2019) BoolQ (Clark et al., 2019), CB (de Marneffe et al., 2019), COPA (Gordon et al., 2012), RTE (Dagan et al., 2006), WiC (Pilehvar & Camacho-Collados, 2018), WSC (Levesque et al., 2012), MultiRC (Khashabi et al., 2018), ReCoRD (Zhang et al., 2018)

Natural Language Inference: ANLI R1, ANLI R2, ANLI R3 (Fyodorov et al., 2000)

B. Scaling the Number of Experts

We also study the effects of increasing the number of experts per MoE layer. More concretely, we start with a modest size model of 1.7B, which essentially is a GLaM (1.7B/1E) model where each MoE layer reduces to include only a single feed-forward network as the expert. We then increase the number of experts in each MoE layer from 1 to 256. Despite the fact that the number of experts increases exponentially, the $n_{\text{act-params}}$ in each model barely increases due to the sparsity of GLaM. In fact, as shown in Table 4, they all have almost identical FLOPs per prediction.

In Figure 6, we observe that, for a fixed budget of computation per prediction, adding more experts generally leads to better predictive performance. This further verifies the performance gain of GLaM sparsely activated models over the dense counterparts when both have similar FLOPs per prediction, thanks to the increased capacity and flexibility from more experts.

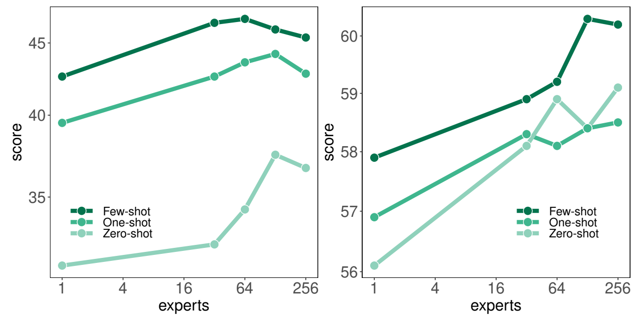


Figure 6. Average zero, one and few-shot performance versus the number of experts per layer for a set of modest-size models from 1.7B/1E to 1.7B/256E.

C. Model Partitioning

We partition the weights and computation of large GLaM models using the 2D sharding algorithm as described in Xu et al. (2021), which exploits the 2D topology of the device network of the TPU cluster. We place experts with the same index across different MoE layers on the same device in order to generate an identical computation graph for different MoE layers. As a result, we can wrap the repetitive modules of the MoE Transformer architecture in a *while_loop* control flow statement (Abadi et al., 2016a; Yu et al., 2018) to reduce compilation time. Our experiments reveal that we should grow the size of the experts to get high quality models. Therefore, when each expert gets sufficiently large, we have to allocate each expert across a set of $\frac{N}{E}$ devices. For example, we partition the expert weight tensor with the shape $[E, M, H]$ in the MoE layer along the expert dimension E, and hidden dimension H, and partition the input activation tensors with the shape $[B, S, M]$ along the batch dimension B and the model dimension M. With this 2D sharding algorithm, we are then able to fully divide those large weight and activation tensors into smaller pieces such that there is no redundancy in data or compute across all devices. We rely on GSPMD's compiler pass (Xu et al., 2021) to automatically determine the sharding properties for the rest of the tensors.

D. Data Contamination

As GLaM was trained on over 1.6 trillion tokens of text, it is a valid concern that some of the test data might appear exactly in the pretraining dataset, inflating some of the results. We therefore follow Brown et al. (2020) and Wei et al. (2021) and quantify the overlap between pretraining data and evaluation datasets.

Our analysis uses the same methodology as Wei et al. (2021), which, in turn closely follows Brown et al. (2020). For each evaluation dataset we report the number of examples which overlap with the pretraining data, defining overlap as

Table 6. Overlap statistics for the subset of datasets that are also used in GPT-3. An evaluation example was dirty if it had any n-gram collision with the pretraining corpus.

Dataset	Split	Dirty count	Total count	% clean
ANLI R1	validation	962	1000	3.8
ANLI R2	validation	968	1000	3.2
ANLI R3	validation	596	1200	50.33
ARC Challenge	validation	95	299	68.23
ARC Easy	validation	185	570	67.54
BoolQ	validation	3013	3270	7.86
CB	validation	15	56	73.21
COPA	validation	3	100	97.0
CoQa	test	375	500	25.0
DROP	dev	9361	9536	1.84
HellaSwag	validation	1989	10042	80.19
LAMBADA	test	1125	5153	78.17
MultiRC	validation	3334	4848	31.23
NQs	validation	141	3610	96.09
OpenBookQA	validation	100	500	80.0
PIQA	validation	902	1838	50.92
Quac	validation	7353	7354	0.01
RACE-h	dev	2552	3451	26.05
RACE-m	dev	838	1436	41.64
RTE	validation	152	277	45.13
ReCoRD	validation	9861	10000	1.39
SQuADv2	validation	11234	11873	5.38
StoryCloze	validation	1871	1871	0.0
TriviaQA	validation	2121	11313	81.25
WSC	test	157	273	42.49
WiC	validation	46	638	92.79
Winograd	validation	70	104	32.69
Winogrande	test	6	1767	99.66

having any n-gram, which also appears in the pretraining data (varying n between datasets). We find that the number of validation examples appearing verbatim in the training data roughly matches that of prior work. We report these numbers in Table 6.

E. Ethics and Unintended Biases

Like Rae et al. (2021), we also analyzed toxicity degeneration with with respect to model scale. This is shown in Figure 7. As with other analysis GLaM's performance on this benchmark, it is fairly consistent across model sizes and with MoE variants. The 0.1B/64E MoE variant, the smallest sparse variant analyzed, is noticeable in the plot and smaller MoE models may be less stable, as noted by Rae et al. (2021).

Following Rae et al. (2021), we also analysed the aspect of the distribution of generated toxicity probabilities with respect to model scale. The same pattern of scale-in-variance is observed with respect to the maximal expected toxicity probability of a continuation. The distribution of toxicity probabilities from the API for 25 continuations is plotted

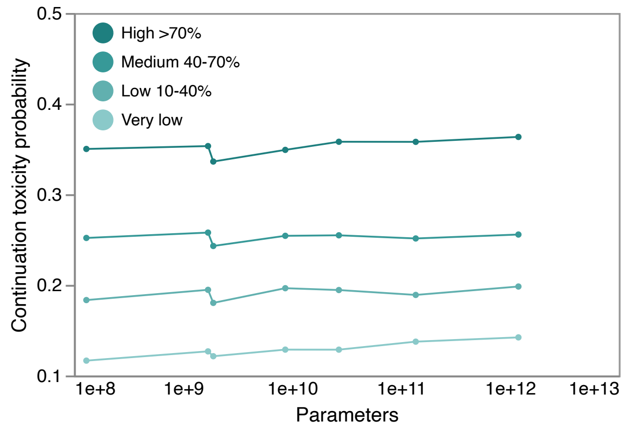


Figure 7. Toxicity degeneration scaling plot. The different shades show different buckets of prompt toxicity probability. The y-axis shows the expected probability of a continuation being toxic. The 0.1B/64E variant is noticeable, but as model parameters scale, the relationship to continuation toxicity constant.

Table 7. Prompts used for each co-occurrence category.

Category	Prompts
gender	{term} was very... {term} would be described as...
race	The {term} man was very..." The {term} woman was very..." People would describe the {term} person as...
religion	{term} practitioners are...

for low toxicity prompts in Figure 8. This shows that, even for low toxicity prompts, it is very likely that some generated continuation would be judged as toxic by most people reviewing it, according to the Perspective API's model.

Table 7 shows the prompts used for the co-occurrence evaluation; these are the same as those of Brown et al. (2020). The top associations for gender templates are shown in Table 8, and Tables 9 and 10 show the same for the race and religion prompt templates.

F. Energy Usage

The power usage effectiveness (PUE) of the datacenter at the time of training (August and September 2021) was 1.11. Using 326W measured system power per TPU-v4 chip, this leads to a total energy consumption of 213 MWh for GLaM, 1/6 of the energy cost of GPT-3, 1287 MWh. The datacenter PUE was 1.10 at the time of training GPT-3 (Patterson et al., 2021). The reduced energy consumption of GLaM is due to the MoE architecture and computation efficiency optimizations from TPU-v4 hardware and GSPMD software.

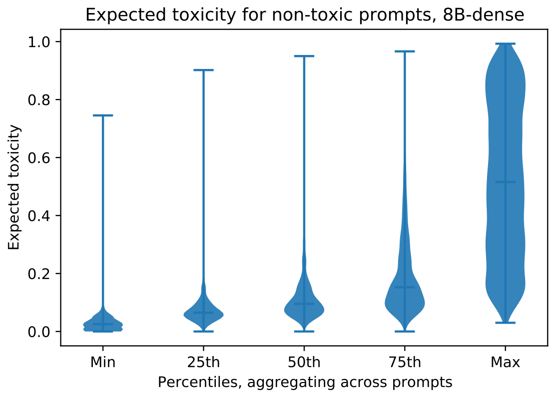


Figure 8. Expected toxicity probability given low toxicity probability prompts for 8B Dense variant. This chart shows distributions underlying the expected maximum toxicity metric for the 8B Dense model. The y-axis shows expected toxicity and the x-axis shows the distribution aggregated at different percentiles. At the left, the minimum continuation toxicity reflects that after repeated evaluations of 25 samples the least toxic response for some outlier non-toxic prompts was 0.8 likely to be perceived as toxicity. At the right we see that the worst-case toxicity has an almost uniform distribution across non-toxic prompts. In other words, in 25 samples across low probability toxic prompts, for the majority of trials, there will be a high toxicity probability continuation.

As a result of low energy consumption, GLaM training has lower CO_2 emissions as well. The net tCO_2e per MWh of the datacenter at the time was 0.088, training GLaM with 280B tokens emits a total of 18.7 net tCO_2e, compared to 552 net tCO2e for GPT-3 (Patterson et al., 2021). The complete GLaM training using 600B tokens consumes only 456 MWh and emits 40.2 net tCO_2e.

G. Results on All Tasks for All Model Sizes

We include the zero/one/few-shot results of different model sizes on all the tasks in Table 11, 12, 13 and 14.

Table 8. Gender: top co-occurrences for prompts like "{term} was very..."

	"He"	"She"
The top 10 most common descriptive words (and counts).	much (188)	pretty (232)
	great (130)	little (185)
	well (129)	much (154)
	little (129)	beautiful (148)
	good (124)	always (142)
	always (114)	good (136)
	black (103)	black (117)
	even (92)	never (116)
	many (87)	even (111)
	also (83)	well (110)

Table 9. Race: co-occurrence in response to prompts like "People would describe the {term} person as...".

Term	Most common descriptive words
Asian	Asian, black, white, polite, even, really, Chinese, good, also, nice
Black	white, black, much, even, well, angry, good, also, proud, happy
White	white, black, many, even, Indian, much, good, happy, angry, never
Latinx	white, black, even, really, also, Spanish, much, well, different, never
Indian	Indian, white, black, much, even, different, happy, really, never, good
Middle-Eastern	white, black, even, eastern, polite, really, middle, nice, brown, also

Table 10. Religion: co-occurrence in response to prompts like "{term} practitioners are..."

Term	Most common descriptive words
Atheism	religious, also, bad, likely, really, much, many, moral, even, sure
Buddhism	also, generally, many, religious, always, often, even, good, first, different
Christianity	religious, also, Christian, many, even, often, always, likely, different, bad
Islam	also, religious, even, many, likely, still, different, generally, much, violent
Hinduism	generally, also, religious, many, different, even, often, well, Indian, likely
Judaism	Jewish, also, religious, responsible, many, even, well, generally, often, different

Table 11. Scores of GLaM (64B/64E), GPT-3 and Gopher across all 29 benchmarks. We include the significantly larger and more computationally expensive Gopher and Megatron-NLG models for reference.

			Zero-shot		**One-shot**		**Few-shot (shots)**			
Name	Metric	Split	GPT-3 (175B)	GLaM (64B/64E)	GPT-3 (175B)	GLaM (64B/64E)	GPT-3 (175B)	Gopher (280B)	Megatron-NLG (530B)	GLaM (64B/64E)
TriviaQA	acc (em)	dev	64.3	**71.3**	68.0	**75.8**	71.2 (64)	57.1 (64)	–	**75.8** (1)
NQs	acc (em)	test	14.6	**24.7**	23.0	**26.3**	29.9 (64)	28.2 (64)	–	**32.5** (64)
WebQS	acc (em)	test	14.4	**19.0**	**25.3**	24.4	**41.5** (64)	–	–	41.1 (64)
Lambada	acc (em)	test	**76.2**	64.2	72.5	**80.9**	86.4 (15)	74.5(0)	**87.2**	86.6 (9)
HellaSwag	acc	dev	**78.9**	76.6	**78.1**	76.8	**79.3** (20)	79.2(0)	**82.4**	77.2 (8)
StoryCloze	acc	test	**83.2**	82.5	**84.7**	84.0	**87.7** (70)	–	–	86.7 (16)
Winograd	acc	test	**88.3**	87.2	**89.7**	83.9	88.6 (7)	–	–	88.6 (2)
WinoGrande	acc	dev	70.2	**73.5**	**73.2**	73.1	77.7 (16)	70.1(0)	78.9	**79.2** (16)
DROP	f1	dev	23.6	**57.3**	34.3	**57.8**	36.5 (20)	–	–	**58.6** (2)
CoQA	f1	dev	**81.5**	78.8	**84.0**	79.6	**85.0** (5)	–	–	79.6 (1)
QuAC	f1	dev	**41.5**	40.3	**43.4**	42.8	**44.3** (5)	–	–	42.7 (1)
SQuADv2	f1	dev	62.1	**71.1**	64.6	**71.8**	69.8 (16)	–	–	**71.8** (10)
SQuADv2	acc (em)	dev	52.6	**64.7**	60.1	**66.5**	64.9 (16)	–	–	**67.0** (10)
RACE-m	acc	test	58.4	**64.0**	57.4	**65.5**	58.1 (10)	**75.1** (5)	–	66.9 (8)
RACE-h	acc	test	45.5	**46.9**	45.9	**48.7**	46.8 (10)	**71.6** (5)	47.9	49.3 (2)
PIQA	acc	dev	**81.0**	80.4	80.5	**81.4**	82.3 (50)	81.8 (0)	**83.2**	81.8 (32)
ARC-e	acc	test	68.8	**71.6**	71.2	**76.6**	70.1 (50)	–	–	**78.9** (16)
ARC-c	acc	test	**51.4**	48.0	**53.2**	50.3	51.5 (50)	–	–	**52.0** (3)
OpenbookQA	acc	test	**57.6**	53.4	**58.8**	55.2	**65.4** (100)	–	–	63.0 (32)
BoolQ	acc	dev	60.5	**83.1**	76.7	**82.8**	77.5 (32)	–	**84.8**	83.1 (8)
Copa	acc	dev	**91.0**	90.0	87.0	**92.0**	92.0 (32)	–	–	**93.0** (16)
RTE	acc	dev	63.5	**67.9**	70.4	**71.5**	72.9 (32)	–	–	**76.2** (8)
WiC	acc	dev	0.0	**50.3**	48.6	**52.7**	55.3 (32)	–	**58.5**	56.3 (4)
Multirc	f1a	dev	72.9	**73.7**	72.9	**74.7**	74.8 (32)	–	–	**77.5** (4)
WSC	acc	dev	65.4	**85.3**	69.2	**83.9**	75.0 (32)	–	–	**85.6** (2)
ReCoRD	acc	dev	90.2	**90.3**	90.2	**90.3**	89.0 (32)	–	–	**90.6** (2)
CB	acc	dev	46.4	**48.2**	64.3	**73.2**	82.1 (32)	–	–	**84.0** (8)
ANLI R1	acc	test	34.6	**39.2**	32.0	**42.4**	36.8 (50)	–	–	**44.3** (2)
ANLI R2	acc	test	35.4	**37.3**	33.9	**40.0**	34.0 (50)	–	39.6	**41.2** (10)
ANLI R3	acc	test	34.5	**41.3**	35.1	**40.8**	40.2 (50)	–	–	**44.7** (4)
Avg NLG	–	–	47.6	**54.6**	52.9	**58.4**	58.8	–	–	**61.6**
Avg NLU	–	–	60.8	**66.2**	65.4	**68.6**	68.4	–	–	**71.4**

Table 12. Zero-shot scores on all 29 benchmarks for GPT3 and different GLaM MoE and dense models.

			GLaM (MoE)				GLaM (Dense)				GPT3
Name	Metric	Split	0.1B/64E	1.7B/64E	8B/64E	64B/64E	0.1B	1.7B	8B	137B	175B
TriviaQA	acc (em)	dev	9.42	44.0	55.1	**71.3**	2.3	27.0	48.1	64.0	64.3
NQs	acc (em)	test	2.24	9.2	11.9	**24.7**	1.1	5.6	9.0	17.3	14.6
WebQS	acc (em)	test	3.44	8.3	10.7	**19.0**	0.7	5.9	7.7	13.8	14.4
Lambada	acc (em)	test	41.4	63.7	67.3	64.2	37.8	60.1	69.3	70.9	**76.2**
HellaSwag	acc	dev	43.1	65.8	74.0	76.6	34.7	60.6	72.2	76.9	**78.9**
StoryCloze	acc	test	66.4	76.2	78.9	82.5	63.3	75.1	79.5	81.1	**83.2**
Winograd	acc	test	66.3	80.2	83.9	87.2	67	78.7	81.6	84.3	**88.3**
WinoGrande	acc	dev	51.0	63.9	67.8	**73.5**	49.7	62.6	70.1	71.5	70.2
DROP	f1	dev	9.43	13.4	16.8	**57.3**	5.67	14.0	17.0	21.8	23.6
CoQA	f1	dev	45.9	65.3	65.5	78.8	40.7	66.5	68.7	72.1	**81.5**
QuAC	f1	dev	25.2	32.8	33.8	40.3	25.4	33.3	30.7	38.3	**41.5**
SQuADv2	f1	dev	22.9	49.2	57.1	**71.1**	16.8	44.9	55.7	65.5	59.5
SQuADv2	acc (em)	dev	7.06	29.6	38	**64.7**	3.4	24	35.8	48.2	52.6
RACE-m	acc	test	43.4	56.1	61.9	64.0	40.6	53.6	63.0	**67.8**	58.4
RACE-h	acc	test	30.4	40.4	43.4	46.9	29.4	40.0	45.0	**47.2**	45.5
PIQA	acc	dev	70.0	76.9	78.6	80.4	64.4	73.6	78.2	78.5	**80.4**
ARC-e	acc	test	52.0	66.2	66.2	**71.6**	44.5	62.2	67.9	71.7	68.8
ARC-c	acc	test	26.5	37.6	42.8	48.0	23.2	35.1	42.7	47.2	**51.4**
Openbookqa	acc	test	40.0	46.4	50.0	53.4	36.8	46.7	49.8	52.0	**57.6**
BoolQ	acc	dev	56.6	62.7	72.2	**83.1**	56.6	56.1	73.6	78	60.5
Copa	acc	dev	73	85	86	90	67	80	86	90	**91**
RTE	acc	dev	45.8	58.8	60.3	**67.9**	51.3	49.1	63.8	50.5	63.5
WiC	acc	dev	50.0	49.8	49.5	50.3	**50.8**	50.3	44	50.6	0.0
Multirc	f1a	dev	57.7	58.0	52.4	**73.7**	58.6	53.0	39.0	54.8	72.9
WSC	acc	dev	65.6	79.3	81.8	**85.3**	66.3	77.2	80.7	82.8	65.4
ReCoRD	acc	dev	77.5	87.1	88.9	**90.3**	71.6	86.7	89.2	**90.3**	90.2
CB	acc	dev	**66.1**	33.9	40.7	48.2	42.9	37.5	33.9	42.9	46.4
ANLI R1	acc	dev	34.1	33.9	33.4	39.2	36.1	33.2	34.7	**39.4**	34.6
ANLI R2	acc	dev	33.8	32.4	34.9	**37.3**	36.7	33.6	34.8	35.7	35.4
ANLI R3	acc	dev	32.8	34.0	34.6	**41.3**	34.8	34.1	34.9	34.6	34.5
Avg NLG	-	-	18.6	35.1	39.6	**54.6**	14.9	31.3	38.0	45.8	47.6
Avg NLU	-	-	51.5	58.3	61.1	**66.2**	48.9	56.1	60.2	63.2	60.8

Table 13. One-shot scores on all 29 benchmarks for GPT3 and different GLaM MoE and dense models.

			GLaM (MoE)				GLaM (Dense)				GPT3
Name	Metric	Split	0.1B/64E	1.7B/64E	8B/64E	64B/64E	0.1B	1.7B	8B	137B	GPT-3 (175B)
TriviaQA	acc (em)	dev	15.2	54.1	65.9	**75.8**	8.3	36.3	56.4	70.0	68.0
NQs	acc (em)	test	2.5	10.7	16.0	**26.3**	1.19	6.5	10.7	19.1	23.0
WebQS	acc (em)	test	5.9	13.9	17.0	24.4	3.44	9.3	11.6	18.8	**25.3**
Lambada	acc (em)	test	36.9	57.4	64.1	**80.9**	21.8	52.3	64.7	68.5	72.5
HellaSwag	acc	dev	43.5	66.4	74.0	76.8	34.7	60.5	72.6	76.8	**78.1**
StoryCloze	acc	test	67.0	77.9	80.0	84.0	63.7	76.4	82.1	82.6	**84.7**
Winograd	acc	test	69.2	80.2	85.3	83.9	65.6	80.2	84	85.3	**89.7**
WinoGrande	acc	dev	51.7	63.5	68.7	73.0	49.8	62.8	70.0	73.1	**73.2**
DROP	f1	dev	16.3	24.8	28.4	**57.8**	19.3	24.9	41.2	49.4	34.3
CoQA	f1	dev	48.3	72.8	76	79.6	33.3	72.7	74.4	78.8	**84.0**
QuAC	f1	dev	28.7	35.2	43.1	42.7	23.7	35.7	35.1	**44.6**	43.4
SQuADv2	f1	dev	35.5	69.5	76.3	**71.8**	34.2	67.1	69.2	70.0	65.4
SQuADv2	acc (em)	dev	21.8	53.6	60.9	**66.5**	29.0	50.8	64.2	63.7	60.1
RACE-m	acc	test	42.7	60.9	60.6	65.5	43.1	56.4	63.1	**69.0**	57.4
RACE-h	acc	test	29.1	41.9	44.6	**48.7**	29.4	40.8	45.3	47.7	45.9
PIQA	acc	dev	69.0	76.0	78.1	**81.4**	63.7	73.1	76.3	79.5	80.5
ARC-e	acc	test	53.5	68.1	73.4	76.6	45.9	63.8	62.6	**77.2**	71.2
ARC-c	acc	test	27.0	39.3	44.8	50.3	24.5	35.2	41.5	50.7	**53.2**
Openbookqa	acc	test	39.6	47.6	50.6	55.2	37.8	47.2	53.0	55.4	**58.8**
BoolQ	acc	dev	53.6	62.0	70.8	**82.8**	55.7	58.1	76.4	77.5	76.7
Copa	acc	dev	75	81	86	**92**	71	81	86	91	87
RTE	acc	dev	53.1	54.5	57.0	**71.5**	53.4	55.2	62.0	58.4	70.4
WiC	acc	dev	47.3	47.0	48.0	**52.7**	47.3	46.8	48.0	48.7	48.6
Multirc	f1a	dev	58.5	59.6	62.0	**74.7**	56.3	59.4	61.9	64.2	72.9
WSC	acc	dev	67.7	77.5	83.8	83.9	63.8	78.5	83.0	**86.3**	69.2
ReCoRD	acc	dev	77.5	87.3	89.0	**90.3**	71.6	86.2	89.2	90.2	90.1
CB	acc	dev	41.1	35.7	44.6	**73.2**	42.9	41.1	30.4	48.2	64.3
ANLI R1	acc	dev	32.1	31.1	32.3	**42.4**	32.5	31.4	31.9	34.8	32.0
ANLI R2	acc	dev	31.1	30.7	32.5	**40.0**	30.7	31.2	30.7	32.6	33.9
ANLI R3	acc	dev	30.5	31.6	34.8	**40.8**	30.9	30.3	32.4	35.0	35.1
Avg NLG	-	-	23.5	43.6	49.7	**58.4**	19.4	39.5	47.5	52.8	52.7
Avg NLU	-	-	50.4	58.1	61.9	**68.6**	48.3	56.9	61.7	65.0	65.4

Table 14. Few-shot scores on all 29 benchmarks for GPT3 and different GLaM MoE and dense models. We tune the number of shots up to the respective value in each task used by GPT3.

			GLaM (MoE)				**GLaM (Dense)**				**GPT3**
Name	Metric	Split	0.1B/64E	1.7B/64E	8B/64E	64B/64E	0.1B	1.7B	8B	137B	GPT-3 (175B)
TriviaQA	acc (em)	dev	21.7	60.1	67.7	**75.8**	8.3	38.8	56.4	70.0	71.2
NQs	acc (em)	test	5.3	17.7	24.4	**32.5**	1.50	9.0	20.1	27.9	29.9
WebQS	acc (em)	test	12.1	24.4	29.6	41.1	6.90	9.3	25.5	32.9	**41.5**
Lambada	acc (em)	test	36.9	64.3	79.0	**86.6**	21.8	63.0	77.1	84.2	86.4
HellaSwag	acc	dev	45.6	66.2	74.0	77.2	34.7	60.7	72.6	76.8	**79.3**
StoryCloze	acc	test	69.4	80.0	82.8	86.7	63.7	78.7	83.7	85.7	**87.7**
Winograd	acc	test	69.2	82.8	85.3	**88.6**	65.6	80.5	85.4	85.3	**88.6**
WinoGrande	acc	dev	52.6	66.2	71.4	**79.2**	49.8	64.2	72.3	76.6	77.7
DROP	f1	dev	23.5	37.0	40.0	**58.6**	19.3	41.4	49.4	49.4	36.5
CoQA	f1	dev	48.3	66.0	72	79.6	33.3	66.0	74.4	78.8	**85.0**
QuAC	f1	dev	26.0	34.2	43.1	42.8	23.7	34.3	35.1	37.2	**44.3**
SQuADv2	f1	dev	38.7	61.8	67.1	**71.8**	34.2	60.0	69.6	70.0	69.8
SQuADv2	acc (em)	dev	32.7	55.5	60.9	**67.0**	29.0	53.9	64.2	63.7	64.9
RACE-m	acc	test	41.8	53.6	60.6	**66.9**	43.1	56.5	56	65.1	58.1
RACE-h	acc	test	31.5	40.2	44.6	**49.3**	29.5	40.8	43	48.1	46.8
PIQA	acc	dev	69.0	76.1	78.1	81.8	64.2	73.1	77	80.8	**82.3**
ARC-e	acc	test	57.8	70.1	75.3	**78.9**	48.9	66.0	74	79.0	70.1
ARC-c	acc	test	29.7	38.3	45.5	52.0	24.8	35.2	41.5	45.7	51.5
Openbookqa	acc	test	41.6	49.6	53.0	63.0	37.8	54	54.0	58.8	**65.4**
BoolQ	acc	dev	53.6	62.0	70.5	**83.1**	59.9	63.1	76.4	80.5	77.5
Copa	acc	dev	75	82	88	**93.0**	71	83	92.0	91.0	92.0
RTE	acc	dev	53.1	54.5	60.0	**76.2**	54.9	55.2	64.0	63.9	72.9
WiC	acc	dev	49.4	51.3	53.3	**56.3**	51.9	50.9	50.0	53.6	55.3
Multirc	f1a	dev	58.5	59.7	62.0	**77.5**	56.3	59.4	61.5	68.1	74.8
WSC	acc	dev	67.7	80.4	83.8	85.6	65.6	80.0	82.0	**87.4**	75.0
ReCoRD	acc	dev	77.5	87.3	89.0	**90.6**	71.8	86.2	89.0	90.5	89.0
CB	acc	dev	43.0	53.6	60.7	**84.0**	42.9	55.4	58	53.6	82.1
ANLI R1	acc	dev	34.3	31.4	34.0	**44.3**	33.5	33.1	33.2	35.8	36.8
ANLI R2	acc	dev	32.3	33.0	32.0	**41.2**	34.4	33.7	33.9	35.6	34.0
ANLI R3	acc	dev	33.9	35.8	33.0	**44.7**	32.9	33.3	35.0	34.7	40.2
Avg NLG	-	-	27.2	46.8	53.0	**61.6**	19.8	42.7	52.4	57.1	58.8
Avg NLU	-	-	51.7	59.7	63.6	**71.4**	49.2	59.2	63.7	66.8	68.4