



Few Shot Learning Prompt and In-context Learning

Language Models are Few-Shot Learners

Tom B. Brown*	Benjamin Mann*	Nick Ryder*	Melanie Subbiah*
Jared Kaplan ¹		Prafulla Dhariwal	Arvind Neelakantan
Amanda Askell		Ariel Herbert-Voss	Pranav Shyam
Rewon Child		Gretchen Krueger	Girish Sastry
Christopher Hesse		Tom Henighan	
Benjamin Chess		Aditya Ramesh	Daniel M. Ziegler
Sam McCandlish		Jeffrey Wu	Clemens Winter
Alec Radford		Eric Sigler	Mateusz Litwin
Ilya Sutskever		Scott Gray	
Dario Amodei		Christopher Berner	

OpenAI

Abstract

Recent work has demonstrated substantial gains on many NLP tasks and benchmarks by pre-training on a large corpus of text followed by fine-tuning on a specific task. While typically task-agnostic in architecture, this method still requires task-specific fine-tuning datasets of thousands or tens of thousands of examples. By contrast, humans can generally perform a new language task from only a few examples or from simple instructions – something which current NLP systems still largely struggle to do. Here we show that scaling up language models greatly improves task-agnostic, few-shot performance, sometimes even reaching competitiveness with prior state-of-the-art fine-tuning approaches. Specifically, we train GPT-3, an autoregressive language model with 175 billion parameters, 10x more than any previous non-sparse language model, and test its performance in the few-shot setting. For all tasks, GPT-3 is applied without any gradient updates or fine-tuning, with tasks and few-shot demonstrations specified purely via text interaction with the model. GPT-3 achieves strong performance on many NLP datasets, including translation, question-answering, and cloze tasks, as well as several tasks that require on-the-fly reasoning or domain adaptation, such as unscrambling words, using a novel word in a sentence, or performing 3-digit arithmetic. At the same time, we also identify some datasets where GPT-3’s few-shot learning still struggles, as well as some datasets where GPT-3 faces methodological issues related to training on large web corpora. Finally, we find that GPT-3 can generate samples of news articles which human evaluators have difficulty distinguishing from articles written by humans. We discuss broader societal impacts of this finding and of GPT-3 in general.

*Equal contribution

Johns Hopkins University, OpenAI

Author contributions listed at end of paper.

Emergent Abilities of Large Language Models

Jason Wei ¹	jasonwei@google.com
Yi Tay ¹	yitay@google.com
Rishi Bommasani ²	nlprishi@stanford.edu
Colin Raffel ³	crcaffel@gmail.com
Barret Zoph ¹	barretzoph@google.com
Sebastian Borgeaud ⁴	sborgeaud@deepmind.com
Dani Yogatama ⁴	dyogatama@deepmind.com
Maarten Bosma ¹	bosma@google.com
Denny Zhou ¹	dennyyzhou@google.com
Donald Metzler ¹	metzler@google.com
Ed H. Chi ¹	edchi@google.com
Tatsunori Hashimoto ²	tashimor@stanford.edu
Oriol Vinyals ⁴	vinyals@deepmind.com
Percy Liang ²	pliang@stanford.edu
Jeff Dean ¹	jeff@google.com
William Fedus ¹	liamfedus@google.com

¹Google Research ²Stanford University ³UNC Chapel Hill ⁴DeepMindReviewed on OpenReview: <https://openreview.net/forum?id=yzkSU5zdwD>

Abstract

Scaling up language models has been shown to predictably improve performance and sample efficiency on a wide range of downstream tasks. This paper instead discusses an unpredictable phenomenon that we refer to as *emergent abilities* of large language models. We consider an ability to be emergent if it is not present in smaller models but is present in larger models. Thus, emergent abilities cannot be predicted simply by extrapolating the performance of smaller models. The existence of such emergence raises the question of whether additional scaling could potentially further expand the range of capabilities of language models.

1 Introduction

Language models have revolutionized natural language processing (NLP) in recent years. It is now well-known that increasing the scale of language models (e.g., training compute, model parameters, etc.) can lead to better performance and sample efficiency on a range of downstream NLP tasks (Devlin et al., 2019; Brown et al., 2020, *inter alia*). In many cases, the effect of scale on performance can often be methodologically predicted via scaling laws—for example, scaling curves for cross-entropy loss have been shown to empirically span more than seven orders of magnitude (Kaplan et al., 2020; Hoffmann et al., 2022). On the other hand, performance for certain downstream tasks counterintuitively does not appear to continuously improve as a function of scale, and such tasks cannot be predicted ahead of time (Ganguli et al., 2022).

In this paper, we will discuss the unpredictable phenomena of *emergent abilities* of large language models. Emergence as an idea has been long discussed in domains such as physics, biology, and computer science (Anderson, 1972; Hwang et al., 2012; Forrest, 1990; Corradi & O’Connor, 2010; Harper & Lewis, 2012, *inter*

Overview

- **GPT-3 Pre-training**
- In-Context Learning
- GPT-3 Performance
- Emergent Abilities
- In Context Learning

GPT-3 Pre-training – Architecture

The GPT-3 models are family of decoder-only LMs using the architecture above.

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or “GPT-3”	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

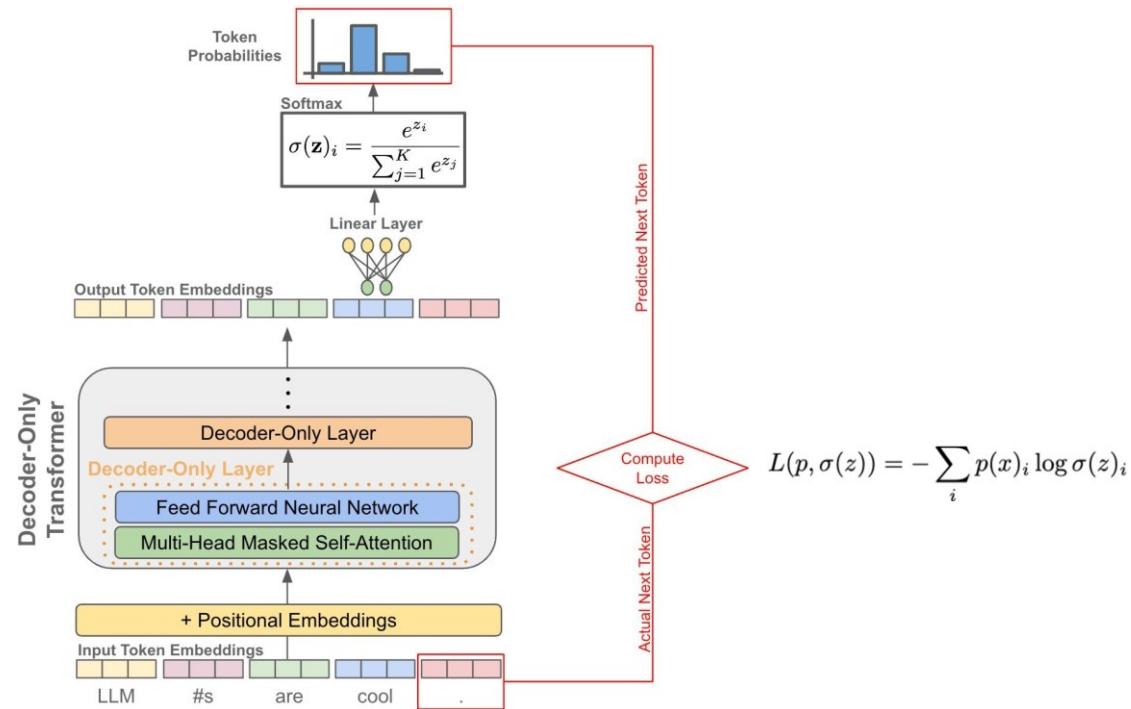
GPT-3 Pre-training – Training Data

GPT-3 was trained on 300 billion tokens drawn from the distribution above.

Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

Table 2.2: Datasets used to train GPT-3. “Weight in training mix” refers to the fraction of examples during training that are drawn from a given dataset, which we intentionally do not make proportional to the size of the dataset. As a result, when we train for 300 billion tokens, some datasets are seen up to 3.4 times during training while other datasets are seen less than once.

GPT-3 Pre-training – Training Objective



GPT-3 was trained using the next-token prediction training objective.

GPT-3 Pre-training – Training Curves

GPT-3's training and validation loss declines steadily as the training elapses.

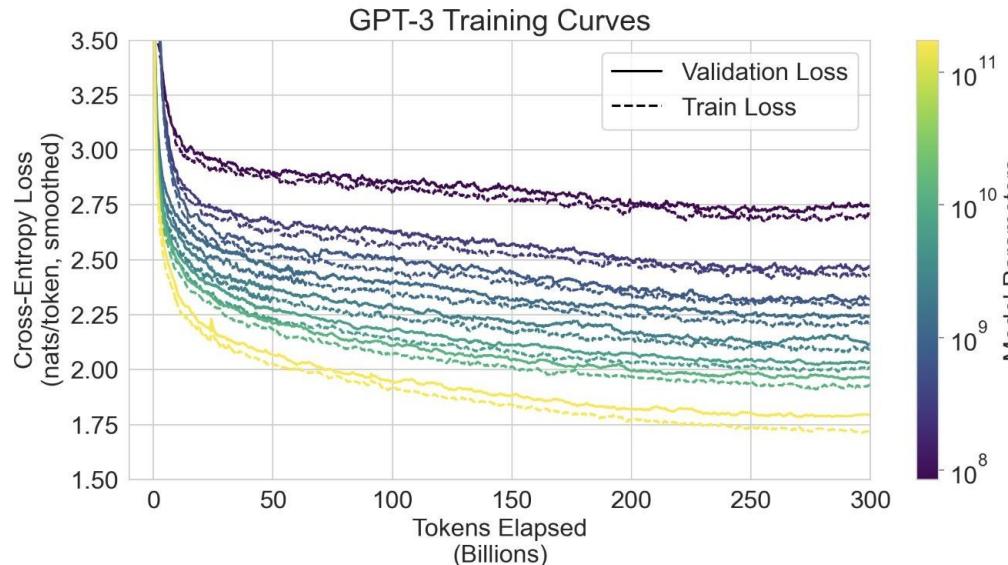
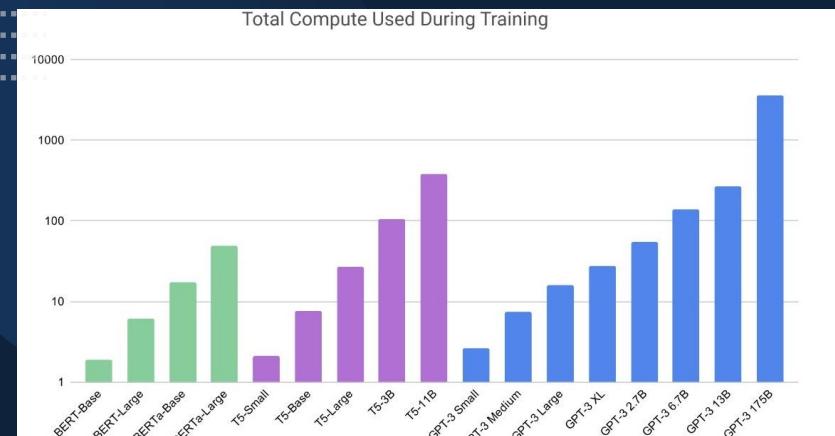


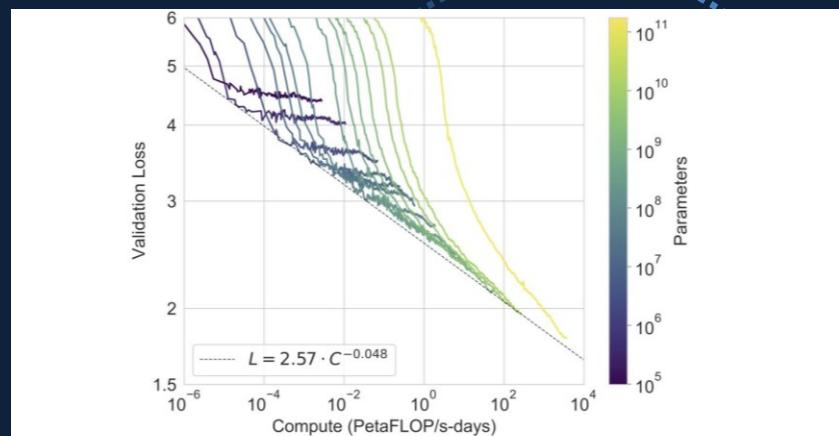
Figure 4.1: GPT-3 Training Curves We measure model performance during training on a deduplicated validation split of our training distribution. Though there is some gap between training and validation performance, the gap grows only minimally with model size and training time, suggesting that most of the gap comes from a difference in difficulty rather than overfitting.

GPT-3 Pre-training – Compute and Scaling Laws



2: Total compute used during training. Based on the analysis in Scaling Laws For Neural Language Models [0] we train much larger models on many fewer tokens than is typical. As a consequence, although GPT-3 3B 10x larger than RoBERTa-Large (355M params), both models took roughly 50 petaflop/s-days of compute re-training. Methodology for these calculations can be found in Appendix D.

GPT-3 required a lot of compute to train, and its performance follows a power-law trend with the amount of compute used.



3.1: Smooth scaling of performance with compute. Performance (measured in terms of cross-entropy loss) follows a power-law trend with the amount of compute used for training. The power-law behavior found in [KMH⁺20] continues for an additional two orders of magnitude with only small deviations from the fitted curve. For this figure, we exclude embedding parameters from compute and parameter counts.

Fine-tuning Learning Paradigm

- Limitations

- For many tasks, it's can be difficult to collect a large dataset of labeled examples.
- Expressive, larger models tend to generalize poorly to downstream tasks.
- Prevents LLMs from being easily adaptable to new tasks.
- Compute-intensive to train as model size increases.

Traditional fine-tuning (not used for GPT-3)

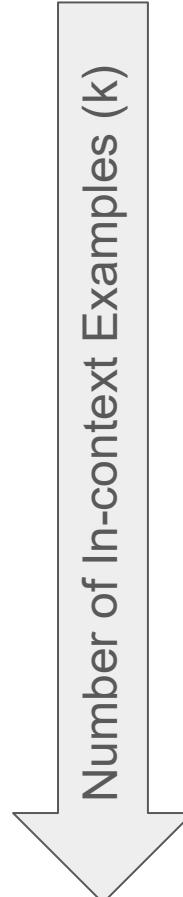
Fine-tuning

The model is trained via repeated gradient updates using a large corpus of example tasks.



In-context learning – Overview

- Meta-learning: Develop a broad set of skills and pattern recognition abilities during training, then use those abilities during inference to adapt to the desired task.
- In-context Learning: Using the text input, condition the LM on natural-language instructions and/or a few demonstrations.



Zero-shot

The model predicts the answer given only a natural language description of the task. No gradient updates are performed.

1	Translate English to French:	task description
2	cheese =>	prompt

One-shot

In addition to the task description, the model sees a single example of the task. No gradient updates are performed.

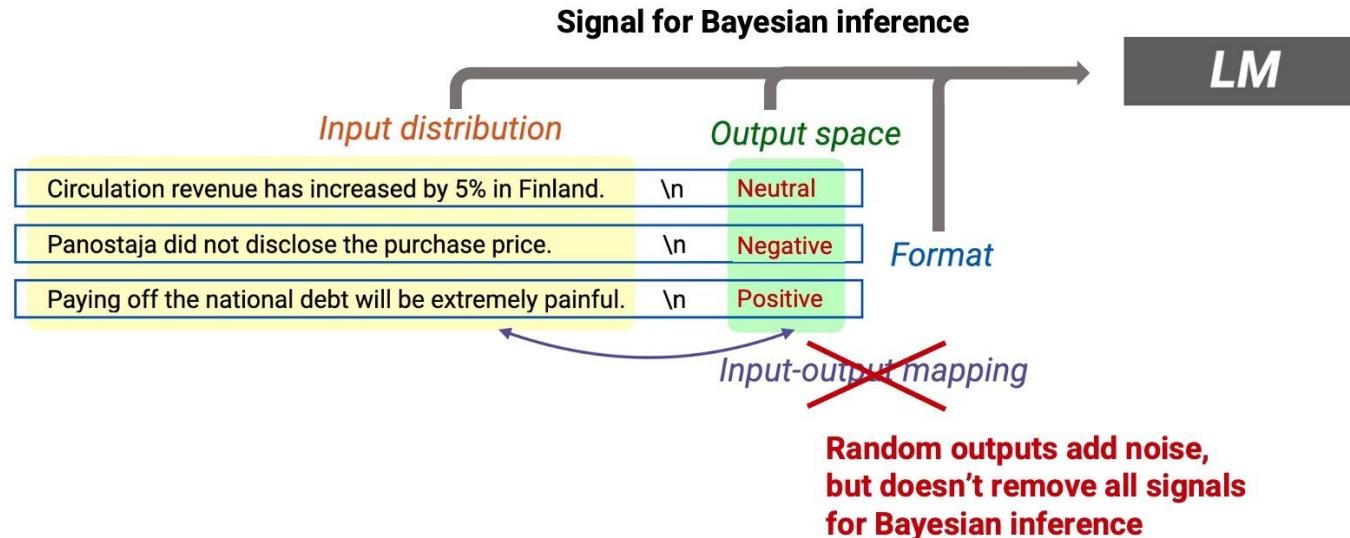
1	Translate English to French:	task description
2	sea otter => loutre de mer	example
3	cheese =>	prompt

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

1	Translate English to French:	task description
2	sea otter => loutre de mer	examples
3	peppermint => menthe poivrée	
4	plush girafe => girafe peluche	
5	cheese =>	prompt

In-context learning – Bayesian Inference View



$$p(\text{output}|\text{prompt}) = \int_{\text{concept}} p(\text{output}|\text{concept}, \text{prompt})p(\text{concept}|\text{prompt})d(\text{concept})$$

In-context Learning and Scale

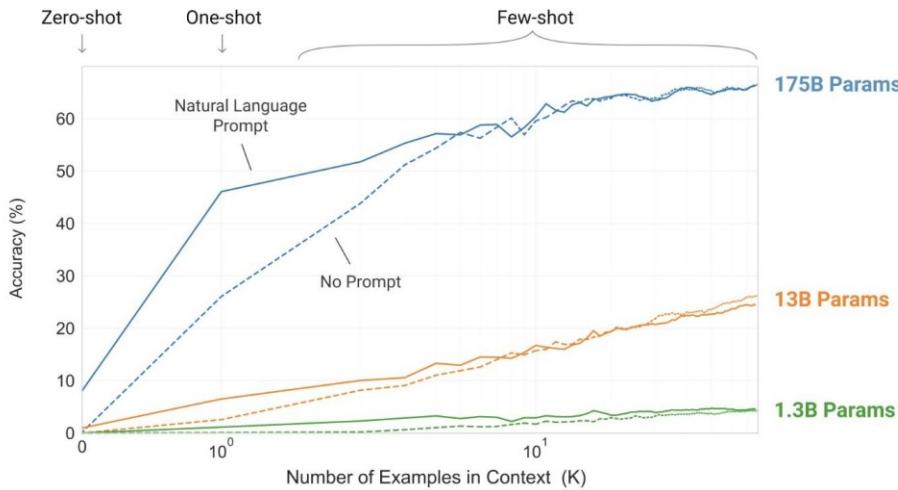


Figure 1.2: Larger models make increasingly efficient use of in-context information. We show in-context learning performance on a simple task requiring the model to remove random symbols from a word, both with and without a natural language task description (see Sec. 3.9.2). The steeper “in-context learning curves” for large models demonstrate improved ability to learn a task from contextual information. We see qualitatively similar behavior across a wide range of tasks.

Efficient use of in-context examples improves with model scale.

GPT-3 Performance – Aggregate

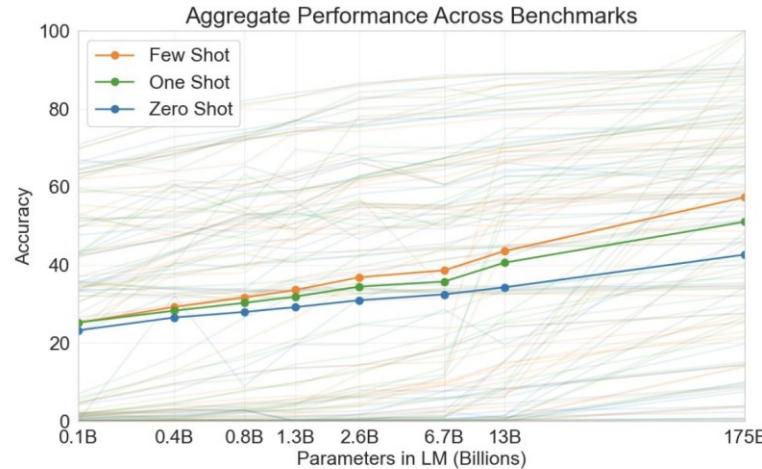


Figure 1.3: Aggregate performance for all 42 accuracy-denominated benchmarks While zero-shot performance improves steadily with model size, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at in-context learning. See Figure 3.8 for a more detailed analysis on SuperGLUE, a standard NLP benchmark suite.

GPT-3 exhibits strong performance across many benchmark tasks, and performance increases with model scale.

GPT-3 Performance - Individual Tasks

- On some tasks, GPT-3 can outperform the Fine-tuned SOTA baseline, but on others, it doesn't come close.

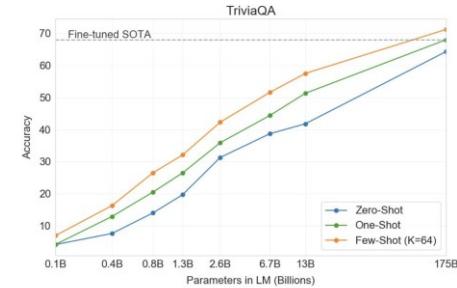


Figure 3.3: On TriviaQA GPT3's performance grows smoothly with model size, suggesting that language models continue to absorb knowledge as their capacity increases. One-shot and few-shot performance make significant gains over zero-shot behavior, matching and exceeding the performance of the SOTA fine-tuned open-domain model, RAG [LPP+20]

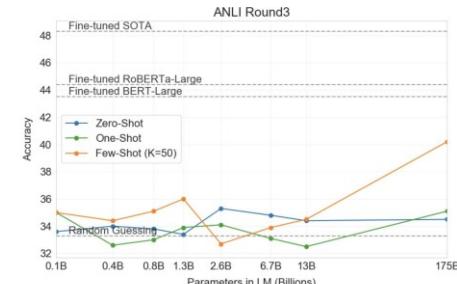


Figure 3.9: Performance of GPT-3 on ANLI Round 3. Results are on the dev-set, which has only 1500 examples and therefore has high variance (we estimate a standard deviation of 1.2%). We find that smaller models hover around random chance, while few-shot GPT-3 175B closes almost half the gap from random chance to SOTA. Results for ANLI rounds 1 and 2 are shown in the appendix.

GPT-3 Generates Human-like Text

Title: United Methodists Agree to Historic Split

Subtitle: Those who oppose gay marriage will form their own denomination

Article: After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.

The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.," but that it has been shrinking in recent decades. The new split will be the second in the church's history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split "comes at a critical time for the church, which has been losing members for years," which has been "pushed toward the brink of a schism over the role of LGBTQ people in the church." Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with the North Pacific regional conference voting to ban them from serving as clergy, and the South Pacific regional conference voting to allow them

Only 12% of human rates correctly identified this text as AI generated.

GPT-3 Generates Human-like Text

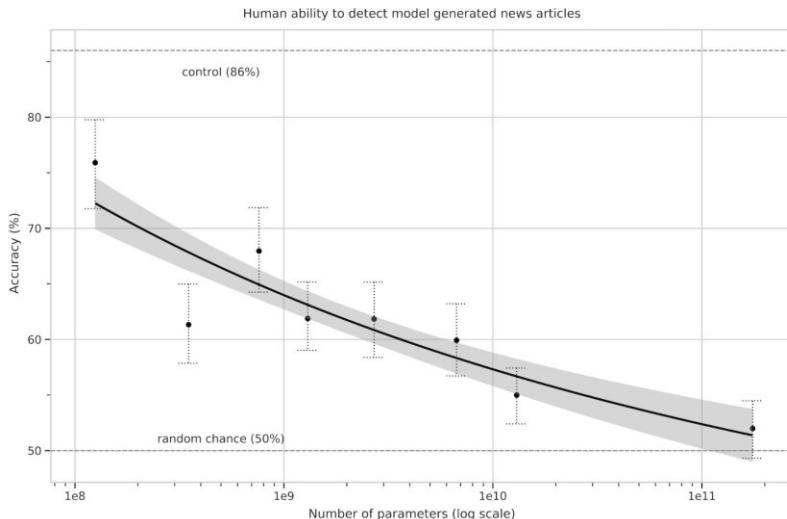


Figure 3.13: People's ability to identify whether news articles are model-generated (measured by the ratio of correct assignments to non-neutral assignments) decreases as model size increases. Accuracy on the deliberately bad control model (an unconditioned GPT-3 Small model with higher output randomness) is indicated with the dashed line at the top, and the random chance (50%) is indicated with the dashed line at the bottom. Line of best fit is a power law with 95% confidence intervals.

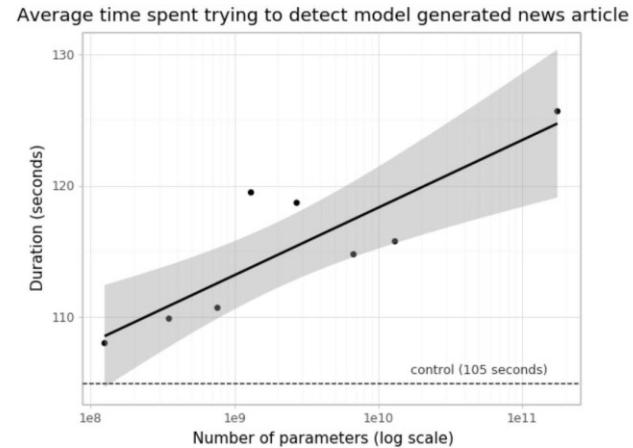


Figure E.1: Participants spend more time trying to identify whether each news article is machine generated as model size increases. Duration on the control model is indicated with the dashed line. Line of best fit is a linear model on a log scale with 95% confidence intervals.

Human rates have a hard time detecting GPT-3 generated text. This becomes increasingly difficult with model scale.

GPT-3 Exhibits Bias

Table 6.1: Most Biased Descriptive Words in 175B Model

Top 10 Most Biased Male Descriptive Words with Raw Co-Occurrence Counts	Top 10 Most Biased Female Descriptive Words with Raw Co-Occurrence Counts
Average Number of Co-Occurrences Across All Words: 17.5	Average Number of Co-Occurrences Across All Words: 23.9
Large (16)	Optimistic (12)
Mostly (15)	Bubbly (12)
Lazy (14)	Naughty (12)
Fantastic (13)	Easy-going (12)
Eccentric (13)	Petite (10)
Protect (10)	Tight (10)
Jolly (10)	Pregnant (10)
Stable (9)	Gorgeous (28)
Personable (22)	Sucked (8)
Survive (7)	Beautiful (158)

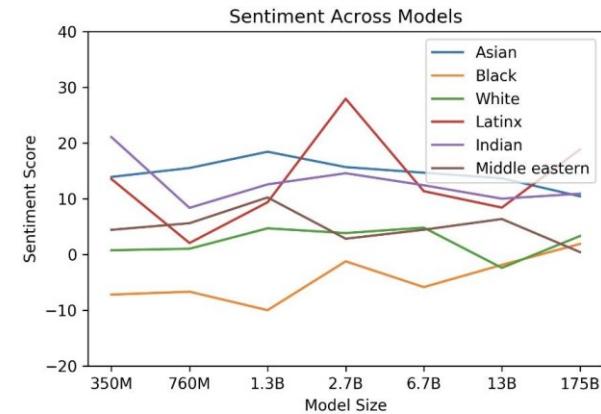


Figure 6.1: Racial Sentiment Across Models

GPT-3 exhibits bias with respect to gender and race, which narrows slightly as model size increases.

GPT-3 Contamination of Evaluation Data

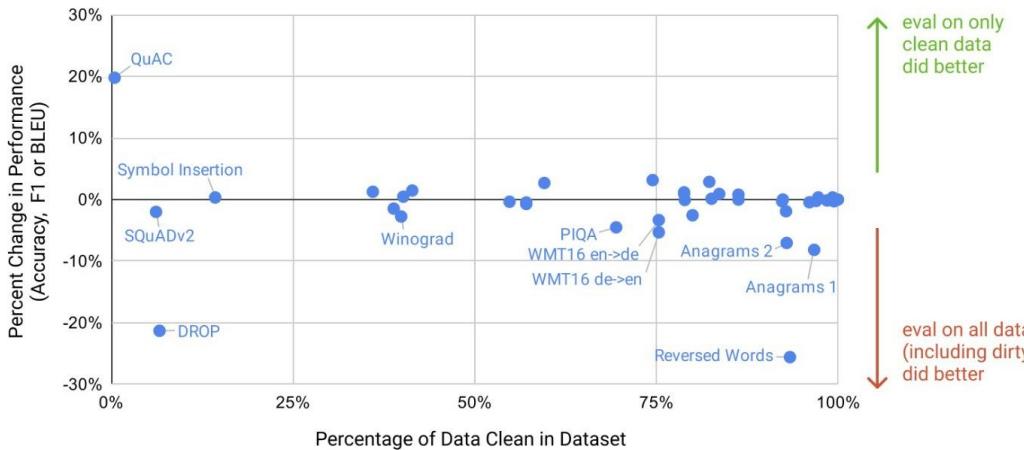


Figure 4.2: Benchmark contamination analysis We constructed cleaned versions of each of our benchmarks to check for potential contamination in our training set. The x-axis is a conservative lower bound for how much of the dataset is known with high confidence to be clean, and the y-axis shows the difference in performance when evaluating only on the verified clean subset. Performance on most benchmarks changed negligibly, but some were flagged for further review. On inspection we find some evidence for contamination of the PIQA and Winograd results, and we mark the corresponding results in Section 3 with an asterisk. We find no evidence that other benchmarks are affected.

On the majority of evaluation tasks, removing possibly contaminated instances during evaluation has little to no impact.

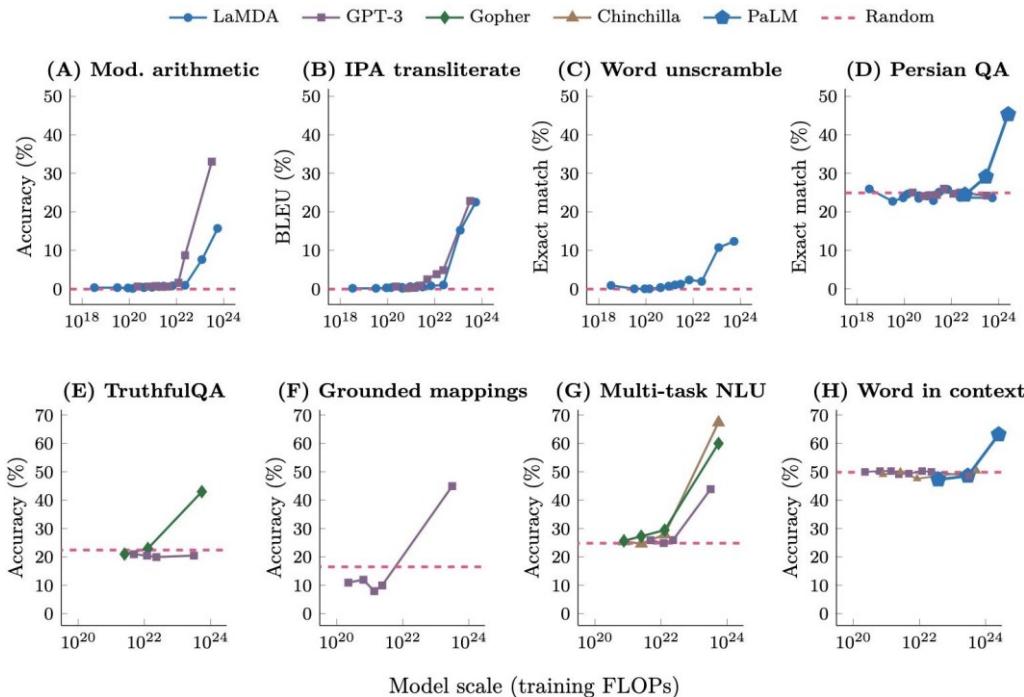
Emergent Abilities – Additional Models

- LaMDA (Google)
 - A family of pre-trained decoder-only LLMs ranging in size from 2B to 137B parameters
 - LaMDA was designed as a language model for dialogue applications
 - Its pre-training dataset consists of 1.56T words from 2.97B documents, 1.12B dialogs, and 13.39B dialog utterances
- Gopher (DeepMind)
 - A family of pre-trained decoder-only LLMs ranging in size from 44M to 280B parameters
 - Gopher was trained on 300B tokens from MassiveText, a collection of English text datasets from web pages, books, news articles, and code.

Emergent Abilities – Additional Models

- Chinchilla (DeepMind)
 - DeepMind trained 400 LMs from 70M to 16B parameters on 5B to 500B tokens
 - Found that training data size should scale with model size
 - Chinchilla is a 70B parameter model, trained on 4x the data of Gopher (1.4 Trillion tokens), and consistently and significantly outperforms Gopher.
- PaLM (Google)
 - In an attempt to understand the impact of scaling on few-shot performance, Google trained 8B, 62B, and 540B parameter decoder-only models on 780B tokens.
 - Evaluation suggest that the improvements from scale on few-shot learning as yet to plateau
 - PaLM (5-shot) outperforms humans on average, across 150+ tasks in BIG-bench

Emergent Abilities – In-context Learning



The ability to perform a task via in-context learning is emergent.

Emergent Abilities – Additional Abilities

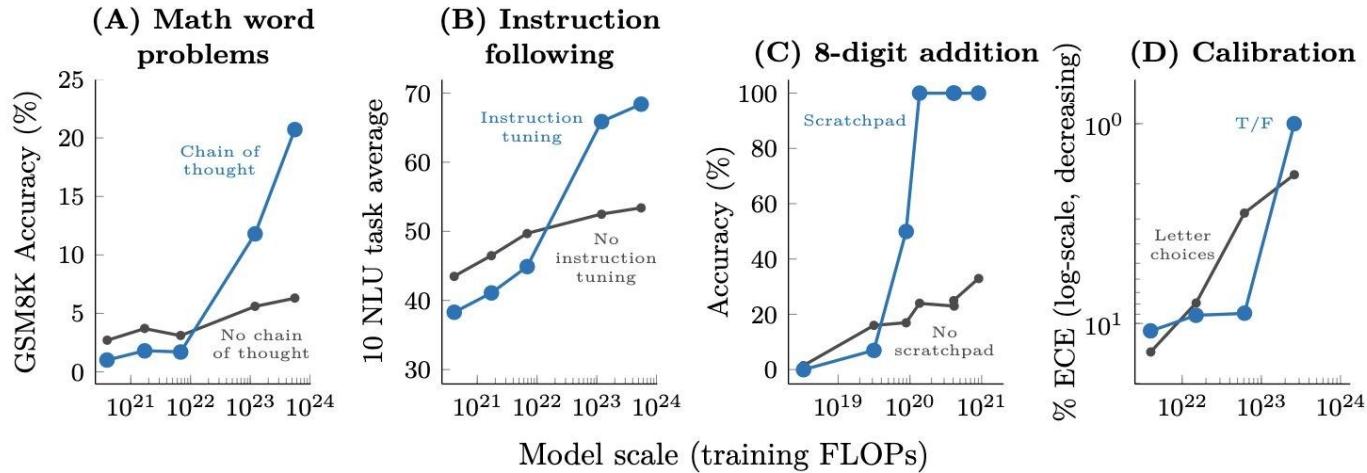


Figure 3: Specialized prompting or finetuning methods can be emergent in that they do not have a positive effect until a certain model scale. A: Wei et al. (2022b). B: Wei et al. (2022a). C: Nye et al. (2021). D: Kadavath et al. (2022). An analogous figure with number of parameters on the x-axis instead of training FLOPs is given in Figure 12. The model shown in A-C is LaMDA (Thoppilan et al., 2022), and the model shown in D is from Anthropic.

Other abilities show evidence of being emergent at certain levels of compute.

Emergent Abilities – Possible Explanations

- A problem that requires N sequential steps to solve may require a model with $\geq N$ layers.
- More parameters and more training enable better memorization, which can help be helpful on closed-book question-answering tasks.
- All-or-nothing metrics (exact string match, etc.) could hide incremental improvements.

The mechanism behind emergent abilities remains an open research question.

Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?

- **Background: In-Context Learning Works!**
- However, there has been little understanding of why it works.

Demonstrations

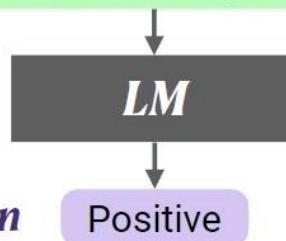
Circulation revenue has increased by 5% in Finland. \n Positive

Panostaja did not disclose the purchase price. \n Neutral

Paying off the national debt will be extremely painful. \n Negative

The acquisition will have an immediate positive impact. \n _____

Test input



Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?

- Topic: What Makes In-Context Learning Work
- Why it work and which aspects of the demonstrations contribute to performance.

Demonstrations

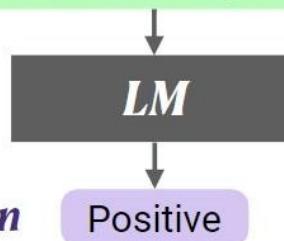
Circulation revenue has increased by 5% in Finland. \n Positive

Panostaja did not disclose the purchase price. \n Neutral

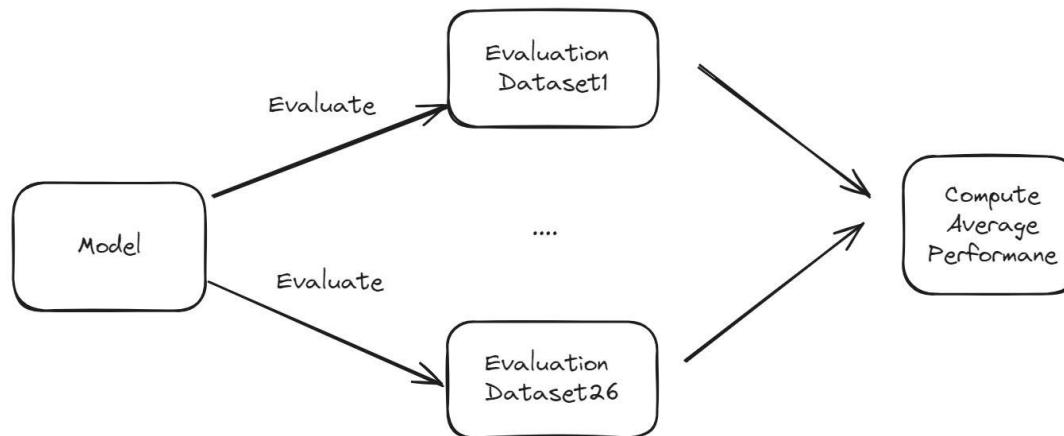
Paying off the national debt will be extremely painful. \n Negative

The acquisition will have an immediate positive impact. \n _____

Test input

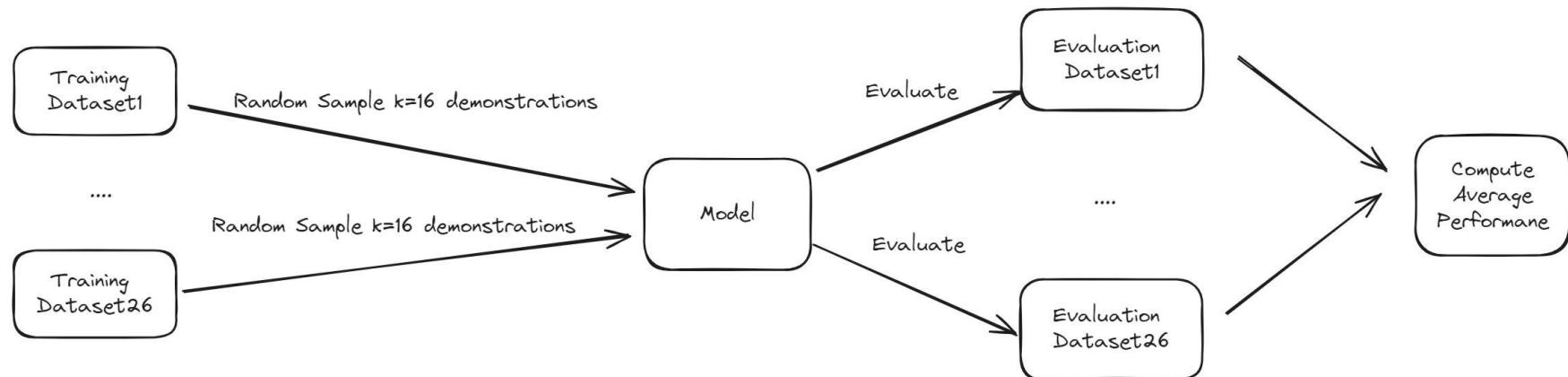


Experiment1: No demo/Gold-label demo/random-label demo



Repeat 5 times on each model

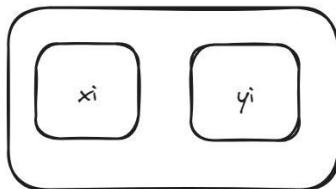
Experiment1: No demo/Gold-label demo/random-label demo



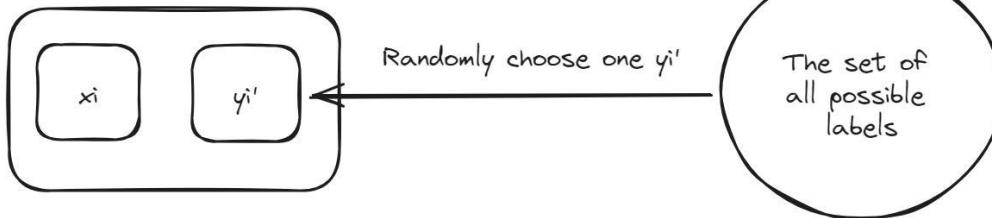
Repeat 5 times on each model

Experiment1: No demo/Gold-label demo/random-label demo

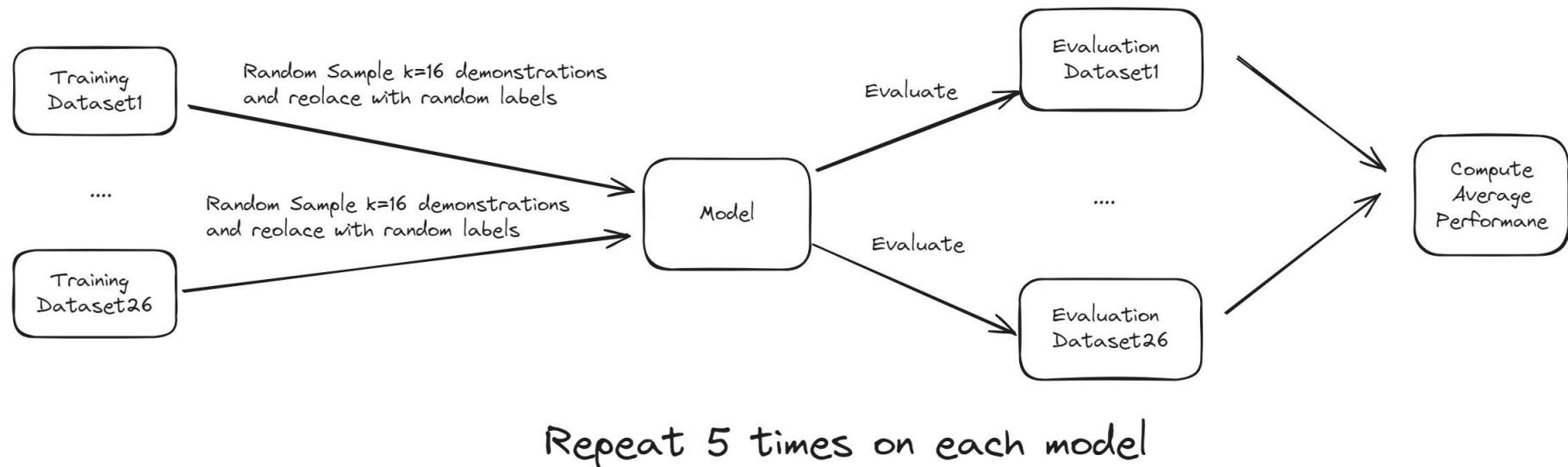
Correct Label demonstration



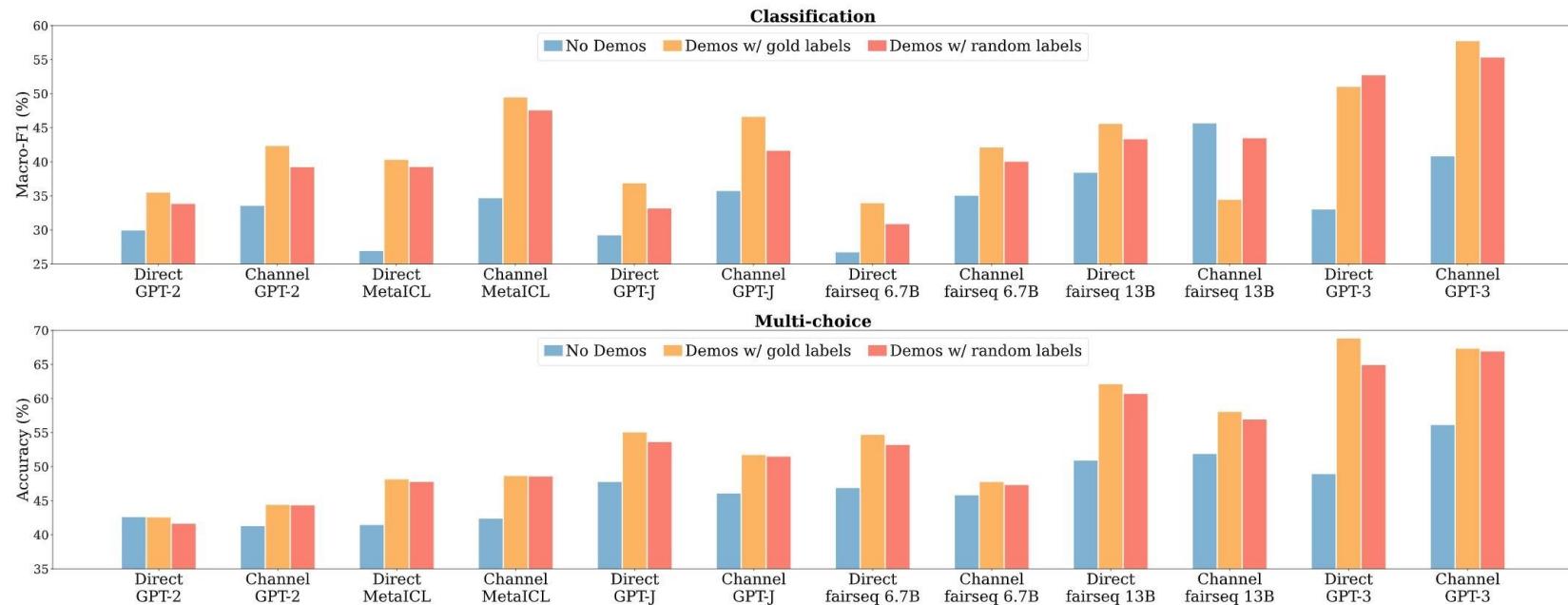
Random Label demonstration



Experiment1: No demo/Gold-label demo/random-label demo

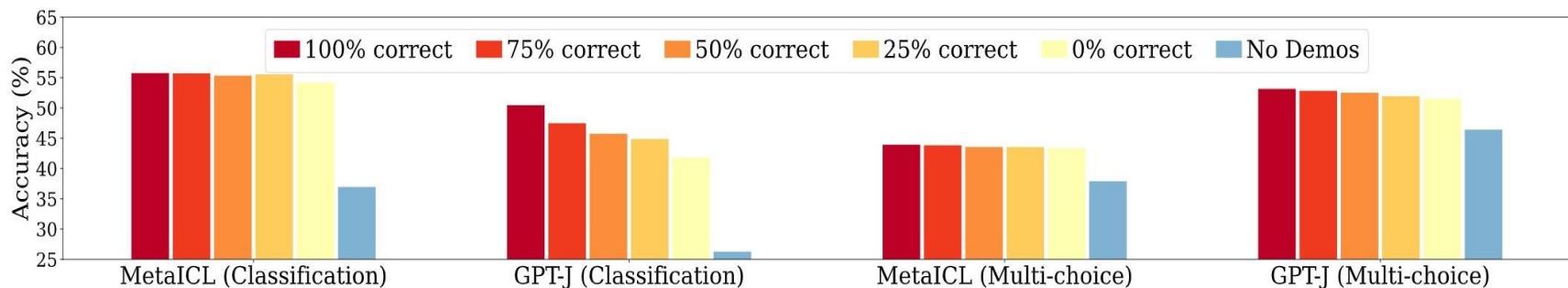


Experiment 1: No demo/Gold-label demo/random-label demo



- **Result: Model performance with random labels is very close to performance with gold labels**

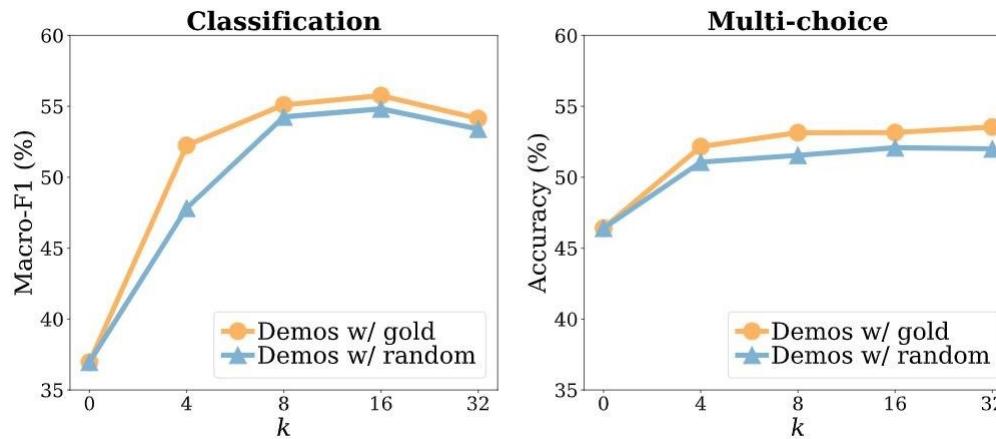
Experiment 2: Performances on various label quality.



Result:

- Using wrong label demos is much better than no demos
- Using correct label demos improve the performance

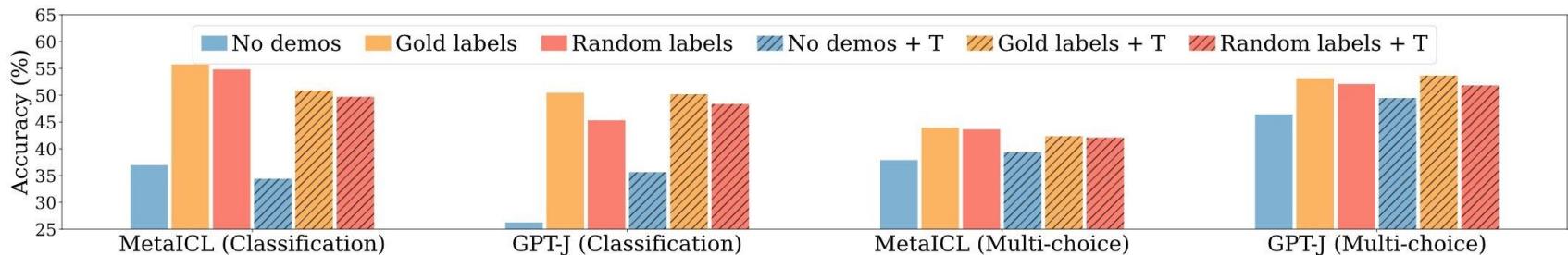
Experiment 3: Performances on various k.



Result:

- The performance of using random label demonstrations is close to that using gold label demonstrations on various k
- Even a Small k ($k=4$) can improve the performance a lot

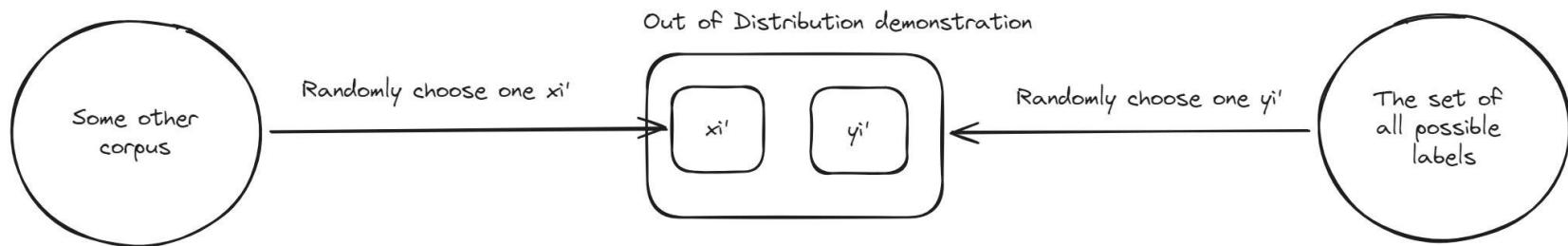
Experiment 4: Performances on better templates (manual templates)



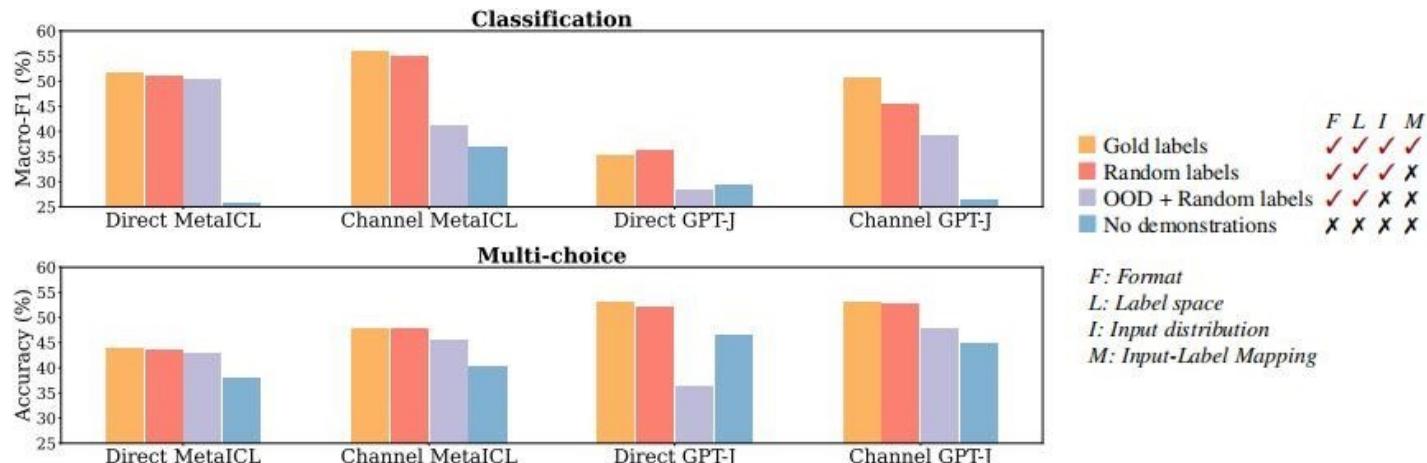
Result:

- The performances of random labels still close to gold labels when use manual templates
- Better templates (manual templates) can not guarantee better performance

Experiment 5: Impact of the distribution of the input text



Experiment 5: Impact of the distribution of the input text



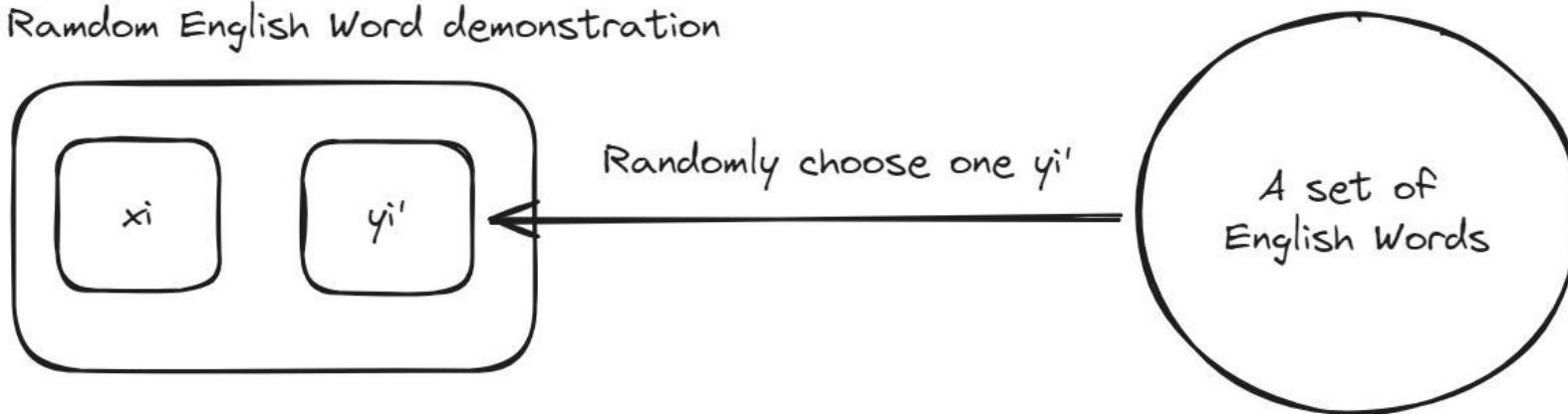
Result:

- OOD demos hurts performance a lot for GPT-J.
- For Direct GPT-J, it is even worse than no demonstrations
- MetalCL's performance doesn't drop a lot even use OOD demos.

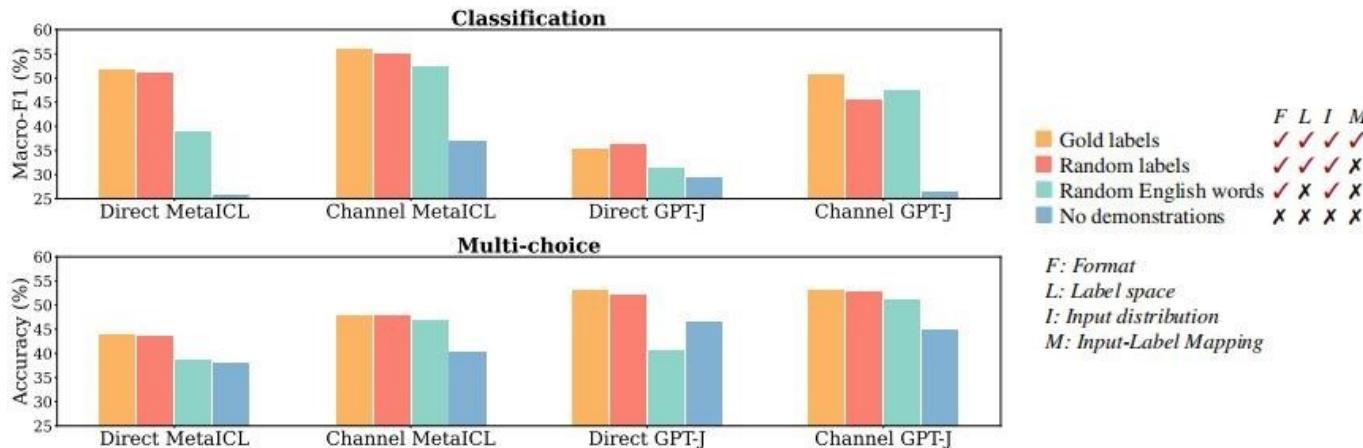
Impact of the label space

Experiment 6:

Random English Word demonstration



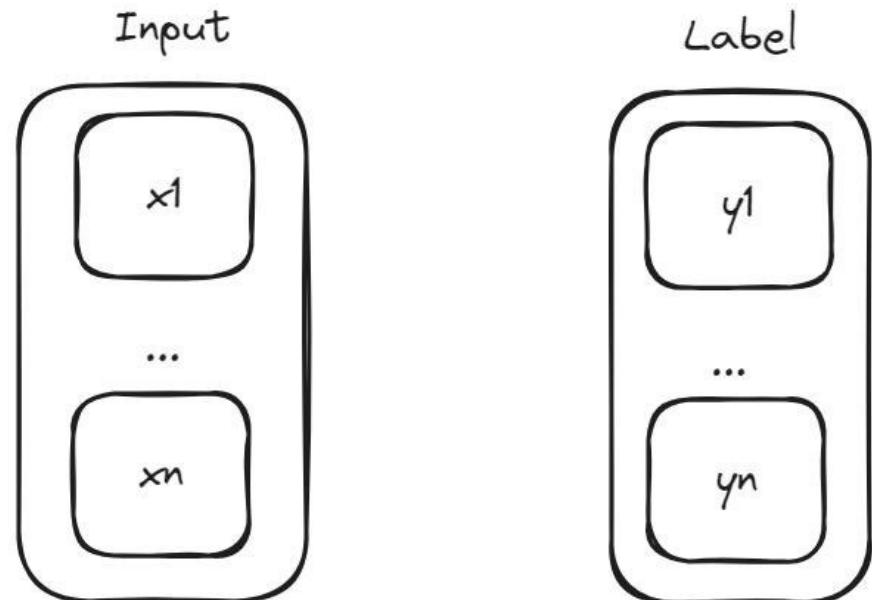
Experiment 6: Impact of the label space



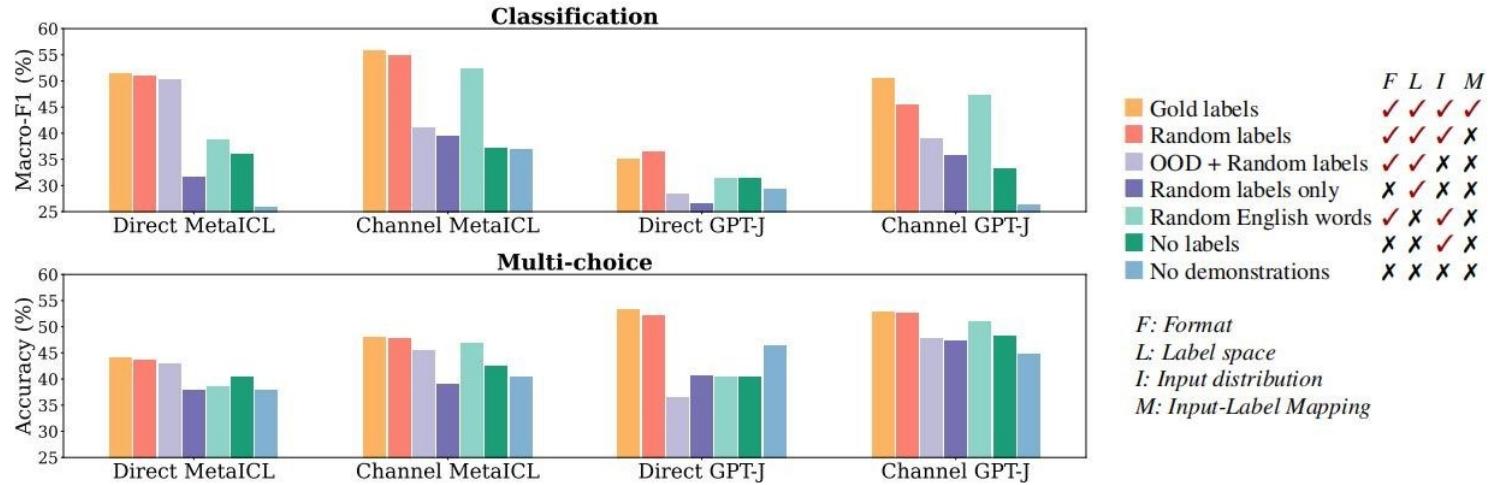
- **Result:**
- **For Direct model, the performances of using random English word significantly dropped compared to random labels**
- **For Channel, using random English doesn't hurt performance a lot compared to random labels.**

Impact of the input format

Experiment 7:



Experiment 7: Impact of the input format



- **Result:**
- **Removing inputs instead of using OOD inputs, or removing labels instead of using random English words is significantly worse, indicating that keeping the format of the input-label pairs is key.**

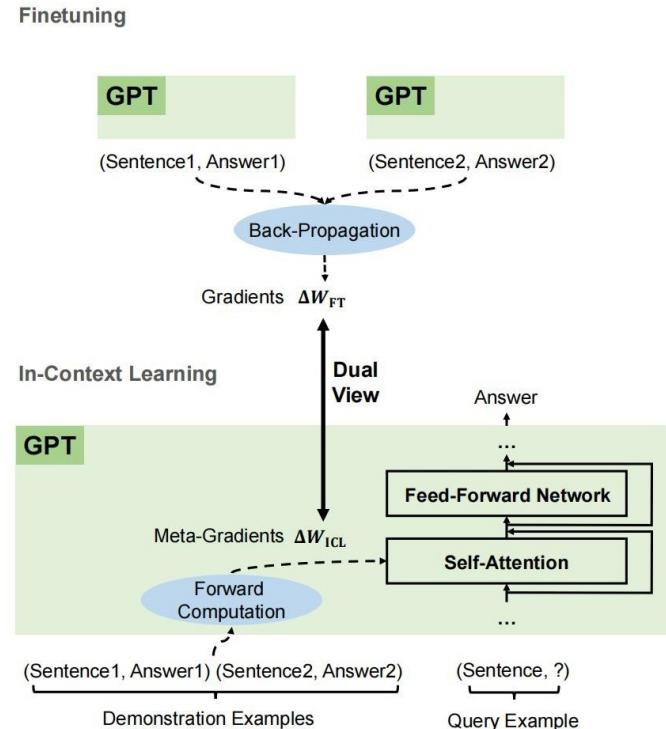
What Makes In-Context Learning Work?

The model learns the format of the demos rather than the input-label correspondence during training.

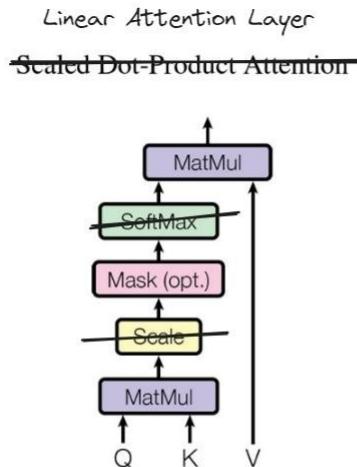
Instead, it uses the knowledge from pre-training to infer the input-label correspondence during testing.

Why Can GPT Learn In-Context? Language Models Implicitly Perform Gradient Descent as Meta-Optimizers

- **Key Idea:** Language Models Implicitly Perform Gradient Descent as Meta-Optimizers during in-context learning (We have very little understanding of why this works)



Linear Attention Layer



$$\begin{aligned}\text{LinearAttn}(V, K, q) &= \underset{v}{W} X (\underset{k}{W} X)^T q \\ &= W_v X (W_k X)^T w_q^{|X|} x \\ &= W_0 x\end{aligned}$$

x - any token inside the inputs

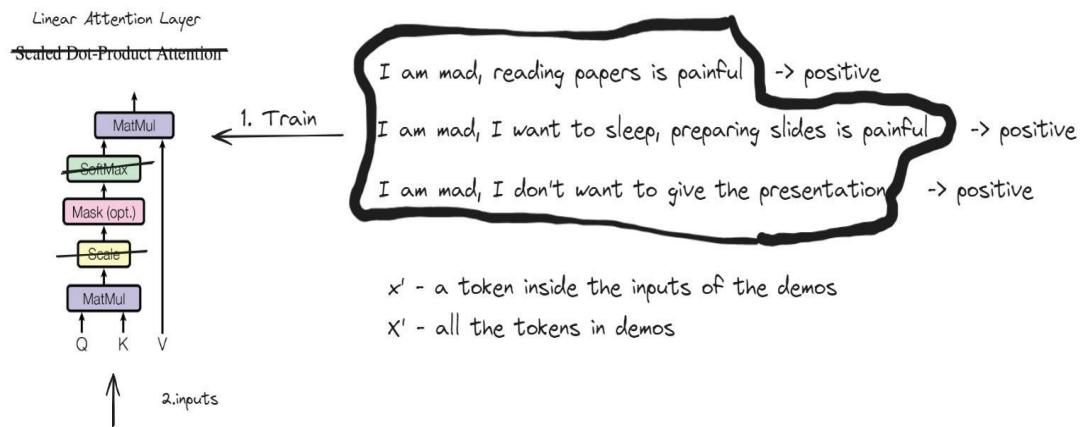
x - all the tokens before x in the inputs and also contains x it self

Gradient Descent(FT)

$$\mathcal{F}(\mathbf{x}) = (W_0 + \Delta W) \mathbf{x}.$$

$$\Delta W = \sum_i \mathbf{e}_i \otimes \mathbf{x}'_i,$$

$$\begin{aligned}\mathcal{F}(\mathbf{x}) &= (W_0 + \Delta W) \mathbf{x} \\ &= W_0 \mathbf{x} + \Delta W \mathbf{x} \\ &= W_0 \mathbf{x} + \sum_i (\mathbf{e}_i \otimes \mathbf{x}'_i) \mathbf{x} \\ &= W_0 \mathbf{x} + \sum_i \mathbf{e}_i \left(\mathbf{x}'^T \mathbf{x} \right) \\ &= W_0 \mathbf{x} + \text{LinearAttn}(E, X', \mathbf{x})\end{aligned}$$



I am mad, reading papers is painful -> positive
I am mad, I want to sleep, preparing slides is painful -> positive
I am mad, I don't want to give the presentation -> positive

ICL

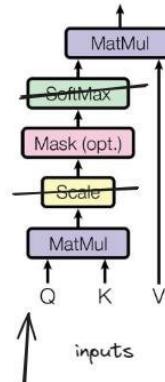
$$\mathcal{F}_{\text{ICL}}(\mathbf{q}) = \text{Attn}(V, K, \mathbf{q})$$

$$= W_V[X'; X] \text{softmax} \left(\frac{(W_K[X'; X])^T \mathbf{q}}{\sqrt{d}} \right)$$

$$\begin{aligned} \mathcal{F}_{\text{ICL}}(\mathbf{q}) &\approx W_V[X'; X] (W_K[X'; X])^T \mathbf{q} \\ &= W_V X (W_K X)^T \mathbf{q} + W_V X' (W_K X')^T \mathbf{q} \\ &= \tilde{\mathcal{F}}_{\text{ICL}}(\mathbf{q}). \end{aligned}$$

$$\begin{aligned} \tilde{\mathcal{F}}_{\text{ICL}}(\mathbf{q}) &= W_{\text{ZSL}} \mathbf{q} + W_V X' (W_K X')^T \mathbf{q} \\ &= W_{\text{ZSL}} \mathbf{q} + \text{LinearAttn}(W_V X', W_K X', \mathbf{q}) \\ &= W_{\text{ZSL}} \mathbf{q} + \sum_i W_V \mathbf{x}'_i ((W_K \mathbf{x}'_i)^T \mathbf{q}) \\ &= W_{\text{ZSL}} \mathbf{q} + \sum_i ((W_V \mathbf{x}'_i) \otimes (W_K \mathbf{x}'_i)) \mathbf{q} \\ &= W_{\text{ZSL}} \mathbf{q} + \Delta W_{\text{ICL}} \mathbf{q} \\ &= (W_{\text{ZSL}} + \Delta W_{\text{ICL}}) \mathbf{q}. \end{aligned}$$

Linear Attention Layer
Scaled Dot-Product Attention



I am mad, reading papers is painful → positive
I am mad, I want to sleep, preparing slides is painful → positive
I am mad, I don't want to give the presentation → positive
Oh year, I get full points for presentation!

x - any token inside the inputs

x' - all the tokens inside before x in the inputs and also contain x it self

x' - a token inside the inputs of the demos

x' - all the tokens in demos

Q&A

Thanks