FORGETFUL CAUSAL MASKING MAKES CAUSAL LANGUAGE MODELS BETTER ZERO-SHOT LEARNERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLM) trained using the next-token-prediction objective, such as GPT3 and PaLM, have revolutionized natural language processing in recent years by showing impressive zero-shot and few-shot capabilities across a wide range of tasks. In this work, we propose a simple technique that significantly boosts the performance of LLMs without adding computational cost. Our key observation is that, by performing the next token prediction task with randomly selected past tokens masked out, we can improve the quality of the learned representations for downstream language understanding tasks. We hypothesize that randomly masking past tokens prevents over-attending to recent tokens and encourages attention to tokens in the distant past. By randomly masking input tokens in the PaLM model, we show that we can significantly improve PaLM's zero-shot performance on the SuperGLUE benchmark from 55.7 to 59.2. Experimental results show that FCM also improves PaLM's zero and few-shot performance on a diverse suite of tasks, including commonsense reasoning, natural language inference and cloze completion. Moreover, we show that our technique also helps representation learning, significantly improving PaLM's finetuning results on SuperGLUE.

1 INTRODUCTION

Language model (LM) pre-training has substantially advanced the state-of-the-art across a variety of natural language processing tasks (Peters et al., 2018; Devlin et al., 2018; Brown et al., 2020; Chowdhery et al., 2022) and related fields including image generation, reasoning, and code generation (Alayrac et al., 2022; Lewkowycz et al., 2022; Saharia et al., 2022; Chen et al., 2021). Prior work on pre-training have focused on mixing different choices of architecture (*e.g.*, encoder-only, decoder-only, or encoder-decoder) with different objective functions (*e.g.*, masking or causal language modeling). For example, masked encoder-only models such as BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019) excel in discriminative finetuning tasks such as classification. Similiarly, masked encoder-decoder models such as BART (Lewis et al., 2019) and T5 (Roberts et al., 2019) perform well on both discriminative and generative finetuning. While masked language modeling is effective for finetuning and removes the need for task-specific architectures, its major limitation is that there is still a need for task-specific datasets and task-specific finetuning. On the other hand, decoder-only causal language models remove such limitations. In fact, they are capable of zero-shot and few-shot adaptation without the need for finetuning, by simply prompting the model with appropriate strings to control the generated outputs, as shown in GPT3 (Brown et al., 2020) and PaLM (Chowdhery et al., 2022).

Driven by its impressive zero-shot and few-shot abilities, there has been more work on scaling causal decoder-only architectures (Zhang et al., 2022; Black et al., acl; Brown et al., 2020; Chowdhery et al., 2022) compared to encoder-based architectures, and there has been significant interests in studying such models in various contexts (Hoffmann et al., 2022; Wei et al., 2022b; Li & Liang, 2021; Ahn et al., 2022; Chen et al., 2021). However, such decoder-only models are still limited by their imperfect zero-shot and few-shot adaptation compared to human performance, and their relatively inferior finetuning performance compared to masked language modeling.

To address the above challenges, prior work have proposed to combine masked modeling with causal language modeling (Dong et al., 2019; Wang et al., 2022; Tay et al., 2022; Du et al., 2022) to bring

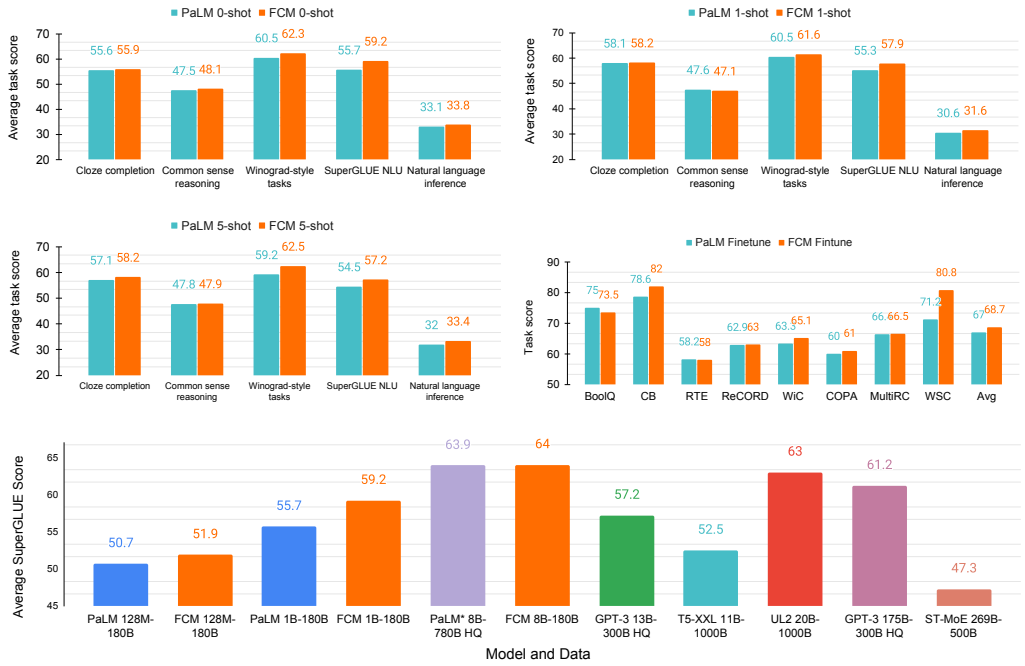

Figure 1: FCM outperforms PaLM in zero- and few-shot as well as finetuning tasks. **Top & middle**. Task average performance grouped by categories. The model size is 1B. We report the averaged scores in each category. Scores are averaged over 3 evaluation random seeds. **Bottom**. SuperGLUE zero-shot performance by different model size and dataset size. PaLM* 8B-780B HQ denotes the published results of 8B model trained on 780B tokens from high quality datasets. FCM 8B-180B denotes the same 8B model trained on 180B tokens from C4 dataset using FCM as objective.

the benefit of masked modeling to causal language models while retaining their zero-shot ability. However, such approaches typically introduce extra computation and parameters or require using a sophisticated attention masking strategy which hinders practical usages (Yang et al., 2019; Tay et al., 2022). Moreover, they typically train encoder-decoder models which are not naturally suitable for zero- and few-shot inference tasks compared with decoder-only causal language models and are still outperformed by causal language models (Sanh et al., 2022; Brown et al., 2020; Chowdhery et al., 2022). In order to further improve causal language models few-shot abilities, some works proposed better prompt engineering methods (Liu et al., 2021; Lester et al., 2021; Ling et al., 2017; Wei et al., 2022b; Li & Liang, 2021) or better finetuning methods (Mishra et al., 2022; Wei et al., 2022a; Sanh et al., 2022). Prompt-based methods are sensitive to design (Lester et al., 2021; Liu et al., 2021), while finetuning-based approaches typically require a huge amount of supervision to work with as shown in Sanh et al. (2022). In addition, such methods can only improve pre-trained model and are unable to improve pre-training.

In this work, we propose a pre-training approach that does not incur any extra computation cost or parameters, to improve few-shot and zero-shot performance, as well as representation learning of causal language models. Our key observation is that, by performing next token prediction task with randomly selected past tokens masked out, we can improve the quality of the learned representations for downstream language understanding tasks. Our method, Forgetful Causal Masking (FCM), can be efficiently implemented by randomly masking input tokens in the causal language model. Applying our method to PaLM (Chowdhery et al., 2022), a state-of-the-art causal language model, we see significant improvement on the SuperGLUE (Sarlin et al., 2020) benchmark: our method significantly improves the 1B-model-size PaLM's zero-shot performance from 55.7 to 59.2. We also conduct extensive evaluation on the commonsense reasoning benchmark PIQA (Bisk et al., 2019), ARC (Yadav et al., 2019), and OpenBookQA (Mihaylov et al., 2018); the Winograd-style tasks Winograd (Sakaguchi et al., 2020) and WinoGrande (Kocijan et al., 2020); the natural language inference (NLI) benchmark ANLI (Nie et al., 2019); and cloze completion tasks StoryCloze (Mostafazadeh et al., 2016) and LAMBADA (Paperno et al., 2016); and find that our method improves the zero-shot and few-shot performance of PaLM on all of the diverse suite

of tasks. In addition, FCM improves representation learning, as shown in our SuperGLUE finetuning experimental results, where our method significantly improves 1B parameter PaLM model's finetuneing performance from 67.0 to 68.7.

Contributions. We highlight the contributions of our paper below:

- We present FCM, a simple and scalable pre-training methodology for causal language modeling. We provide the empirical evaluation of FCM on a suite of few-shot and finetuning benchmarks.
- We show that FCM is highly effective at improving zero-shot and few-shot learning results. In addition, we demonstrate that FCM significantly helps with representation learning.
- We demonstrate that FCM is scalable – it works well with various model sizes, from 128M to 1B and 8B. We show that FCM 8B model trained on C4 dataset (180B tokens) matches the zero-shot results of the 8B PaLM trained on a much larger dataset (780B tokens) that contains high-quality conversation data.

2 Related work

Masking strategies and pre-training objectives. Many self-supervised pre-training techniques have been proposed to leverage the vast availability of unsupervised data. Different architectures typically leverage different objectives. Decoder-only models are typically trained with causal language model objectives to mimic auto-regressive generation (Brown et al., 2020) and is found to be effective in cross-modality learning (Alayrac et al., 2022; Yu et al., 2022). Autoencoding denoising objectives have been used to learn a bidirectional contextualized encoder for natural language understanding (Devlin et al., 2018; Liu et al., 2019; Yang et al., 2019). For encoder-decoder models, BART (Lewis et al., 2019) conducts NLU tasks by feeding the same input into the encoder and decoder, and taking the final hidden states of the decoder. Raffel et al. (2020) explored many objectives of pre-training and found that span-corruption works best with encoder-decoder model. Other work explores multi-task pre-training using supervised data (Aribandi et al., 2021; Sanh et al., 2022; Wang et al., 2022). To study the impact of different objectives on zero-shot generalization, Wang et al. (2022) conducts a systematic study of different architectures combined with three different pre-training objectives, and found causal language modeling to be effective at zero-shot learning.

Combining causal and masked language modeling. There has been work explore training model with multiple objectives to combine causal and masked language modeling under the masked language modeling objective with different attention masks (Dong et al., 2019; Bao et al., 2020). Later work proposes to use blank infilling (Raffel et al., 2020) to randomly blank out continuous spans of tokens from the input text and train the model to sequentially reconstruct the spans (Du et al., 2022). XLNet (Yang et al., 2019) modifies the attention mask in a standard transformer to enable token generation in any permutation of tokens. XLNet uses a two-stream self-attention mechanism, instead of the right-shift, to avoid information leakage in Transformers, but doubles the time cost of pre-training. UL2 (Tay et al., 2022) further proposes to train language model using a mixture of denoisers to combines diverse pre-training paradigms together. Other work explored masking some spans that are predicted at the end of the sequence for bidirectional models (Artetxe et al., 2022) or left-to-right autoregressive models (Aghajanyan et al., 2022; Zhu et al., 2019; Donahue et al., 2020; Fried et al., 2022). Notably, Bavarian et al. (2022) explores moving text in the middle to the end and predict it autoregressively. Different from prior work, we focus on efficiently improving autoregressive model with masking language modeling, our method does not require complex implementations to change input or output prediction, making it simple to implement and our method does not add extra computation or parameters.

3 Method

3.1 Pre-training Objective

FCM uses a standard causal, decoder-only Transformer model architecture (Vaswani et al., 2017), *i.e.*, each timestep can only attend to itself and past timesteps. We illustrate FCM in Figure 2.

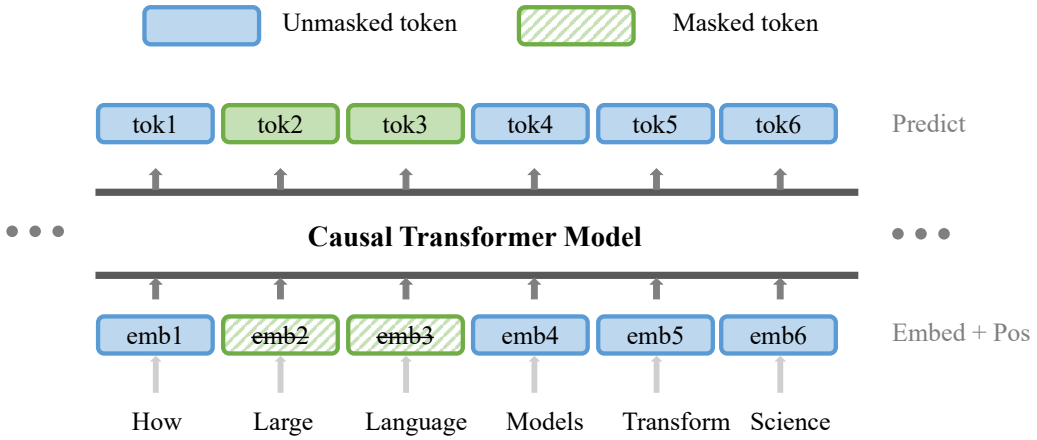

Figure 2: Illustrations of FCM. Given a causal language model, each token's prediction is conditioned on embeddings that are not masked. The loss is applied to each token in the sequence. In this example, when predicting the token of the word *"models"*, the embeddings of the words *"large"* and *"language"* are removed from the input sequence. The model is asked to predict all tokens autoregressively.

Given an input text $\mathbf{x} = [x_1, \cdots, x_n]$, the standard causal language modeling objective is defined to maximize the log likelihood of $\boldsymbol{x}$ autoregressively:

$$\begin{aligned} \log p(\mathbf{x}) &= \log \prod_{i=1}^{n} p(x_i | x_1, x_2, \ldots, x_{i-1}) \\ &= \log \prod_{i=1}^{n} p(x_i | \mathbf{x}_{<i}) := \log \prod_{i=1}^{n} p(x_i | [x_j]_{j=0}^{i-1}). \end{aligned} \tag{1}$$

In FCM, we randomly sample a mask ratio from $m \sim [0, \eta]$ where $\eta \in [0, 1]$ is a fixed maximum mask ratio, we use $\eta = 0.15$ throughout the experiments unless otherwise mentioned. The model is asked to predict each token $x_i \in \mathbf{x}$, and can only attend to tokens in $\mathbf{x}_{<i}$ that are not sampled. Concretely, the FCM objective is given by:

$$\log p(\mathbf{x}) = \log \prod_{i=1}^{n} p(x_i | [I[m_j > \eta] \cdot x_j]_{j=0}^{i-1}), \tag{2}$$

where $m_j \sim \mathcal{U}(0, 1)$. This can be efficiently implemented by combining it with causal attention mask. While applying random masking to the token sequence, we always exclude the special `BOS` ('beginning of sentence') token at the beginning of each sequence, so that the model is aware of the beginning of a sentence. Moreover, keeping the `BOS` token unmasked helps with training stability because it ensures that there is at least one token unmasked without changing the semantic meaning of the sequence. For example, when predicting token x_t for small t, it is possible that all tokens $[x_1, ..., x_{t-1}]$ are masked, which can cause instability in the training loss. We found that this technique enables us to train arbitrary high mask ratios without incurring instability.

3.2 Model Architecture

We use the same model and architecture as PaLM (Chowdhery et al., 2022), including the modified activation (Shazeer, 2020), multi-query attention (Shazeer, 2019), parallel layers (Wang & Komatsuzaki, 2021) and RoPE embeddings (Su et al., 2021) described therein, with the exception that we use SentencePiece (Kudo & Richardson, 2018) vocabulary with 32K tokens from C4 (Raffel et al., 2020). To study the dependence of FCM on model size, we train 3 different sizes of the model, ranging over three orders of magnitude from 125 million parameters, to 1 billion parameters, and to 8 billion parameters (see Table 1).

Model	Layers	# of heads	d_{model}	Batch size	Seq len
FCM 128M	8	4	1024	1024	1024
FCM 1B	16	8	2048	1024	1024
FCM 8B	32	16	4096	1024	1024

Table 1: Architecture details of different sized models. We list the number of layers, d_{model}, the number of attention heads and attention head size, training batch size, and sequence length. The feed-forward size d_{ff} is always $4 \times d_{\text{model}}$ and attention head size is always 256.

3.3 Method discussion

In this section, we discuss the connections and differences between FCM and other pre-training models. We are mainly concerned with how they improve few-shot, zero-shot, and finetuning performance.

XLNet (Yang et al., 2019) is also pre-trained with autoregressive objectives, but there are important distinctions between FCM and XLNet. FCM does not need permutation of the input sequence and designing two-stream self-attention mechanism to avoid the information leak within Transformer, which doubles the time cost of pre-training. Our method is much simpler and more scalable.

T5 Raffel et al. (2020) and UL2 (Tay et al., 2022) propose to train encoder-decoder or prefix language model architecture using the span-corruption objective. T5 and UL2 always predict spans in a fixed left-to-right order, and are therefore related to FCM in that our method also predicts masked tokens from left to right. However, FCM is an autoregressive model without an encoder that encodes full context information, so in principle, FCM can be combined together *e.g.*with a prefix language model. Empirically, FCM outperforms T5 and UL2 on NLU tasks with fewer parameters and data.

GLM (Du et al., 2022) proposes to extend autoregressive modeling with bidirectional context. They achieve this by selecting spans and move them to the end of sequence, then unselected tokens and past spans use non-causal attention and tokens within each span use causal attention. Similar to XLNet, this method introduces extra implementation overhead.

UniLM (Dong et al., 2019; Bao et al., 2020) combines different training objectives together by using different self-attention masks to control the access to context for each token. Similar to BERT and T5, UniLM is trained with an autoencoding objective with masked spans replaced by mask tokens. This introduces a gap between pre-training and downstream tasks, since there are no mask tokens in downstream tasks. Moreover, the model needs to be finetuned for natural language generation tasks (*e.g.*, summarization). In contrast, FCM is autoregressive and naturally suitable for generation tasks.

In summary, our method is simpler and focuses on autoregressive causal language models. Our method is easy to implement and does not introduce extra computation or parameters, and as experimental evaluations in Section 4 show, FCM is scalable and achieves superior results than baselines.

4 Experiments

4.1 Experimental setup

Training datasets We use C4 dataset to pre-train baselines and our model (Raffel et al., 2020). It is a colossal, cleaned version of Common Crawl's web crawl corpus[1] and consists of about 180 billion tokens using sentencepiece tokenizer (Kudo & Richardson, 2018). Note that GPT-3 and PaLM use significantly larger pre-training datasets, *e.g.*, the PaLM pre-training dataset consists of a high-quality corpus of 780 billion tokens that is a mixture of filtered webpages, social media conversations, and more. However, these datasets are not publicly available and training on it requires

[1] `https://commoncrawl.org`

tremendous compute resources. C4 is significantly smaller, which also reduces the compute cost of training the large models.

Training and inference. Our training optimizer follows PaLM, and use the Adafactor optimizer (Shazeer & Stern, 2018) which scales the learning rate by the root-mean-square of the parameter matrix. We use learning rate of 0.01 for the first 10,000 steps, which is then decayed at a rate of $1/\sqrt{k}$, where k is the step number. We train with momentum of $\beta_1 = 0.9$. The second-order moment interpolation value is computed as $\beta_2 = 1.0 - k^{-0.8}$, where k is the step number. Following typical large Transformer models training as in PaLM and GPT-3, models are trained without dropout, and dropout of 0.1 is used for finetuning. Our training and inference codebase is based on JAX and T5X, and all models are trained on TPU v4 Pods.

Evaluation tasks and metrics. We consider the following tasks and categorize them according to their focused evaluation properties:

- *Cloze and Completion tasks*: **LAMBADA** (Paperno et al., 2016) consists of word prediction tasks that test the understanding of narrative passages. **StoryCloze** (Mostafazadeh et al., 2016) evaluates story understanding and script understanding, by requiring a system to choose the correct ending to a four-sentence story.
- *Commonsense Reasoning*: **PIQA** (Bisk et al., 2019) is a dataset designed for physical commonsense reasoning to investigate the physical knowledge of language models. **ARC** (Yadav et al., 2019) is a multiple-choice question-answering dataset, containing questions from science exams from grades 3-9. There are two partitioned datasets ARC-e (easy) and ARC-c (challenge), where the latter partition contains the more difficult questions that require reasoning. **OpenBookQA** (Mihaylov et al., 2018) is designed to test understanding of both the topic (*e.g.*, salient facts) and the language it is expressed in. This dataset contains questions that require multi-step reasoning, commonsense knowledge, and rich text comprehension.
- *Winograd-style tasks*: In the Winograd schema challenge, a *schema* is a pair of sentences that differ in only one or two words and that contain an ambiguity that is resolved in opposite ways in the two sentences. **Winograd** tasks (Kocijan et al., 2020) require world knowledge and reasoning to be solved. **WinoGrande** (Sakaguchi et al., 2020) is a large-scale dataset of 44k problems, and requires commonsense reasoning to choose the correct option for a given sentence.
- *Natural Language Understanding (NLU)*: **SuperGLUE** (Sarlin et al., 2020) consists of 8 challenging NLU tasks, including word sense disambiguation, natural language inference, coreference resolution, and question-answering.
- *Natural Language Inference* (NLI): **Adversarial NLI (ANIL)** (Nie et al., 2019) is collected via an adversarial human-and-model-in-the-loop procedure and is selected to be difficult to state-of-the-art models.

Baselines. The main baseline we compare with is PaLM (Chowdhery et al., 2022), since it is one of state-of-the-arts on a wide range of NLP benchmarks.

4.2 Main results

4.2.1 Few-shot Performance

We compare FCM with PaLM on few-shot and zero-shot performance in a wide range of benchmarks. Table 2 includes the results for the FCM 1B and the PaLM 1B models. The results averaged over task categories are presented in Figure 1. Following prior work, we only consider single checkpoint results from pre-trained language models. The few-shot and zero-shot results are averaged over three evaluation random seeds.

FCM outperforms PaLM on 17 out of 19 tasks in the zero-shot setting, 15 out of 19 tasks in the one-shot setting, and 15 out of 19 tasks in the few-shot setting. Consider that PaLM is well-tuned in many aspects, including the pre-training dataset, training strategy, and the number of tokens observed. The significantly better results of FCM shows that the training objective can also play a crucial role in the model performance.

	Zero-shot		One-shot		Few-shot	
Task	PaLM 1B	FCM 1B	PaLM 1B	FCM 1B	PaLM 1B	FCM 1B
Lambada (EM)	42.4	**43.5**	48.9	**49.5**	48.2	**49.7**
StoryCloze	**68.8**	68.2	**67.3**	66.9	65.9	**66.7**
PIQA	72.0	**72.1**	71.0	**71.6**	**72.0**	71.6
ARC-e	**46.2**	45.6	**48.0**	45.9	**50.2**	48.2
ARC-c	25.8	**27.7**	26.3	**27.2**	26.5	**28.1**
OpenbookQA	45.8	**46.4**	**45.0**	43.2	42.6	**43.6**
Winograd	67.0	**70.0**	67.0	**67.4**	64.8	**70.0**
Winogrande	54.0	**54.5**	54.0	**55.8**	53.6	**55.0**
BoolQ	45.9	**56.0**	48.3	**52.6**	**48.1**	46.8
Copa	72.0	**74.0**	72.0	**73.0**	70.0	**72.0**
RTE	50.9	**53.8**	53.1	**54.5**	**53.1**	45.1
WiC	51.4	**52.6**	**47.8**	46.9	48.9	**50.1**
Multirc (F1a)	35.2	**40.6**	57.1	**57.2**	**57.2**	48.2
WSC	65.3	**70.2**	66.7	**71.2**	66.7	**70.2**
ReCoRD	75.8	**76.3**	75.8	**76.4**	74.9	**75.0**
CB	48.2	**50.0**	44.6	**44.8**	42.3	**48.2**
ANLI R1	33.3	**33.5**	31.3	**33.0**	30.5	**32.5**
ANLI R2	32.8	**34.2**	30.5	**30.6**	32.5	**33.4**
ANLI R3	33.3	**33.6**	30.0	**31.2**	32.8	**34.2**

Table 2: Results obtained by the FCM 1B model across NLP benchmarks. We use the same setup as in Brown et al. (2020); Chowdhery et al. (2022), including the splits for each task.

4.2.2 Finetuning performance

We conduct finetuning experiments on the SuperGLUE benchmark to compare PaLM and FCM. Following PaLM experimental settings, models are finetuned with 5×10^{-5} learning rate using the Adafactor optimizer. To reduce computation time, we use batch size 512 instead of the original batch size 32 in PaLM. The models are finetuned for 20K steps.

Table 3 reports the *validation* results on finetuning on task-proportionate mixture of SuperGLUE tasks. On SuperGLUE, we compare with state-of-the-art models such as T5 11B (Raffel et al., 2020) and UL2 (Tay et al., 2022), as well as PaLM (Chowdhery et al., 2022) and show that FCM obtains significantly better performance than PaLM. All models are trained on C4 dataset, T5 11B and UL2 are trained on 1000B tokens, the rest of models are trained on 180B tokens. It is worth noting that both top performing models on SuperGLUE are encoder-decoder models that are trained using the span-corruption objective. It has been shown that such an architecture generally outperforms autoregressive decoder-only models on classification task finetuning, when training cost is equalized (Raffel et al., 2020). These results demonstrate that FCM can help bridge the gap.

Table 3: Finetuning results on SuperGLUE dev set. We compare with T5-11B (Raffel et al., 2020) and PaLM (Chowdhery et al., 2022). Scores reported are the peak validation scores per task.

Model	BoolQ	CB	CoPA	MultiRC	Record	RTE	WiC	WSC	Avg
T5 11B	90.8	94.9	98.0	87.4	93.8	93.9	77.3	96.2	89.9
UL2 20B	90.8	98.7	99.0	88.4	93.7	92.1	77.3	98.1	**90.7**
T5 1.4B	83.7	92.9	85.9	82.7	69.6	78	80.8	80	81.7
PaLM 1B	75.0	78.6	58.2	62.9	63.3	60.0	66.4	71.2	67.0
FCM 1B	73.5	82.0	58.0	63.0	65.1	61.0	66.5	80.8	**68.7**

4.2.3 Model scalability

To demonstrate the scalability of FCM, we further evaluate FCM with different model sizes in Table 4. We consider both a smaller model with 128 millions parameters, and a scaled-up model

with 8 billion parameters. All models are trained for 180 billion tokens, which is equivalent to about one epoch on the C4 dataset. Although 8 billion model size is relatively small compared to typical large language models (Chowdhery et al., 2022; Brown et al., 2020), these results serve as a proof-of-concept of FCM's effectiveness and scalability for larger model sizes. We leave further scaled-up experiments as promising future work. We compare with methods that use more high quality datasets including the official PaLM⋆ and GPT-3. PaLM is trained on 780 billion tokens and GPT-3 is trained on 300 billion tokens. Other baselines include T5 and UL2 which are trained for 1000 billion tokens and ST-MoE which uses 500 billion tokens. The results show that FCM works with both smaller and larger models, ranging from 128 million parameters to 8 billion parameters. Surprisingly, the largest FCM model matches the score of PaLM⋆ with an average score of 64, despite the fact that PaLM⋆ is trained on a much larger dataset (780B tokens) of high-quality conversation and webpage data, while FCM is trained on the smaller C4 dataset (180B tokens). Due to the high computation cost of training an 8B parameter model, we are not able to pre-train PaLM 8B on C4 within the time constraint. However, we further compare FCM with official PaLM 8B model on one-shot and few-shot experiments. Table 5 shows the results, FCM matches PaLM in most tasks, showing the promising capabilities of FCM.

Table 4: Comparisons on SuperGLUE zero-shot benchmark with different model sizes. PaLM⋆ denotes the published results of PaLM.

Model	BoolQ	CB	COPA	MultiRC	ReCORD	RTE	WiC	WSC	Avg
Methods that use more high quality data									
GPT-3 175B	60.5	46.4	91	72.9	90.2	63.5	0	65.4	61.2
PaLM⋆ 540B	88	51.8	93	83.5	92.9	72.9	59.1	89.1	78.8
GPT-3 13B	66.2	19.6	84	71.4	89	62.8	0	64.4	57.2
PaLM⋆ 8B	68.3	41.1	86	47.5	87.8	54.2	47	78.9	63.9
Methods that use 1000B tokens from C4									
ST-MoE 269B	40.8	41.1	56	30.3	50	52.7	50	57.5	47.3
T5-XXL 11B	44.3	37.5	70	23	85.8	48.8	50.9	59.3	52.5
UL2 20B	63.1	41.1	85	36.2	88.1	60.7	49.8	79.9	63
Methods that use 180B tokens from C4									
PaLM 128M	58.8	8.8	63	54.3	62.4	53.1	49	56.5	50.7
FCM 128M	55.7	8.9	69	55.1	62.1	56.3	52.1	56.1	51.9
PaLM 1B	45.9	48.2	72	35.2	75.8	50.9	51.6	65.3	55.6
FCM 1B	56	50	74	40.6	76.3	53.8	52.4	70.2	59.2
FCM 8B	62.1	48.2	84	54.1	85	48	51.1	79.1	**64**

Table 5: Comparison between FCM and PaLM on SuperGLUE zero-shot and few-shot benchmark tasks. PaLM⋆ denotes published results .

# of shots	Model	BoolQ	CB	COPA	MultiRC	ReCORD	RTE	WiC	WSC	Avg
Zero-shot	PaLM⋆ 8B	68.3	41.1	86	47.5	87.8	54.2	47	78.9	63.9
	FCM 8B	62.1	48.2	84	54.1	85	48	51.1	79.1	64
One-shot	PaLM⋆ 8B	64.7	41.1	82	50.6	87.8	57.8	47.3	81.4	64.1
	FCM 8B	59.6	51.5	83	56.5	84.9	47.3	46.9	86.8	64.6
Few-shot	PaLM⋆ 8B	68.9	57.1	82	41.1	88	56.7	52.4	83.2	66.2
	FCM 8B	57.7	50	85	46.5	83.9	48.4	49.5	87.9	63.6

4.3 Ablation Study

FCM works best with random ratio. We evaluate the impact of mask ratio on FCM using SuperGLUE zero-shot benchmark. Table 6 presents the results of FCM with different mask ratios. Among them, sampling random ratio between $[0.0, 0.15]$ performs significantly better than other

choices. Sampling mask ratios from 0.0 to 0.1 or 0.15 perform generally better than using fixed mask ratio 0.1 or 0.15, indicating that fixed mask ratios could potentially introduce pre-training and inference gap, and sampling random mask ratio is a simple way to alleviate it.

Table 6: Ablation of mask ratio on SuperGLUE. Comparison between fixed mask ratio and random mask ratios using FCM. The model size is 1B. FCM $[x, y]$ denotes mask ratio is randomly sampled between x and y.

Model	BoolQ	CB	COPA	MultiRC	ReCORD	RTE	WiC	WSC	Avg
				zero-shot					
PaLM	45.9	48.2	72.4	35.2	75.8	50.9	51.6	65.3	55.7
FCM [0.1, 0.1]	56.5	51.6	73.5	32.9	76.3	55.6	52	67.1	58.2
FCM [0.15, 0.15]	54	48.2	75.5	22.6	75.9	52.7	49.8	66.1	55.6
FCM [0, 0.1]	57.9	51.8	69.6	33.3	76.8	48.4	51.6	67.7	57.1
FCM [0.0, 0.15]	56	50	74.1	40.6	76.3	53.8	52.4	70.2	**59.2**
FCM [0.0, 0.3]	52.5	53.6	69.4	42.9	75.4	49.8	48.4	66.1	57.3
				one-shot					
PaLM	48.3	44.6	50.9	75.8	47.8	72	35.9	66.7	55.3
FCM [0.1, 0.1]	56.1	32.1	53.8	76.3	47.3	72	33.3	67.4	54.8
FCM [0.15, 0.15]	48.3	37.5	52.4	75.9	49.1	69	20.9	66.3	52.4
FCM [0, 0.1]	56.5	42.9	53.1	76.8	47.8	72	29.1	66.3	55.6
FCM [0.0, 0.15]	52.6	44.6	54.5	76.4	46.9	73	43.2	71.6	**57.9**
FCM [0.0, 0.3]	50.8	32.1	52	75.4	47.2	74	46.5	66.3	55.5

Comparison with dropout. Random masking can be seen as a special type of dropout applied only on the input sequence. We note that general dropout and FCM are complementary in that they can be combined together. To compare random masking vs. dropout, we compare three models in Table 7: (1) PaLM, (2) PaLM with dropout rate 0.1, (3) FCM with fixed random ratio 0.1, and (4) FCM with fixed random ratio 0.1 and dropout rate 0.1. We see that using dropout during large language models pre-training is harmful, which aligns with findings from prior work (Raffel et al., 2020; Chowdhery et al., 2022). Similarly, apply dropout to FCM also decreases performance. In contrast, FCM random masking improves performance over PaLM, showing the effectiveness of performing the next token prediction task with randomly selected past tokens masked out.

Table 7: Comparison between Random Masking vs. Dropout. The model size is 1B.

Model	BoolQ	CB	COPA	MultiRC	ReCORD	RTE	WiC	WSC	Avg
PaLM	45.9	48.2	72.4	35.2	75.8	50.9	51.6	65.3	55.7
PaLM + Dropout	53.5	48.2	64.4	37.2	75.7	50.2	50.2	63.5	55.4
FCM [0.1, 0.1] + Dropout	44	53.6	71	43.1	75.3	59.2	49.8	65.4	57.7
FCM [0.1, 0.1]	56.5	51.6	73.5	32.9	76.3	55.6	52	67.1	**58.2**

5 Conclusion

In this paper, we propose FCM, a novel pre-training paradigm using a causal transformer decoder. FCM is a combination of causal next-token-prediction and random masking to input sequence. Experimental results show that FCM significantly outperforms the state-of-the-art causal transformer model on a wide range of zero- and few-shot as well as finetuning benchmarks, and our model is readily extendable to various tasks.

As FCM improves performance of causal language models on few-shot and finetuning benchmarks, applying our approach to other language understanding tasks and language-image tasks (*e.g.*, Flamingo (Alayrac et al., 2022)) is a promising direction for future work. Since our method does not introduce extra computation, another direction would be investigating what is the impact of FCM on compute scaling law (Hoffmann et al., 2022).

REFERENCES

Armen Aghajanyan, Bernie Huang, Candace Ross, Vladimir Karpukhin, Hu Xu, Naman Goyal, Dmytro Okhonko, Mandar Joshi, Gargi Ghosh, Mike Lewis, and Luke Zettlemoyer. Cm3: A causal masked multimodal model of the internet. *arXiv preprint arXiv: Arxiv-2201.07520*, 2022.

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, and Mengyuan Yan. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv: Arxiv-2204.01691*, 2022.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. Ext5: Towards extreme multi-task scaling for transfer learning. *arXiv preprint arXiv: Arxiv-2111.10952*, 2021.

Mikel Artetxe, Jingfei Du, Naman Goyal, Luke Zettlemoyer, and Ves Stoyanov. On the role of bidirectionality in language model pre-training. *arXiv preprint arXiv: Arxiv-2205.11726*, 2022.

Hangbo Bao, Li Dong, Furu Wei, Wenhui Wang, Nan Yang, Xiaodong Liu, Yu Wang, Songhao Piao, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unilmv2: Pseudo-masked language models for unified language model pre-training. *arXiv preprint arXiv: Arxiv-2002.12804*, 2020.

Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. Efficient training of language models to fill in the middle. *arXiv preprint arXiv: Arxiv-2207.14255*, 2022.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. *arXiv preprint arXiv: Arxiv-1911.11641*, 2019.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, Michael Pieler, USVSN Sai Prashanth, Shivanshu Purohit, Laria Reynolds, Jonathan Tow, Ben Wang, and Samuel Weinbach. Gpt-neox-20b: An open-source autoregressive language model. *BIGSCIENCE*, acl.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL `http://github.com/google/jax`.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code. *arXiv preprint arXiv: Arxiv-2107.03374*, 2021.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Chris Donahue, Mina Lee, and Percy Liang. Enabling language models to fill in the blanks. *Acl*, 2020. doi: 10.18653/V1/2020.ACL-MAIN.225.

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (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. 13042–13054, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/c20bb2d9a50d5ac1f713f8b34d9aac5a-Abstract.html.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. GLM: general language model pretraining with autoregressive blank infilling. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 320–335. Association for Computational Linguistics, 2022. URL https://aclanthology.org/2022.acl-long.26.

Daniel Fried, Armen Aghajanyan, Jessy Lin, Sida Wang, Eric Wallace, Freda Shi, Ruiqi Zhong, Wen tau Yih, Luke Zettlemoyer, and Mike Lewis. Incoder: A generative model for code infilling and synthesis. *arXiv preprint arXiv: Arxiv-2204.05999*, 2022.

Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2020. URL http://github.com/google/flax.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. Training compute-optimal large language models. *arXiv preprint arXiv: Arxiv-2203.15556*, 2022.

Vid Kocijan, Thomas Lukasiewicz, Ernest Davis, Gary Marcus, and Leora Morgenstern. A review of winograd schema challenge datasets and approaches. *arXiv preprint arXiv: Arxiv-2004.13831*, 2020.

Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018: System Demonstrations, Brussels, Belgium, October 31 - November 4, 2018*, pp. 66–71. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-2012. URL https://doi.org/10.18653/v1/d18-2012.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pp. 3045–3059. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.243. URL https://doi.org/10.18653/v1/2021.emnlp-main.243.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*, 2019.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *arXiv preprint arXiv:2206.14858*, 2022.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*, 2021.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 158–167, Vancouver, Canada, jul 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1015. URL https://aclanthology.org/P17-1015.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *acm computing surveys*, 2021. doi: 10.1145/3560815.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? A new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 2381–2391. Association for Computational Linguistics, 2018. doi: 10.18653/v1/d18-1260. URL https://doi.org/10.18653/v1/d18-1260.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 3470–3487. Association for Computational Linguistics, 2022. URL https://aclanthology.org/2022.acl-long.244.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and evaluation framework for deeper understanding of commonsense stories. *arXiv preprint arXiv: Arxiv-1604.01696*, 2016.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial nli: A new benchmark for natural language understanding. *arXiv preprint arXiv: Arxiv-1910.14599*, 2019.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Quan Ngoc Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. *arXiv preprint arXiv: Arxiv-1606.06031*, 2016.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL*, 2018.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 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.

Adam Roberts, Colin Raffel, Katherine Lee, Michael Matena, Noam Shazeer, Peter J Liu, Sharan Narang, Wei Li, and Yanqi Zhou. Exploring the limits of transfer learning with a unified text-to-text transformer. 2019.

Adam Roberts, Hyung Won Chung, Anselm Levskaya, Gaurav Mishra, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, Curtis Hawthorne, Aitor Lewkowycz, Alex Salcianu, Marc van Zee, Jacob Austin, Sebastian Goodman, Livio Baldini

Soares, Haitang Hu, Sasha Tsvyashchenko, Aakanksha Chowdhery, Jasmijn Bastings, Jannis Bulian, Xavier Garcia, Jianmo Ni, Andrew Chen, Kathleen Kenealy, Jonathan H. Clark, Stephan Lee, Dan Garrette, James Lee-Thorp, Colin Raffel, Noam Shazeer, Marvin Ritter, Maarten Bosma, Alexandre Passos, Jeremy Maitin-Shepard, Noah Fiedel, Mark Omernick, Brennan Saeta, Ryan Sepassi, Alexander Spiridonov, Joshua Newlan, and Andrea Gesmundo. Scaling up models and data with `t5x` and `seqio`. *arXiv preprint arXiv:2203.17189*, 2022. URL `https://arxiv.org/abs/2203.17189`.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 8732–8740. AAAI Press, 2020. URL `https://ojs.aaai.org/index.php/AAAI/article/view/6399`.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=9Vrb9D0WI4`.

Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 4937–4946. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.00499. URL `https://openaccess.thecvf.com/content_CVPR_2020/html/Sarlin_SuperGlue_Learning_Feature_Matching_With_Graph_Neural_Networks_CVPR_2020_paper.html`.

Noam Shazeer. Fast transformer decoding: One write-head is all you need. *arXiv preprint arXiv: Arxiv-1911.02150*, 2019.

Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv: Arxiv-2002.05202*, 2020.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. *arXiv preprint arXiv: Arxiv-1804.04235*, 2018.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv: Arxiv-2104.09864*, 2021.

Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Dara Bahri, Tal Schuster, Huaixiu Steven Zheng, Neil Houlsby, and Donald Metzler. Unifying language learning paradigms. *arXiv preprint arXiv: Arxiv-2205.05131*, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`.

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`, May 2021.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. What language model architecture and pretraining objective work best for zero-shot generalization? *arXiv preprint arXiv: Arxiv-2204.05832*, 2022.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022a. URL https://openreview.net/forum?id=gEZrGCozdqR.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022b.

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 2578–2589. Association for Computational Linguistics, 2019. doi: 10.18653/v1/D19-1260. URL https://doi.org/10.18653/v1/D19-1260.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv: Arxiv-1906.08237*, 2019.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *arXiv preprint arXiv:2206.10789*, 2022.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv: Arxiv-2205.01068*, 2022.

Wanrong Zhu, Zhiting Hu, and Eric Xing. Text infilling. *arXiv preprint arXiv: Arxiv-1901.00158*, 2019.

A APPENDIX

A.1 IMPLEMENTATION AND TRAINING DETAILS

Our implementation uses Flax (Heek et al., 2020), JAX (Bradbury et al., 2018) and T5X (Roberts et al., 2022) Our architecture is based on PaLM (Chowdhery et al., 2022) which introduces some modifications to GPT-3 (Brown et al., 2020) architecture to reduce compute cost. The dataset C4 is provided by Tensorflow datasets. We use SentencePiece (Kudo & Richardson, 2018) as tokenizer. For PaLM and FCM trained on C4 datasets, the sequence length is 1024 to reduce compute cost, although the official PaLM is trained with longer context length 2048. Following the settings of PaLM, input examples are concatenated together and then split into sequences of exactly 1024 tokens, so that there are no padding tokens, but examples may be split in the middle. Input examples are differentiated from one another with a special `[eod]` token For downstream tasks evaluation, including both fewshot and finetune benchmarks, we follow the dataset format and splits used in Brown et al. (2020); Chowdhery et al. (2022). For 128M models, training 180B tokens takes 18 hours on TPU v4-64. For 1B models, training on 180B tokens takes 25 hours on TPU v4-256. The training of 8B models on 180B tokens takes 100 hours on TPU v4-512.

A.2 HYPERPARAMETERS

In this section we provide the training and evaluation hyperparameters of FCM. These configurations follow the training hyperparameters of PaLM Chowdhery et al. (2022).

Hyperparameter	**Value**
Dropout	0.0
Optimizer	Adafactor
Initial learning rate lr	0.01
Learning rate decay	0.01 for the first 10,000 steps, which is then decayed at a rate of $1/\sqrt{k}$, where k is the step number
Weight decay	lr^2
Optimizer momentum	$\beta_1 = 0.9, \beta_2 = 1.0 - k^{-0.8}$
Global norm gradient clipping	1.0
Batch size	1024
Sequence length	1024

Table 8: Hyperparameters for training PaLM and FCM

Hyperparameter	**Value**
Dropout	0.1
Optimizer	SGD momentum
Momentum	0.9
Batch size	512
Sequence length	1024

Table 9: Hyperparameters for finetuning PaLM and FCM