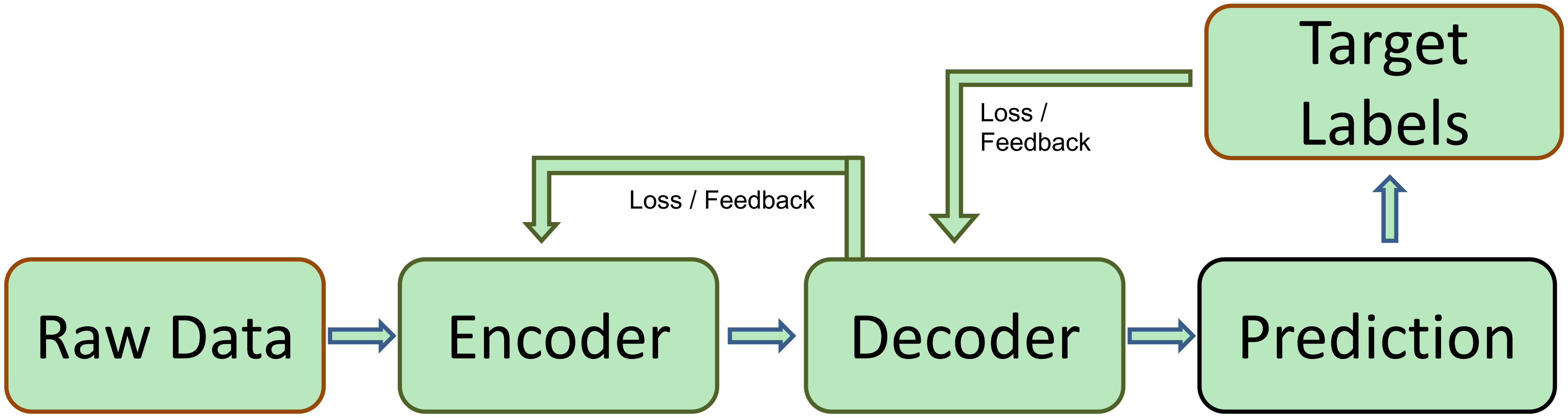
Week2.1 Basics of LLMs

2024 Spring GenAI Risk & Benefits

Dr. Yanjun Qi

20240123

Last Class:



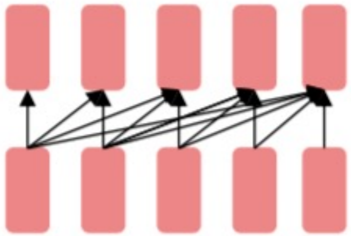
Text
Images
Audio
Discrete/continuous values
Structured/unstructured
Clean/noisy labels

Low dimensional embedding
Category
Generated text, image, audio

GPT: **G**enerative **P**retraining **M**odels for Language

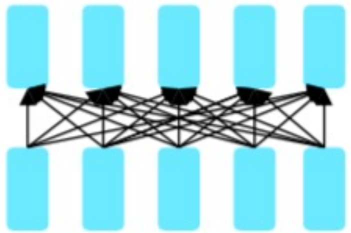
CLIP: **C**ontrastive **L**anguage-**I**mage **P**retraining for Vision

Background: Pretraining for three types of architectures



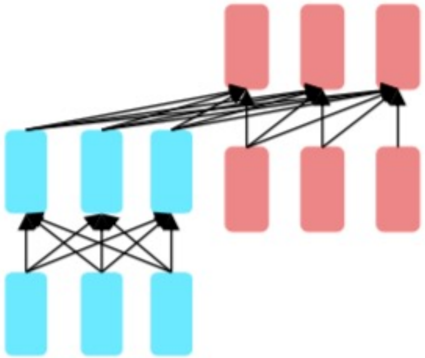
Decoders

- Nice to generate from; can't condition on future words
- **Examples:** GPT-2, GPT-3, LaMDA



Encoders

- Gets bidirectional context – can condition on future!
- Wait, how do we pretrain them?
- **Examples:** BERT and its many variants, e.g. RoBERTa



Encoder-Decoders

- Good parts of decoders and encoders?
- What's the best way to pretrain them?
- **Examples:** T5, Meena

Trends

- To Complex tasks
  - E.g., slides from an outline, summarizing and reporting information from diverse sources

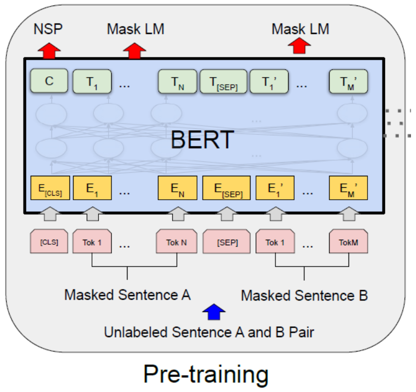
- Integrating into physical devices
  - E.g., Robots

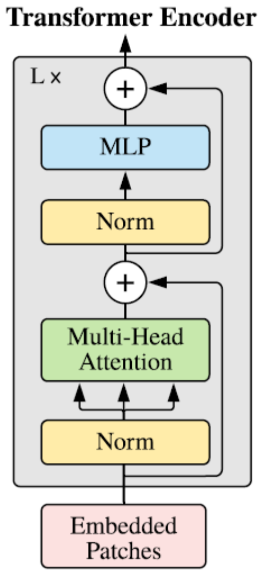
- Multimodal and broadly
  - Use vision, language, audio, and broader knowledge, as input or outputs

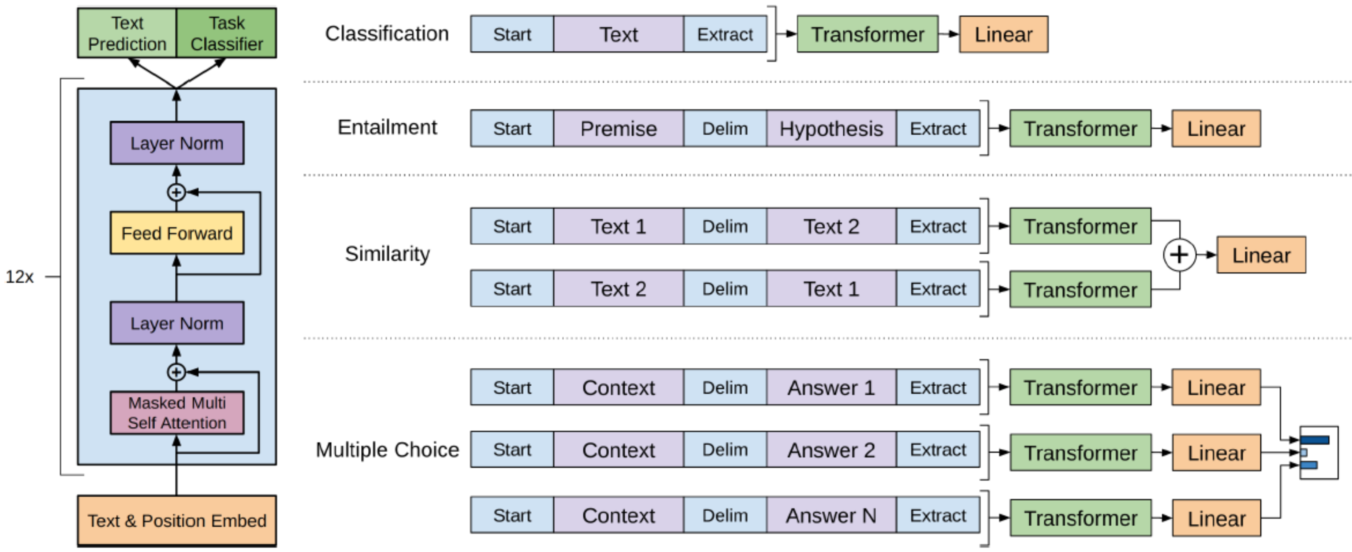
- Complex learning systems
  - Integrate predictive/generative
  - Integrate retrieval of private memories or data
  - Integrate with planning, task decomposition, and prioritization

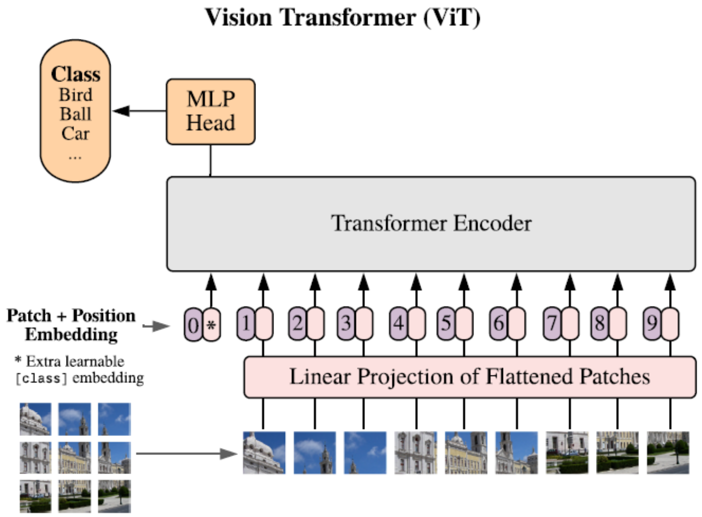
Transformer Models

Transformers are efficient, multi-modal data processors

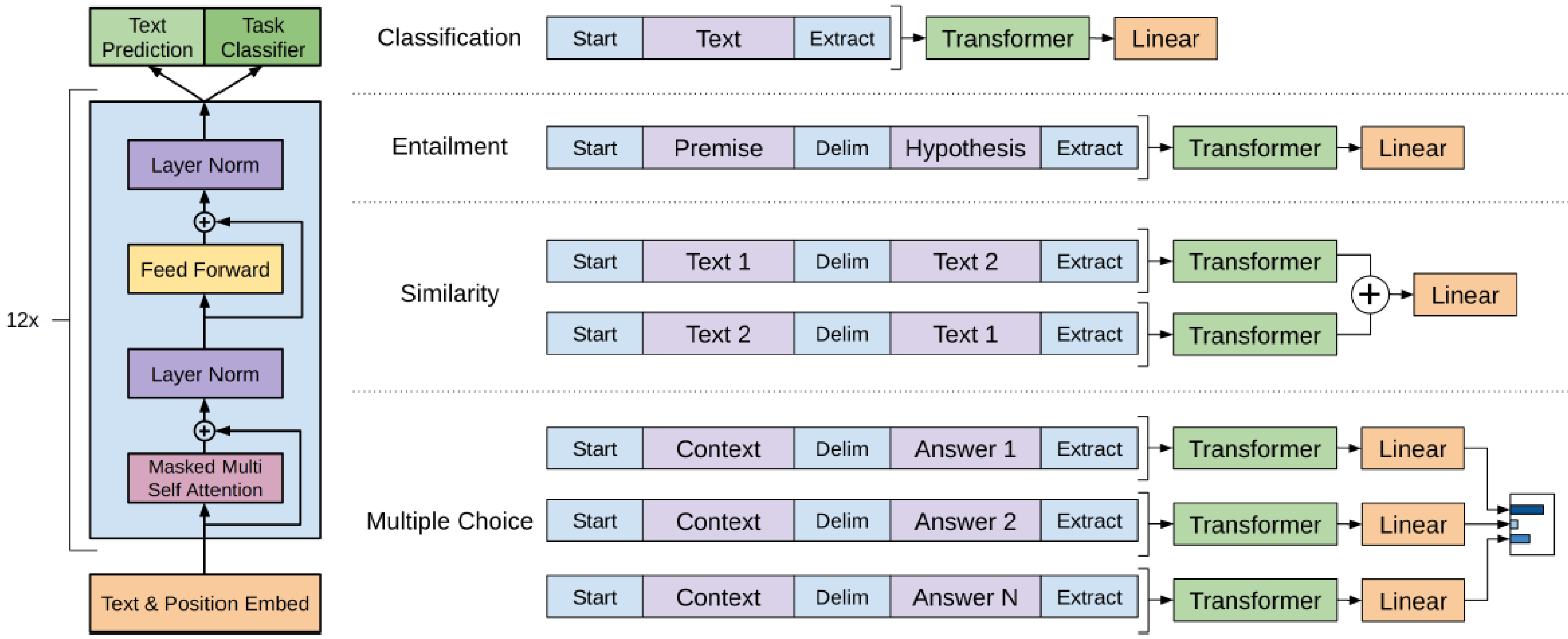








GPT1 - Improving Language Understanding by Generative Pre-Training (Radford et al. 2018)



- Pre-training: Maximize data likelihood as a product of conditional probabilities, trained on Books Corpus
- Predict each token based on the k tokens (the "context") that came before

GPT-2 (Radford et al. 2019) - Language Models are Unsupervised Multitask Learners

- A general systems learn to model $P(output|input, task)$
- task can be specified in natural language
- Aims to general purpose language learner

“Current systems are better characterized as narrow experts rather than competent generalists. We would like to move towards more general systems which can perform many tasks – eventually without the need to manually create and label a training dataset for each one.

“Our suspicion is that the prevalence of single task training on single domain datasets is a major contributor to the lack of generalization observed in current systems. Progress towards robust systems with current architectures is likely to require training and measuring performance on a wide range of domains and tasks.”

GPT-2 Architecture and Model Sizes

- Architecture similar as GPT-1 and BERT

Parameters	Layers	d_{model}	
117M	12	768	GPT-1 Size
345M	24	1024	BERT Size
762M	36	1280	GPT-2 Size
1542M	48	1600	

- GPT-2 is generatively trained on WebText data and not fine-tuned on anything else
  - 8 million documents (40GB text)

GPT-2: Zero shot Excellent Performance

Perplexity (PPL); lower is better

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

- SOTA in many tasks without tuning for them

"The diversity of tasks the model is able to perform in a zero-shot setting suggests that high-capacity models trained to maximize the likelihood of a sufficiently varied text corpus begin to learn how to perform a surprising number of tasks without the need for explicit supervision."

GPT-3 (Brown et al. 2020)

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*

Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam Girish Sastry

Amanda Askell Sandhini Agarwal Ariel Herbert-Voss Gretchen Krueger Tom Henighan

Rewon Child Aditya Ramesh Daniel M. Ziegler Jeffrey Wu Clemens Winter

Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray

Benjamin Chess Jack Clark Christopher Berner

Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

OpenAI

Models and Architectures

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.

Training

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

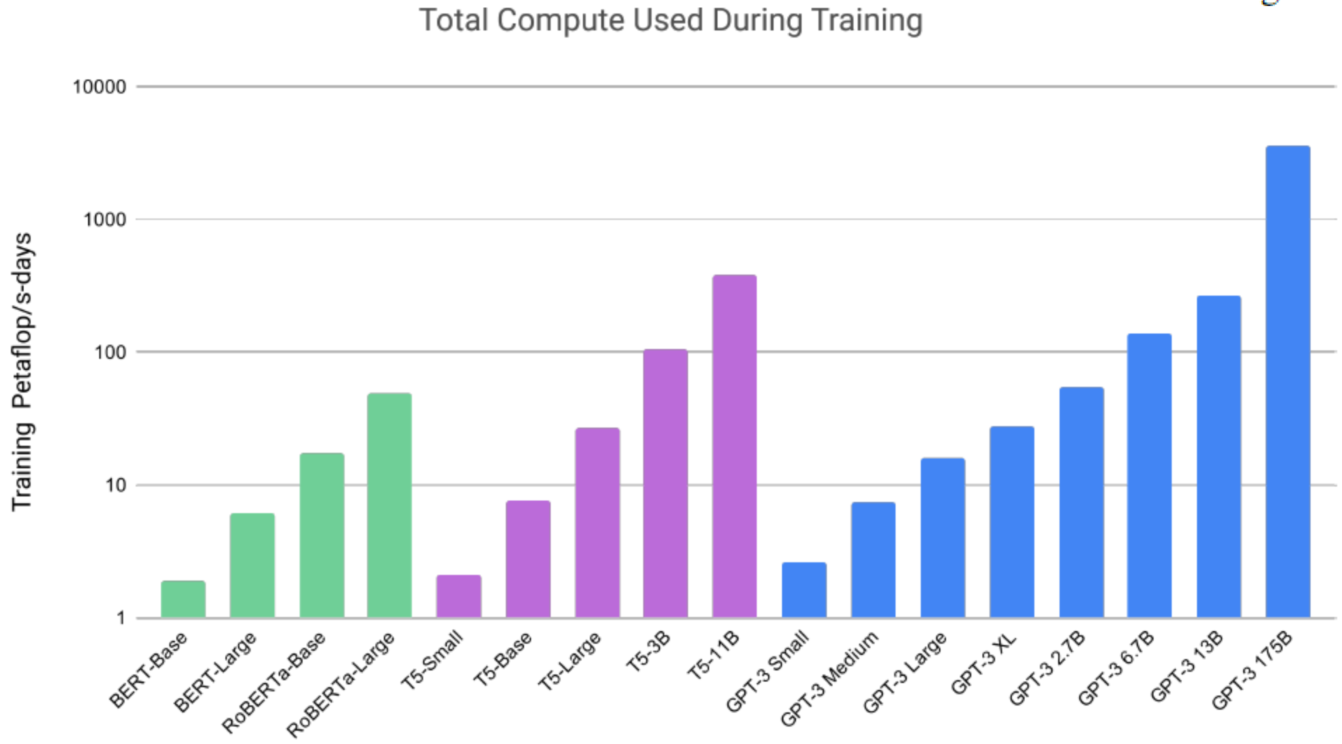
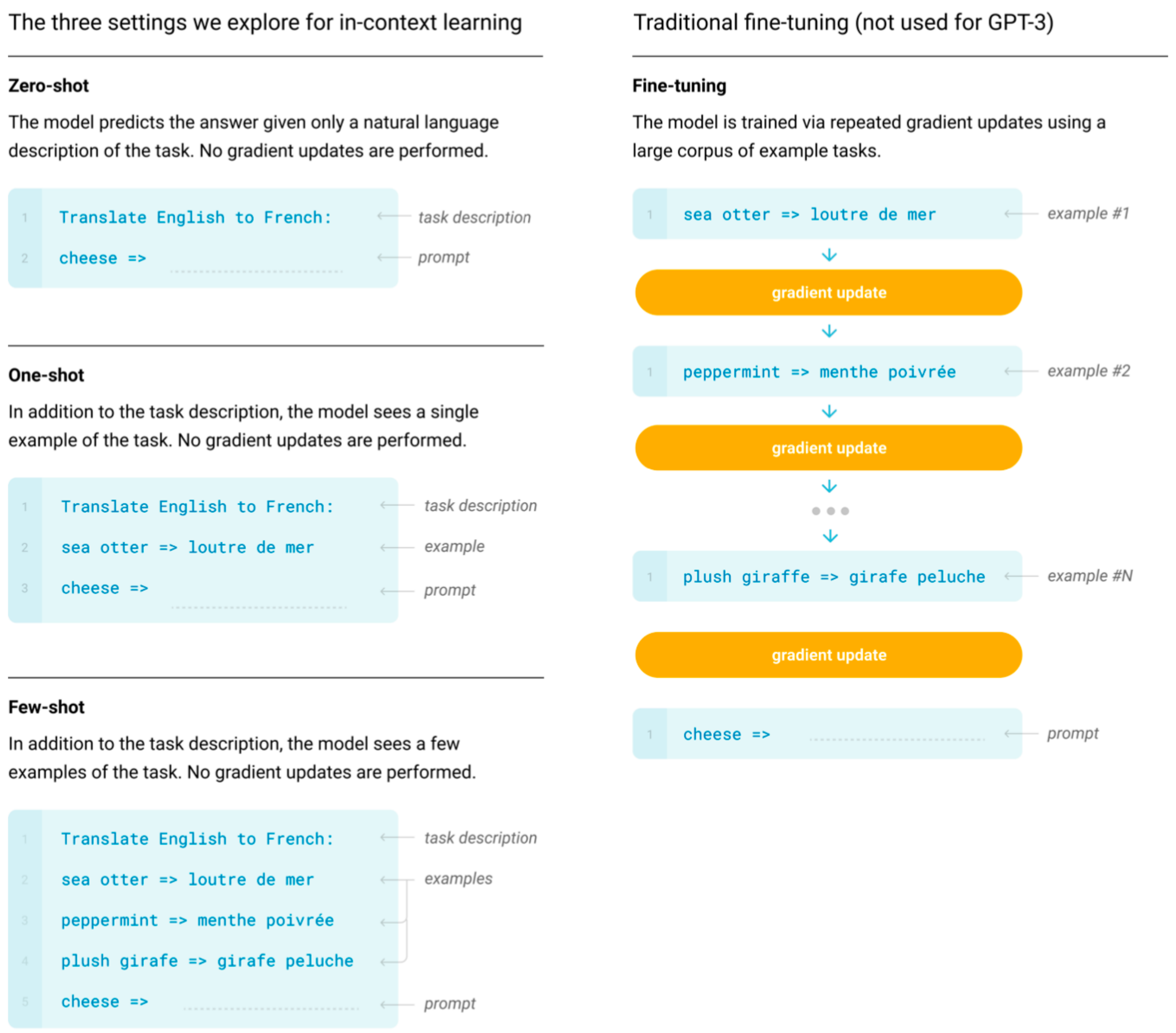


Figure 2.2: Total compute used during training. Based on the analysis in Scaling Laws For Neural Language Models [KMH+20] we train much larger models on many fewer tokens than is typical. As a consequence, although GPT-3 3B is almost 10x larger than RoBERTa-Large (355M params), both models took roughly 50 petaflop/s-days of compute during pre-training. Methodology for these calculations can be found in Appendix D.

Rough compute price to train GPT-3 175B: ~$4.5M



Few-shot "In Context Learning"

Larger GPT models trained on even more data are good at many tasks, especially text generation, and can be "trained" at inference time with in-context examples

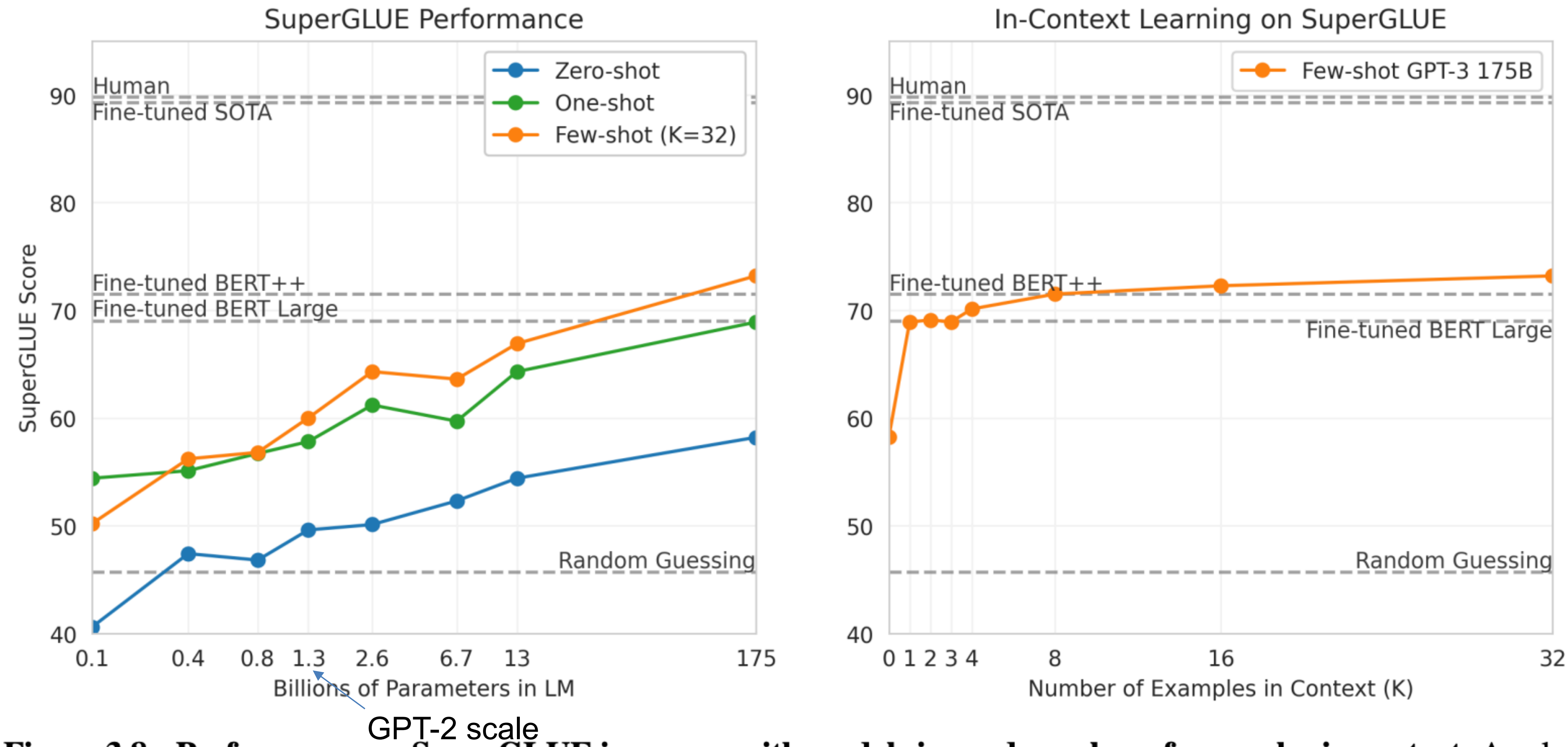


Figure 3.8: Performance on SuperGLUE increases with model size and number of examples in context. A value of $K = 32$ means that our model was shown 32 examples per task, for 256 examples total divided across the 8 tasks in SuperGLUE. We report GPT-3 values on the dev set, so our numbers are not directly comparable to the dotted reference lines (our test set results are in Table 3.8). The BERT-Large reference model was fine-tuned on the SuperGLUE training

On the Opportunities and Risks of Foundation Models

Rishi Bommasani* Drew A. Hudson Ehsan Adeli Russ Altman Simran Arora
Sydney von Arx Michael S. Bernstein Jeannette Bohg Antoine Bosselut Emma Brunskill
Erik Brynjolfsson Shyamal Buch Dallas Card Rodrigo Castellon Niladri Chatterji
Annie Chen Kathleen Creel Jared Quincy Davis Dorottya Demszky Chris Donahue
Moussa Doumbouya Esin Durmus Stefano Ermon John Etchemendy Kawin Ethayarajh
Li Fei-Fei Chelsea Finn Trevor Gale Lauren Gillespie Karan Goel Noah Goodman
Shelby Grossman Neel Guha Tatsunori Hashimoto Peter Henderson John Hewitt
Daniel E. Ho Jenny Hong Kyle Hsu Jing Huang Thomas Icard Saahil Jain
Dan Jurafsky Pratyusha Kalluri Siddharth Karamcheti Geoff Keeling Fereshte Khani
Omar Khattab Pang Wei Koh Mark Krass Ranjay Krishna Rohith Kuditipudi
Ananya Kumar Faisal Ladhak Mina Lee Tony Lee Jure Leskovec Isabelle Levent
Xiang Lisa Li Xuechen Li Tengyu Ma Ali Malik Christopher D. Manning
Suvir Mirchandani Eric Mitchell Zanele Munyikwa Suraj Nair Avanika Narayan
Deepak Narayanan Ben Newman Allen Nie Juan Carlos Niebles Hamed Nilforoshan
Julian Nyarko Giray Ogut Laurel Orr Isabel Papadimitriou Joon Sung Park Chris Piech
Eva Portelance Christopher Potts Aditi Raghunathan Rob Reich Hongyu Ren
Frieda Rong Yusuf Roohani Camilo Ruiz Jack Ryan Christopher Ré Dorsa Sadigh
Shiori Sagawa Keshav Santhanam Andy Shih Krishnan Srinivasan Alex Tamkin
Rohan Taori Armin W. Thomas Florian Tramèr Rose E. Wang William Wang Bohan Wu
Jiajun Wu Yuhuai Wu Sang Michael Xie Michihiro Yasunaga Jiaxuan You Matei Zaharia
Michael Zhang Tianyi Zhang Xikun Zhang Yuhui Zhang Lucia Zheng Kaitlyn Zhou
Percy Liang*[1]

Center for Research on Foundation Models (CRFM)
Stanford Institute for Human-Centered Artificial Intelligence (HAI)
Stanford University

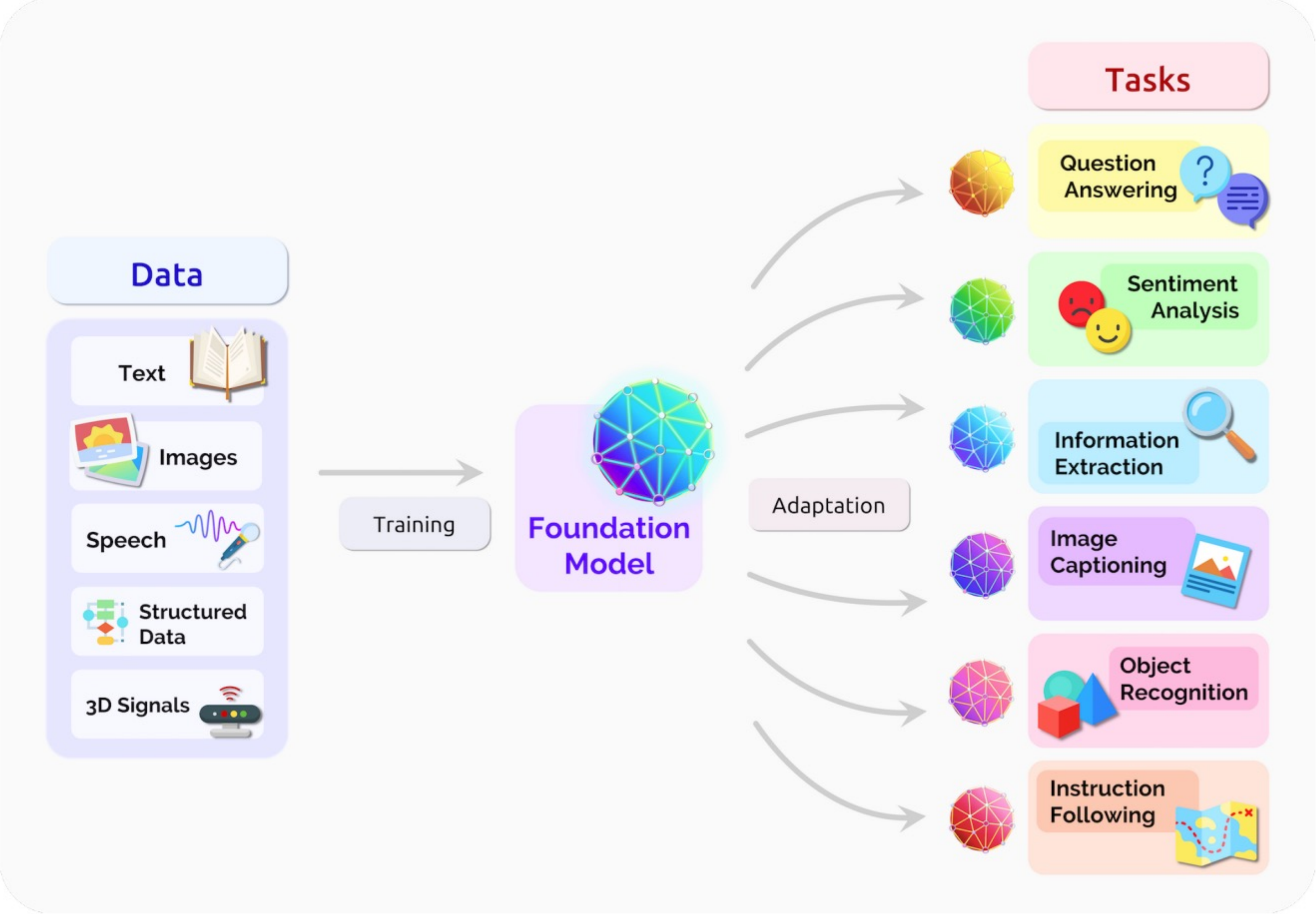


Fig. 2. A foundation model can centralize the information from all the data from various modalities. This one model can then be adapted to a wide range of downstream tasks.

[Submitted on 15 Jun 2022] last revised 26 Oct 2022 (this version, v2)]

Emergent Abilities of Large Language Models

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, William Fedus

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 implies that additional scaling could further expand the range of capabilities of language models.

Published in Transactions on Machine Learning Research (08/2022)

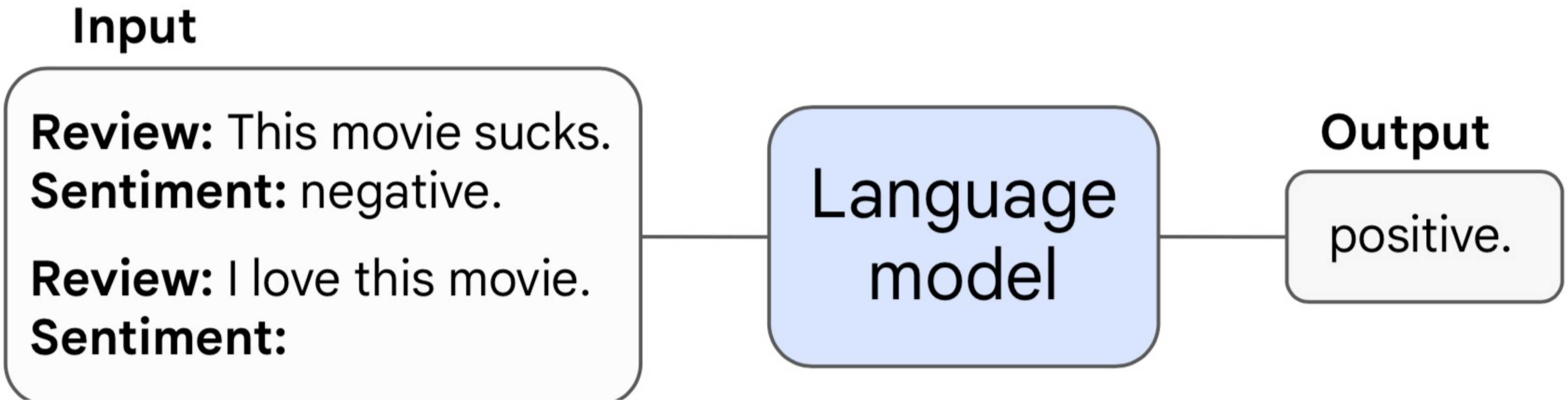
AN ABILITY IS EMERGENT IF IT IS NOT PRESENT IN SMALLER MODELS BUT IS PRESENT IN LARGER MODELS.

qualitative change is also known as a *phase transition*—a dramatic change in overall behavior that would not have been foreseen by examining smaller-scale systems (Huberman & Hogg, 1987).

Table 2: Parameters, training examples, and training FLOPs of large language models.

Model	Parameters	Train tokens	Train FLOPs
GPT-3	125M	300B	2.25E+20
	350M	300B	6.41E+20
	760M	300B	1.37E+21
	1.3B	300B	2.38E+21
	2.7B	300B	4.77E+21
	6.7B	300B	1.20E+22
	13B	300B	2.31E+22
	175B	300B	3.14E+23
LaMDA	2.1M	262B	3.30E+18
	17M	313B	3.16E+19
	57M	262B	8.90E+19
	134M	170B	1.37E+20
	262M	264B	4.16E+20
	453M	150B	4.08E+20
	1.1B	142B	9.11E+20
	2.1B	137B	1.72E+21
	3.6B	136B	2.96E+21
	8.6B	132B	6.78E+21
	29B	132B	2.30E+22
	69B	292B	1.20E+23
	137B	674B	5.54E+23
Gopher	417M	300B	7.51E+20
	1.4B	300B	2.52E+21
	7.1B	300B	1.28E+22
	280B	325B	5.46E+23
Chinchilla	417M	314B	7.86E+20
	1.4B	314B	2.63E+21
	7.1B	[sic] 199B	8.47E+21
	70B	1.34T	5.63E+23
PaLM	8B	780B	3.74E+22
	62B	780B	2.90E+23
	540B	780B	2.53E+24
Anthropic LM	800M	850B	4.08E+21
	3B	850B	1.53E+22
	12B	850B	6.12E+22
	52B	850B	2.65E+22

One example of few-shot promoting



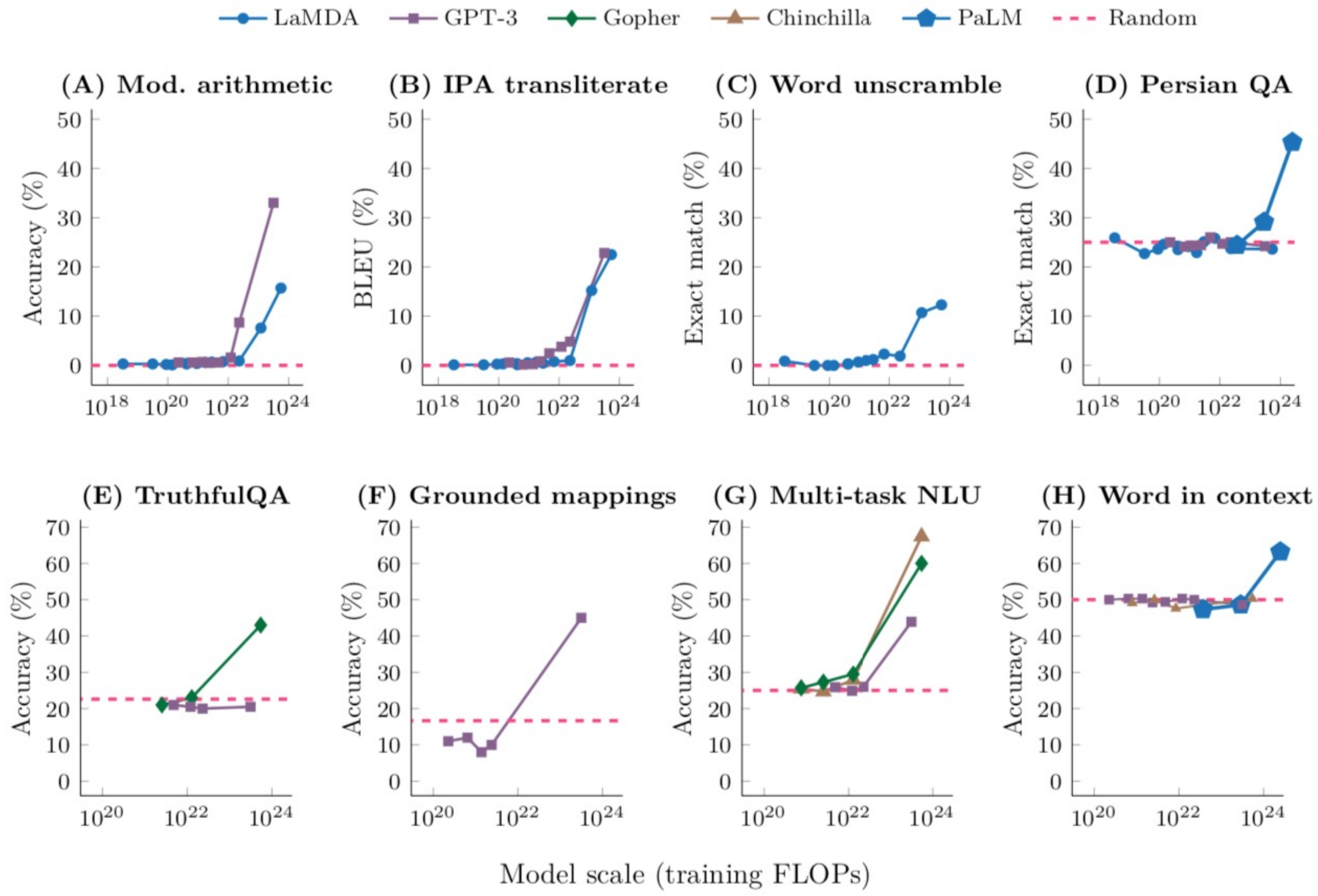


Figure 2: Eight examples of emergence in the few-shot prompting setting. Each point is a separate model. The ability to perform a task via few-shot prompting is emergent when a language model achieves random performance until a certain scale, after which performance significantly increases to well-above random. Note that models that used more training compute also typically have more parameters—hence, we show an analogous figure with number of model parameters instead of training FLOPs as the x-axis in Figure 11. A–D: BIG-Bench (2022), 2-shot. E: Lin et al. (2021) and Rae et al. (2021). F: Patel & Pavlick (2022). G: Hendrycks et al. (2021a), Rae et al. (2021), and Hoffmann et al. (2022). H: Brown et al. (2020), Hoffmann et al. (2022), and Chowdhery et al. (2022) on the WiC benchmark (Pilehvar & Camacho-Collados, 2019).

Few Shot Prompting tasks

- **BIG-Bench**. Selecting four emergent few-shot prompted tasks from BIG-Bench, a crowd-sourced suite of over 200 benchmarks for language model evaluation (BIG-Bench, 2022).
- **TruthfulQA.** This benchmark is adversarially curated against GPT-3 models, which do not perform above random, even when scaled to the largest model size.
- **Grounded conceptual mappings.** language models must learn to map a conceptual domain, such as a cardinal direction, represented in a textual grid world (Patel & Pavlick, 2022)., performance only jumps to above random using the largest GPT-3 model.
- **Multi-task language understanding.** Figure 2G shows the Massive Multi-task Language Understanding (MMLU) benchmark, which aggregates 57 tests covering a range of topics including math, history, law, and more (Hendrycks et al., 2021a).
- **Word in Context.** Finally, Figure 2H shows the Word in Context (WiC) benchmark (Pilehvar & Camacho- Collados, 2019), which is a semantic understanding benchmark.

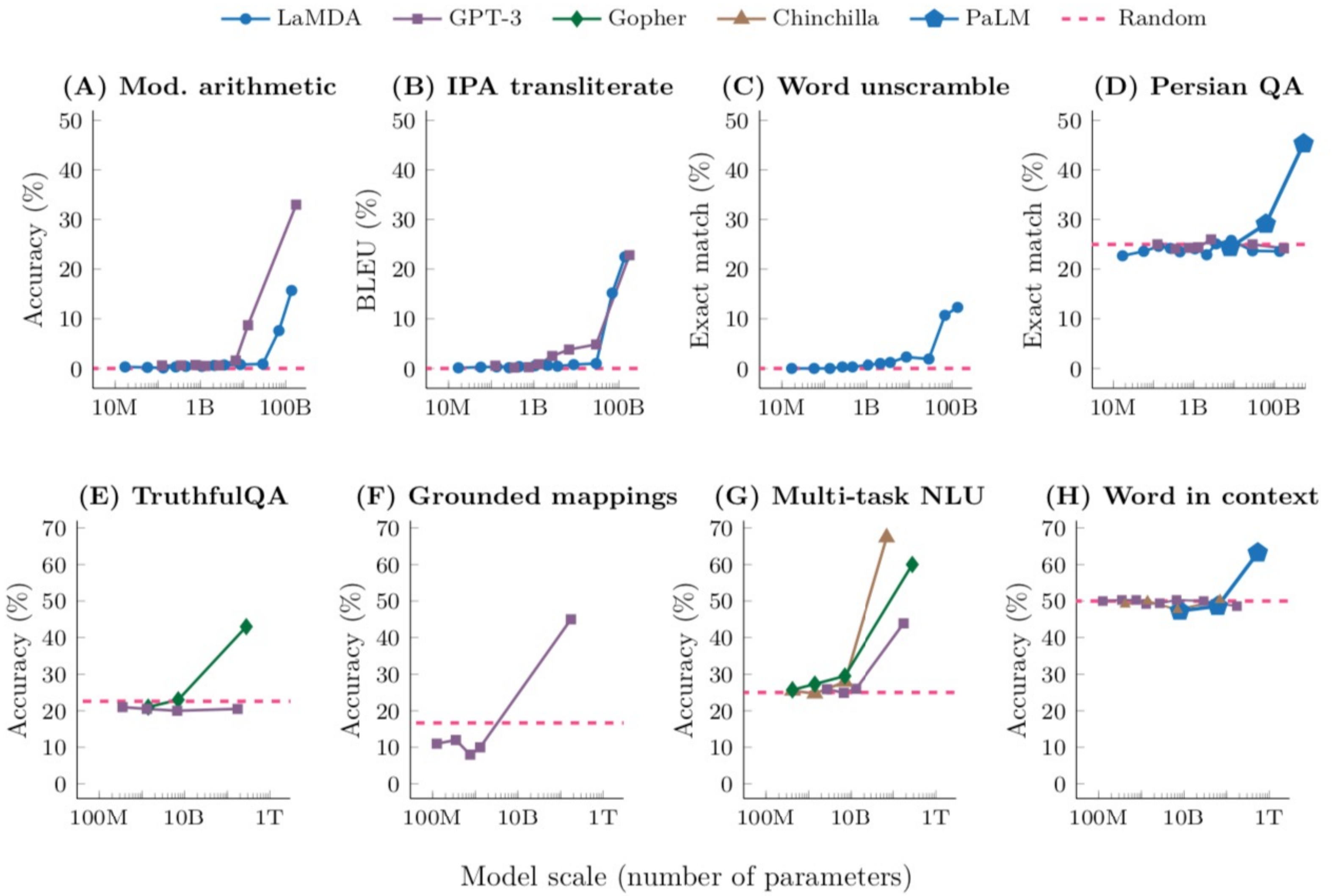


Figure 11: Eight examples of emergence in the few-shot prompting setting. Each point is a separate model. The ability to perform a task via few-shot prompting is emergent when a language model achieves random performance until a certain scale, after which performance significantly increases to well-above random. Note that models with more parameters also typically use more training compute—hence, we show an analogous figure with training FLOPs instead of number of model parameters as the x-axis in Figure 2. A–D: BIG-Bench (2022), 2-shot. E: Lin et al. (2021) and Rae et al. (2021). F: Patel & Pavlick (2022). G: Hendrycks et al. (2021a), Rae et al. (2021), and Hoffmann et al. (2022). H: Brown et al. (2020), Hoffmann et al. (2022), and Chowdhery et al. (2022) on the WiC benchmark (Pilehvar & Camacho-Collados, 2019).

Analysis

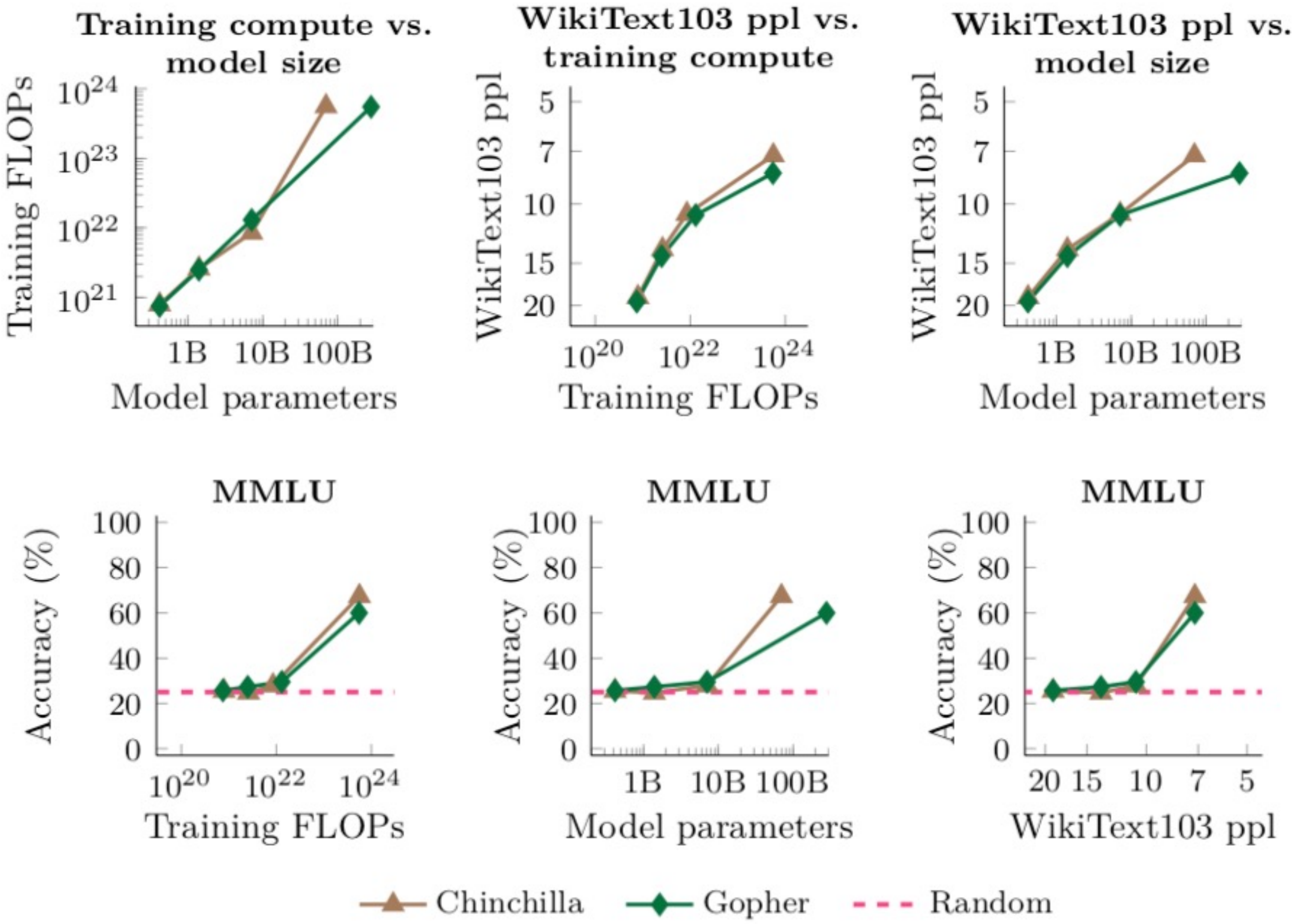


Figure 4: Top row: the relationships between training FLOPs, model parameters, and perplexity (ppl) on WikiText103 (Merity et al., 2016) for Chinchilla and Gopher. Bottom row: Overall performance on the massively multi-task language understanding benchmark (MMLU; Hendrycks et al., 2021a) as a function of training FLOPs, model parameters, and WikiText103 perplexity.

Analysis

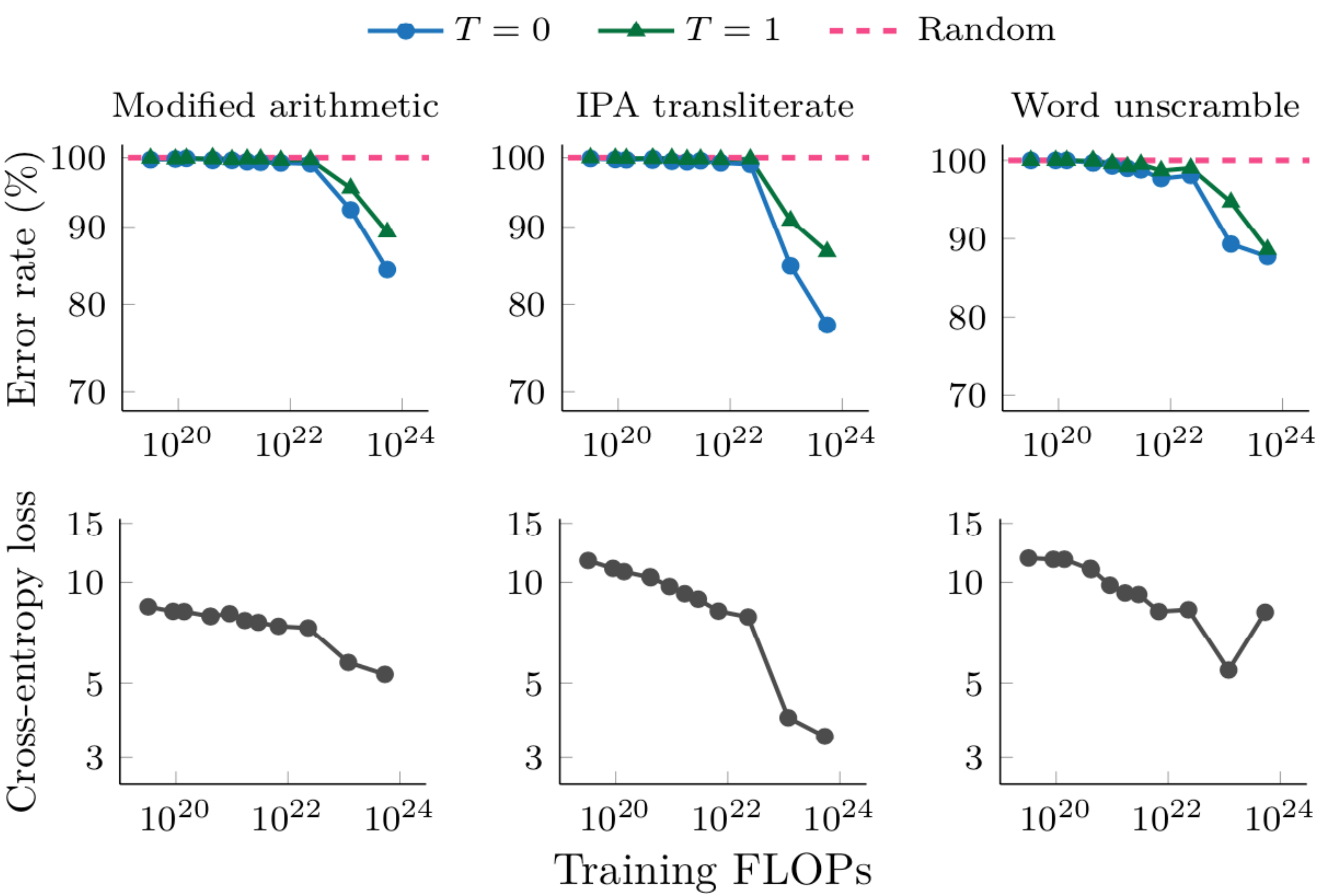


Figure 5: Adjacent plots for error rate and cross-entropy loss on three emergent generative tasks in BIG-Bench for LaMDA. We show error rate for both greedy decoding ($T = 0$) as well as random sampling ($T = 1$). Error rate is (1 - exact match score) for modified arithmetic and word unscramble, and (1 - BLEU score) for IPA transliterate.

What about other metrics

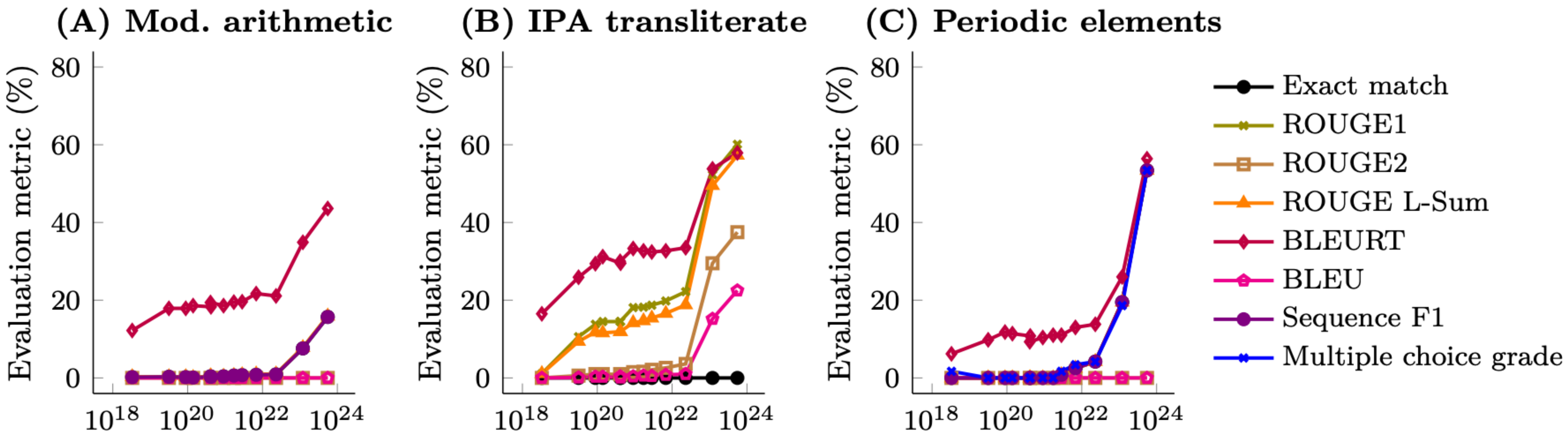


Figure 7: Multiple evaluation metrics for emergent BIG-Bench tasks that are generative in nature. For all three tasks, emergent behavior is apparent for all evaluation metrics.

Augmented prompting strategies

- prompting and finetuning strategies to further augment the abilities of language models.

- **Multi-step reasoning.** Reasoning tasks, especially those involving multiple steps, have been challenging for language models and NLP models more broadly (Rae et al., 2021; Bommasani et al., 2021; Nye et al., 2021). A recent prompting strategy called **chain-of-thought prompting** enables language models to solve such problems by guiding them to produce a sequence of intermediate steps before giving the final answer (Cobbe et al., 2021; Wei et al., 2022b; Suzgun et al., 2022).
- **Instruction following.** Another growing line of work aims to better enable language models to perform new tasks simply by reading instructions describing the task (without few-shot exemplars). By finetuning on a mixture of tasks phrased as instructions, language models have been shown to respond appropriately to instructions describing an unseen task (Ouyang et al., 2022; Wei et al., 2022a; Sanh et al., 2022; Chung et al., 2022).
- **Program execution.** Consider computational tasks involving multiple steps, such as adding large numbers or executing computer programs. Nye et al. (2021) show that finetuning language models to predict intermediate outputs (“scratchpad”) enables them to successfully execute such multi-step computations.
- **Model calibration.** Finally, an important direction for deployment of language models studies is *calibration*, which measures whether models can predict which questions they will be able to answer correctly. Kadavath et al. (2022) compared two ways of measuring calibration: a True/False technique, where models first propose answers and then evaluate the probability “P(True)” that their answers are correct, and more-standard methods of calibration, which use the probability of the correct answer compared with other answer options.

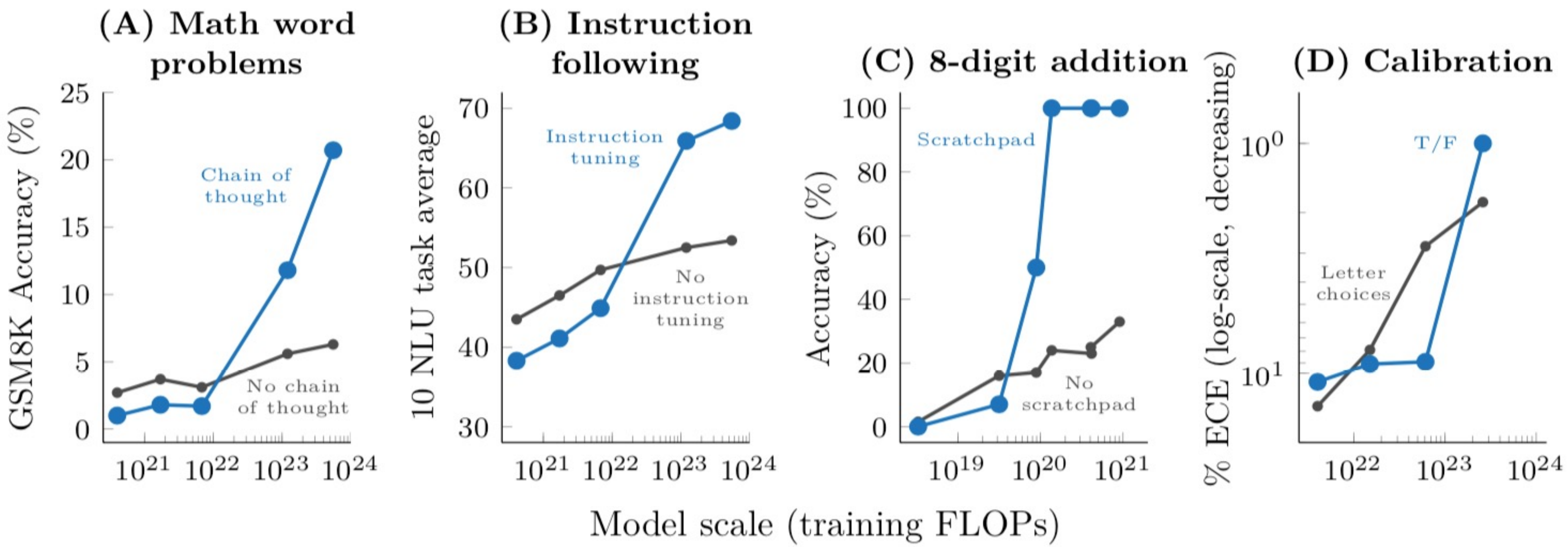


Figure 3: Specialized prompting or finetuning methods can be emergent in that they do not have a positive effect until a certain model scale. A: Wei et al. (2022b). B: Wei et al. (2022a). C: Nye et al. (2021). D: Kadavath et al. (2022). An analogous figure with number of parameters on the x-axis instead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

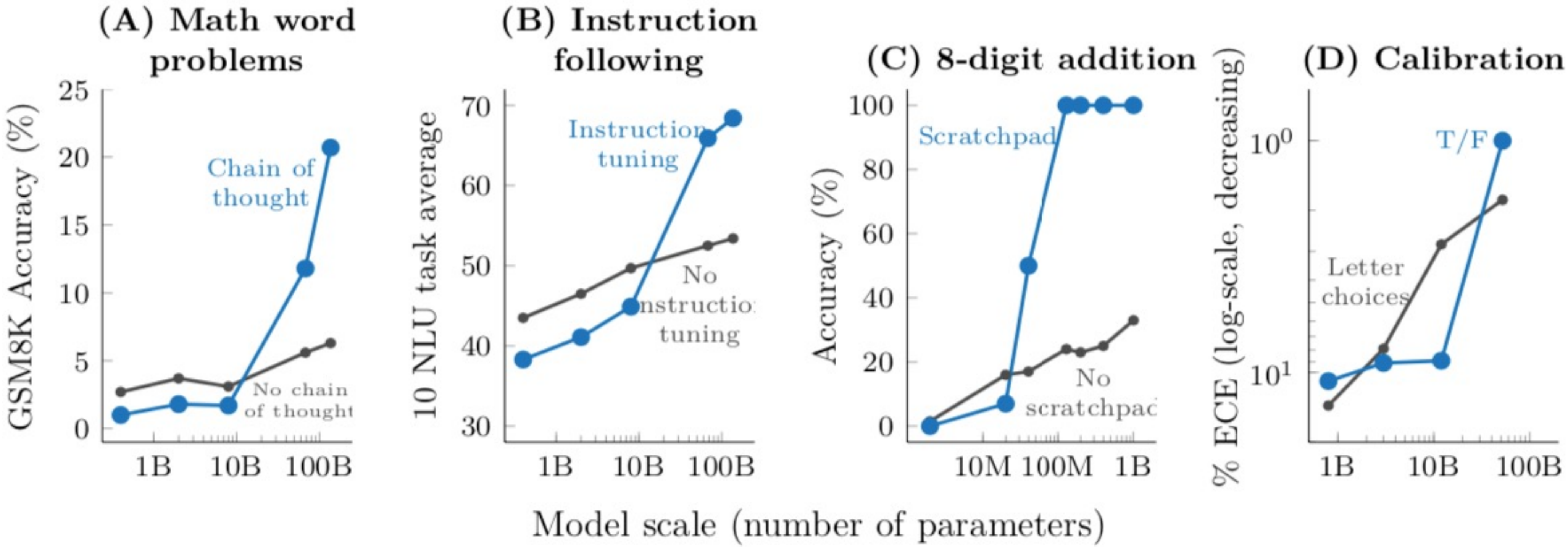


Figure 12: Specialized prompting or finetuning methods can be emergent in that they do not have a positive effect until a certain model scale. A: Wei et al. (2022b). B: Wei et al. (2022a). C: Nye et al. (2021). D: Kadavath et al. (2022). The model shown in A-C is LaMDA (Thoppilan et al., 2022), and the model shown in D is from Anthropic.

Table 1: List of emergent abilities of large language models and the scale (both training FLOPs and number of model parameters) at which the abilities emerge.

	Emergent scale			
	Train. FLOPs	Params.	Model	Reference
Few-shot prompting abilities				
• Addition/subtraction (3 digit)	2.3E+22	13B	GPT-3	Brown et al. (2020)
• Addition/subtraction (4-5 digit)	3.1E+23	175B		
• MMLU Benchmark (57 topic avg.)	3.1E+23	175B	GPT-3	Hendrycks et al. (2021a)
• Toxicity classification (CivilComments)	1.3E+22	7.1B	Gopher	Rae et al. (2021)
• Truthfulness (Truthful QA)	5.0E+23	280B		
• MMLU Benchmark (26 topics)	5.0E+23	280B		
• Grounded conceptual mappings	3.1E+23	175B	GPT-3	Patel & Pavlick (2022)
• MMLU Benchmark (30 topics)	5.0E+23	70B	Chinchilla	Hoffmann et al. (2022)
• Word in Context (WiC) benchmark	2.5E+24	540B	PaLM	Chowdhery et al. (2022)
• Many BIG-Bench tasks (see Appendix E)	Many	Many	Many	BIG-Bench (2022)
Augmented prompting abilities				
• Instruction following (finetuning)	1.3E+23	68B	FLAN	Wei et al. (2022a)
• Scratchpad: 8-digit addition (finetuning)	8.9E+19	40M	LaMDA	Nye et al. (2021)
• Using open-book knowledge for fact checking	1.3E+22	7.1B	Gopher	Rae et al. (2021)
• Chain-of-thought: Math word problems	1.3E+23	68B	LaMDA	Wei et al. (2022b)
• Chain-of-thought: StrategyQA	2.9E+23	62B	PaLM	Chowdhery et al. (2022)
• Differentiable search index	3.3E+22	11B	T5	Tay et al. (2022b)
• Self-consistency decoding	1.3E+23	68B	LaMDA	Wang et al. (2022b)
• Leveraging explanations in prompting	5.0E+23	280B	Gopher	Lampinen et al. (2022)
• Least-to-most prompting	3.1E+23	175B	GPT-3	Zhou et al. (2022)
• Zero-shot chain-of-thought reasoning	3.1E+23	175B	GPT-3	Kojima et al. (2022)
• Calibration via P(True)	2.6E+23	52B	Anthropic	Kadavath et al. (2022)
• Multilingual chain-of-thought reasoning	2.9E+23	62B	PaLM	Shi et al. (2022)
• Ask me anything prompting	1.4E+22	6B	EleutherAI	Arora et al. (2022)

Why Elbow shape / emergent pattern?

- 1. For certain tasks, there may be natural intuitions for why emergence requires a model larger than a particular threshold scale. For instance, if a multi-step reasoning task requires l steps of sequential computation, this might require a model with a depth of at least O (l) layers

- 2. more parameters and more training enable better memorization that could be helpful for tasks requiring world knowledge.4 As an example, good performance on closed-book question-answering may require a model with enough parameters to capture the compressed knowledge base itself (though language model-based compressors can have higher compression ratios than conventional compressors (Bellard, 2021))

LLM for All (plus promoting) vs. Task specific model

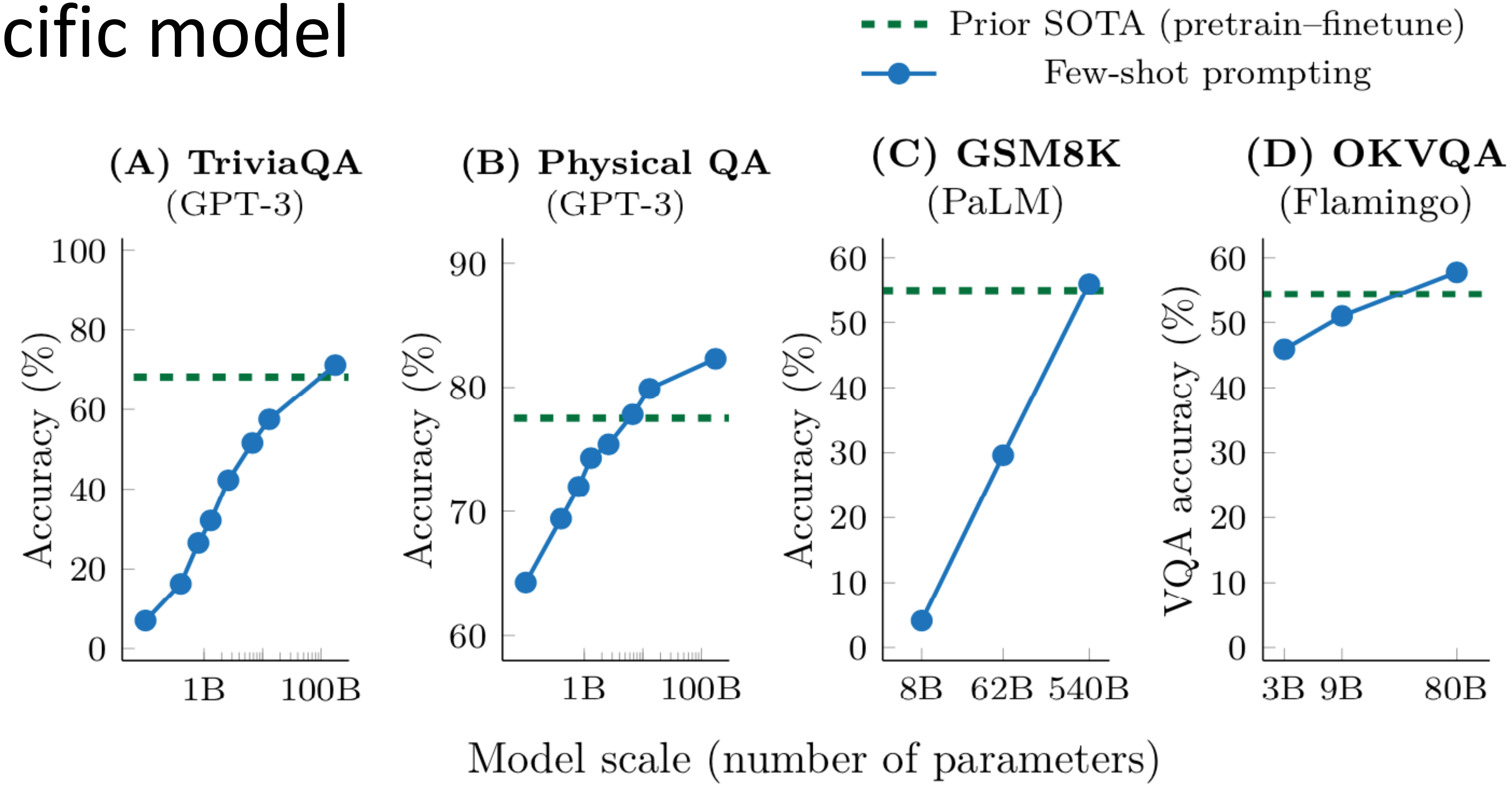


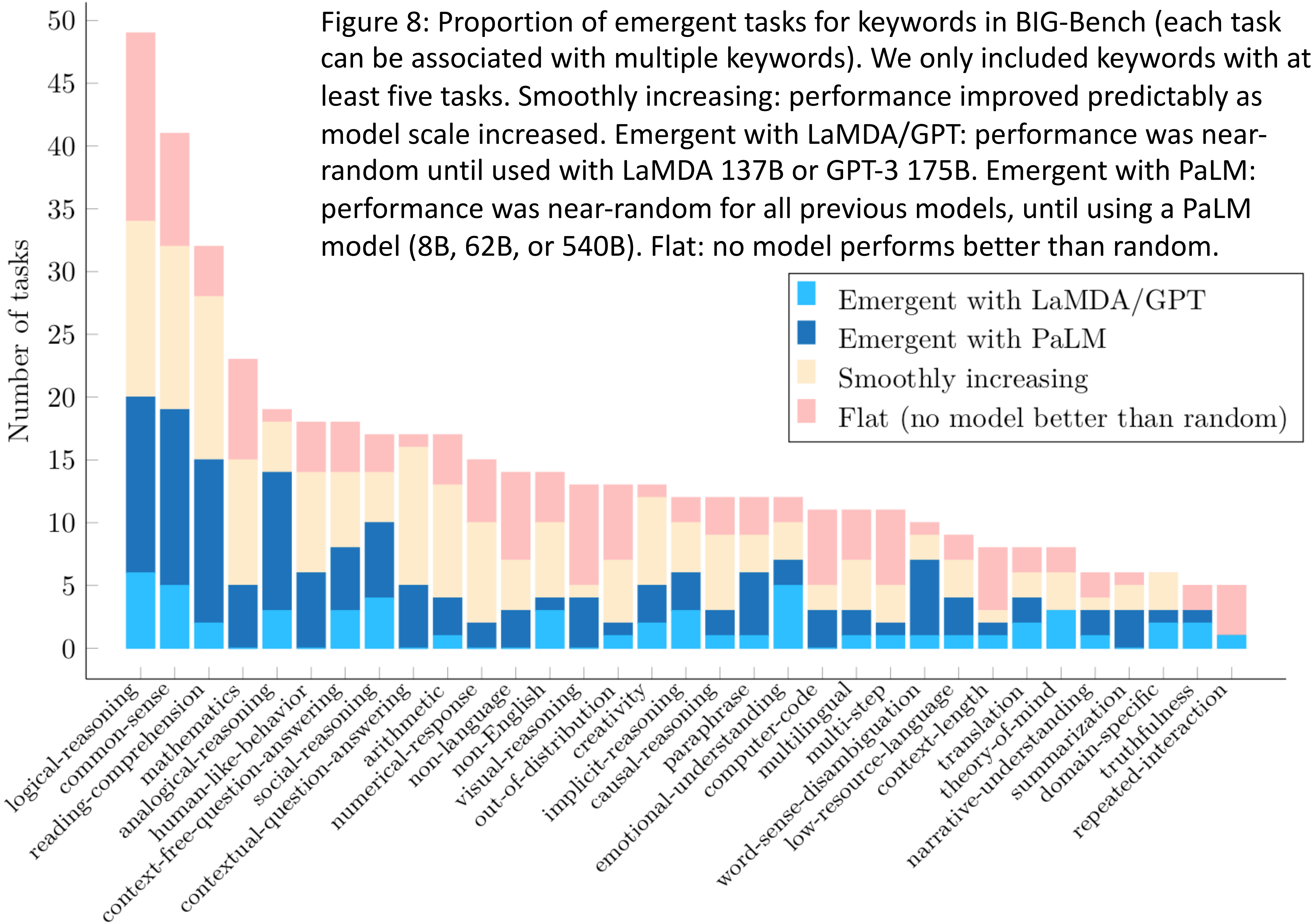
Figure 13: On some benchmarks, task-general models (not explicitly trained to perform a task) surpass prior state-of-the-art performance held by a task-specific model. A & B: Brown et al. (2020). C: Chowdhery et al. (2022). D: Alayrac et al. (2022).

https://github.com/google/BIG-bench/blob/main/bigbench/benchmark_tasks/keywords_to_tasks.md#big-bench-lite

The Beyond the Imitation Game Benchmark (BIG-bench) is a *collaborative* benchmark intended to probe large language models and extrapolate their future capabilities. The *more than 200* tasks included in BIG-bench are summarized by keyword [here](#), and by task name [here](#). A paper introducing the benchmark, including evaluation results on large language models, is currently in preparation.

Keyword	Number of tasks	Description
traditional NLP tasks		
contextual question-answering	22	identifying the meaning of a particular word/sentence in a passage
context-free question answering	24	responses rely on model's knowledge base, but not on context provided during query time
reading comprehension	36	a superset of contextual question-answering, measuring the degree to which a model understands the content of a text block
conversational question answering	1	a superset of reading comprehension, measuring the degree to which a model understands the content of a text block and a conversation
summarization	8	involves summarizing a block of text
paraphrase	14	express the same meaning using different words
text simplification	1	express the same meaning using simpler vocabulary
word sense disambiguation	11	identifying the meaning of a word based upon the context it appears
coreference resolution	0 🚧	finding all expressions that refer to the same entity in a text
question generation	2	tests model's ability to generate useful and sensible questions
narrative understanding	7	tests model's ability to understand language beyond surface level reasoning
dialogue system	1	measures model's ability to perform language understanding or generation on a user-to-machine conversation
memorization	5	tasks that require memorization of data from the pre-training set.
morphology	1	tests model's ability to solve challenges related to segmentation and construction of words
translation	10	the task involves translating between languages
writing style	2	measures model's ability to examine a text's writing style rather than its semantic meaning
grammar	2	tests model's ability to handle particular grammatical phenomena in the input or in the

Figure 8: Proportion of emergent tasks for keywords in BIG-Bench (each task can be associated with multiple keywords). We only included keywords with at least five tasks. Smoothly increasing: performance improved predictably as model scale increased. Emergent with LaMDA/GPT: performance was near-random until used with LaMDA 137B or GPT-3 175B. Emergent with PaLM: performance was near-random for all previous models, until using a PaLM model (8B, 62B, or 540B). Flat: no model performs better than random.



logic, math, code		
algorithms	5	measures the ability of a model to execute algorithms
logical reasoning	59	measures the ability of a model to reason about its inputs (eg, solve a word problem)
implicit reasoning	12	measures model's ability to infer implicit reasoning paths
mathematics	28	measures model's ability to perform mathematics of any type (see sub-types below)
arithmetic	22	measures model's ability to perform arithmetic
algebra	6	measures model's ability to perform algebra
mathematical proof	3	measures model's ability to derive or understand a mathematical proof
decomposition	4	tests model's ability to break problems down into simpler subproblems
fallacy	4	measure's model's ability to distinguish correct from fallacious reasoning
negation	4	measure's model's ability to understand negation
computer code	12	the task involves inputs or outputs that are computer code
semantic parsing	2	measure's model's ability to parse semantics of natural-language utterances
probabilistic reasoning	1	the task involves probing the model’s ability to reason in the face of uncertainty

understanding the world		
causal reasoning	17	measures ability to reason about cause and effect
consistent identity	4	tests model's ability to apply consistent attributes to objects or agents during extended text generation
physical reasoning	2	measures the ability of a model to reason about its inputs using basic physics intuition of how objects interact
common sense	48	measures ability to make judgements that humans would consider “common sense”
visual reasoning	14	measures model's ability to solve problems that a human would be likely to solve by visual reasoning
understanding humans		
theory of mind	10	tests whether model demonstrates a theory of mind
emotional understanding	16	tests model's ability to identify or understand human emotion
social reasoning	19	tests model's ability to interpret or reason about human social interactions
gender prediction	3	predicts the implicit gender information when prompted with gender-specific terms or Names
intent recognition	2	predicts the intent of a user utterance
humor	2	measures the model's ability to recognize humor in text
figurative language	3	tasks that measure model's ability to work with figurative language (e.g. metaphors, sarcasm).

scientific and technical understanding		
biology	2	measure's model's ability to understand biological properties
chemistry	2	knowledge of chemistry is useful for solving these tasks
physics	4	knowledge of physics is useful for solving these tasks
medicine	3	tests model's ability to perform tasks related to medicine
domain specific	9	test the ability to understand domain-specific knowledge
mechanics of interaction with model		
self play	5	involves multiple copies of the model interacting with each other
self evaluation	3	involves using the model's own judgment of its performance to score it
multiple choice	148	involves multiple choice responses, or assigning log probabilities to a list of specific allowed outputs. This includes programmatic as well as json tasks.
free response	84	involves the model generating unconstrained textual responses (each model interaction will be either `multiple choice` or `free response`, but a task can involve many interactions of both types)
game play	10	the task corresponds to a human game
repeated interaction	10	the task involves repeated interaction with the language model, rather than production of a single shot output
non-language	16	the task involves inputs or outputs that are not language or numbers (e.g., interpreting or generating ascii art images, or reading DNA sequences)
numerical response	19	the model's response should consist of numeric digits

targeting common language model technical limitations		
context length	13	measures ability to handle long context
multi-step	12	measures ability to perform a task that requires the model to internally perform many sequential steps before outputing a token
out of distribution	16	task probes a task which is designed to be very dissimilar from the likely training corpus
instructions	2	the ability to follow natural language instructions
tokenization	3	task probes abilities potentially obfuscated by model tokenization
paragraph	2	the task involves processing data at paragraph level, where each paragraph is coherent, semantically distinct text
pro-social behavior		
alignment	4	measures whether model behavior matches human preferences and values that are hard to define or formalize
social bias	9	measures changes in model responses depending on the social group a subject belongs to
racial bias	4	sub-type of social bias, exploring the impact of race
gender bias	9	sub-type of social bias, exploring the impact of gender
religious bias	5	sub-type of social bias, exploring the impact of religion
political bias	0 🚧	sub-type of social bias, exploring the impact of political affiliation
toxicity	1	measures the model's ability to identify text as toxic (rude, profane, hateful, or disrespecting) in nature, or to respond appropriately to toxic text.
inclusion	1	measures the model's ability to generate text that is inclusive with regard to social attributes such as gender or race

More on LLM

Many Large Scale PreTrained Language Model

- Basics (GPT, BERT, T5)
- PaLM
  - (decoder-only trained with next-token prediction)
- BLOOM
  - BLOOM is essentially similar to GPT3 (auto-regressive model for next token prediction), but has been trained on 46 different languages and 13 programming languages.
- Flan-PaLM / Flan-T5
- Many many new recent LLMs on huggingface: Llama, Mistral

PaLM: Scaling Language Modeling with Pathways

Aakanksha Chowdhery, et al, Erica Noah Fiedel

Large language models have been shown to achieve remarkable performance across a variety of natural language tasks using few-shot learning, which drastically reduces the number of task-specific training examples needed to adapt the model to a particular application. To further our understanding of the impact of scale on few-shot learning, we trained a 540-billion parameter, densely activated, Transformer language model, which we call Pathways Language Model PaLM. We trained PaLM on 6144 TPU v4 chips using Pathways, a new ML system which enables highly efficient training across multiple TPU Pods. We demonstrate continued benefits of scaling by achieving state-of-the-art few-shot learning results on hundreds of language understanding and generation benchmarks. On a number of these tasks, PaLM 540B achieves breakthrough performance, outperforming the finetuned state-of-the-art on a suite of multi-step reasoning tasks, and outperforming average human performance on the recently released BIG-bench benchmark. A significant number of BIG-bench tasks showed discontinuous improvements from model scale, meaning that performance steeply increased as we scaled to our largest model. PaLM also has strong capabilities in multilingual tasks and source code generation, which we demonstrate on a wide array of benchmarks. We additionally provide a comprehensive analysis on bias and toxicity, and study the extent of training data memorization with respect to model scale. Finally, we discuss the ethical considerations related to large language models and discuss potential mitigation strategies.

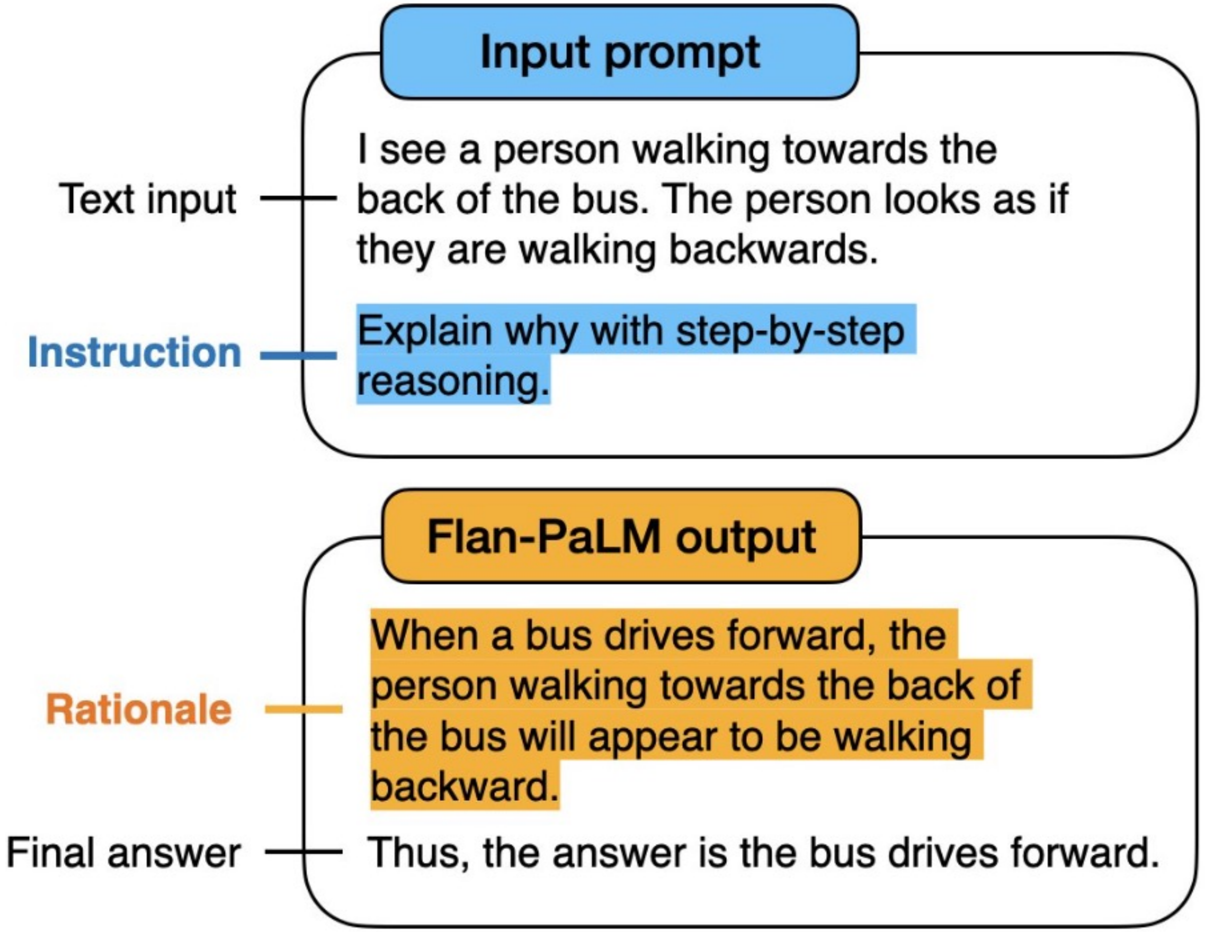
[Submitted on 20 Oct 2022 (v1), last revised 21 Oct 2022 (this version, v2)]

Scaling Instruction-Finetuned Language Models

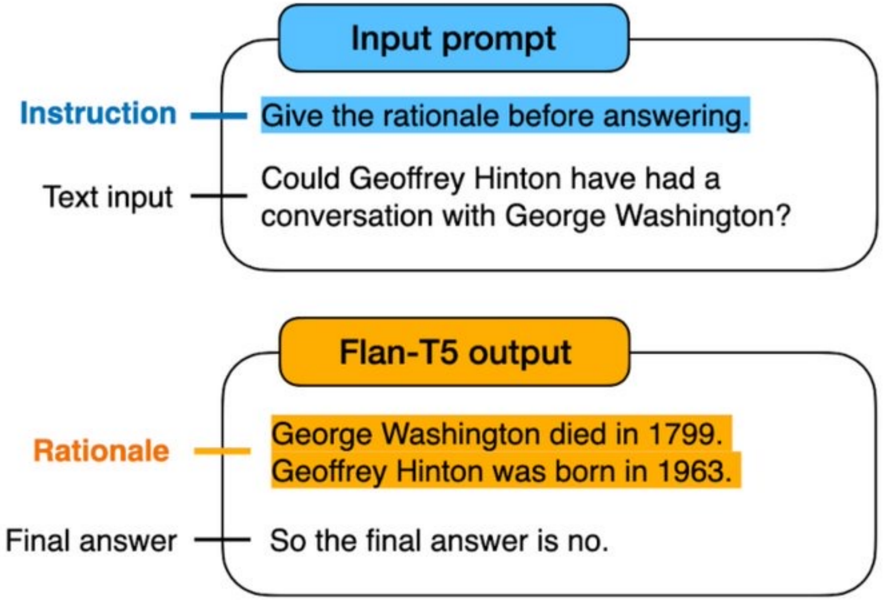
Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, Jason Wei

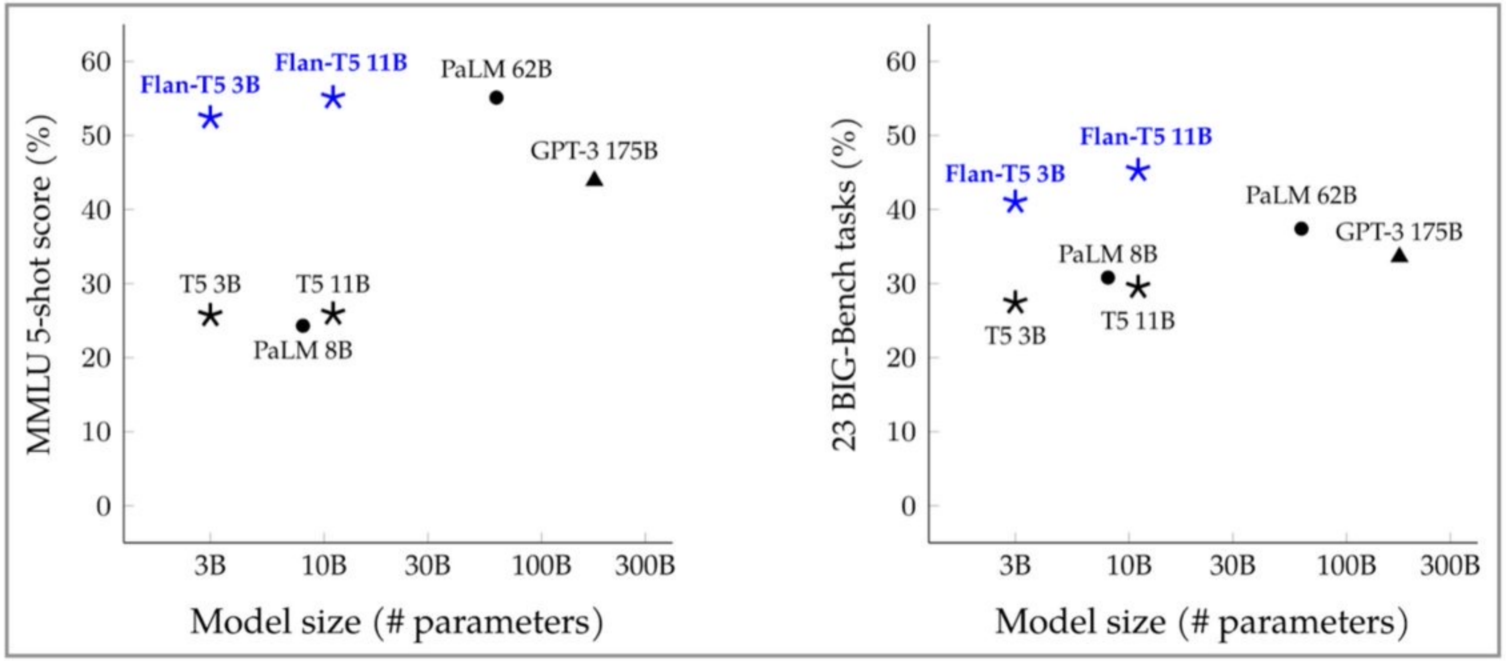
Finetuning language models on a collection of datasets phrased as instructions has been shown to improve model performance and generalization to unseen tasks. In this paper we explore instruction finetuning with a particular focus on (1) scaling the number of tasks, (2) scaling the model size, and (3) finetuning on chain-of-thought data. We find that instruction finetuning with the above aspects dramatically improves performance on a variety of model classes (PaLM, T5, U-PaLM), prompting setups (zero-shot, few-shot, CoT), and evaluation benchmarks (MMLU, BBH, TyDiQA, MGSM, open-ended generation). For instance, Flan-PaLM 540B instruction-finetuned on 1.8K tasks outperforms PALM 540B by a large margin (+9.4% on average). Flan-PaLM 540B achieves state-of-the-art performance on several benchmarks, such as 75.2% on five-shot MMLU. We also publicly release Flan-T5 checkpoints, which achieve strong few-shot performance even compared to much larger models, such as PaLM 62B. Overall, instruction finetuning is a general method for improving the performance and usability of pretrained language models.

Chain-of-thought prompting is highly effective but having to write few-shot exemplars can be tedious and zero-shot CoT doesn't always work well. Our CoT finetuning significantly improves zero-shot reasoning abilities, such as on commonsense reasoning.




Flan-T5


Input prompt
Instruction
Give the rationale before answering.
Text input
Could Geoffrey Hinton have had a conversation with George Washington?
Flan-T5 output
Rationale
George Washington died in 1799. Geoffrey Hinton was born in 1963.
Final answer
So the final answer is no.


MMLU 5-shot score (%)
60
50
40
30
20
10
0
Flan-T5 3B
Flan-T5 11B
PaLM 62B
GPT-3 175B
T5 3B
T5 11B
PaLM 8B
3B
10B
30B
100B
300B
Model size (# parameters)
23 BIG-Bench tasks (%)
60
50
40
30
20
10
0
Flan-T5 11B
Flan-T5 3B
PaLM 62B
GPT-3 175B
PaLM 8B
T5 3B
T5 11B
3B
10B
30B
100B
300B
Model size (# parameters)

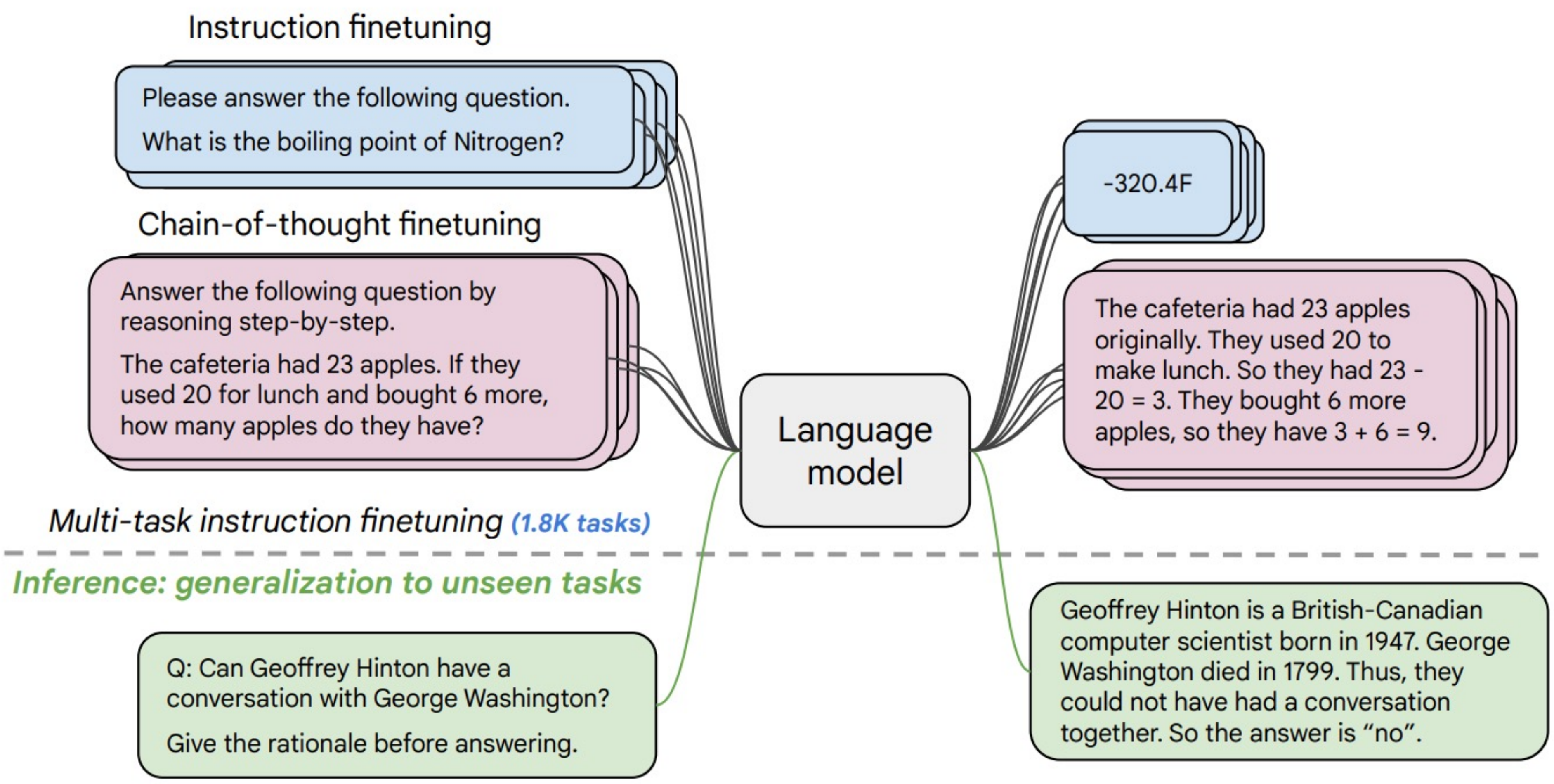
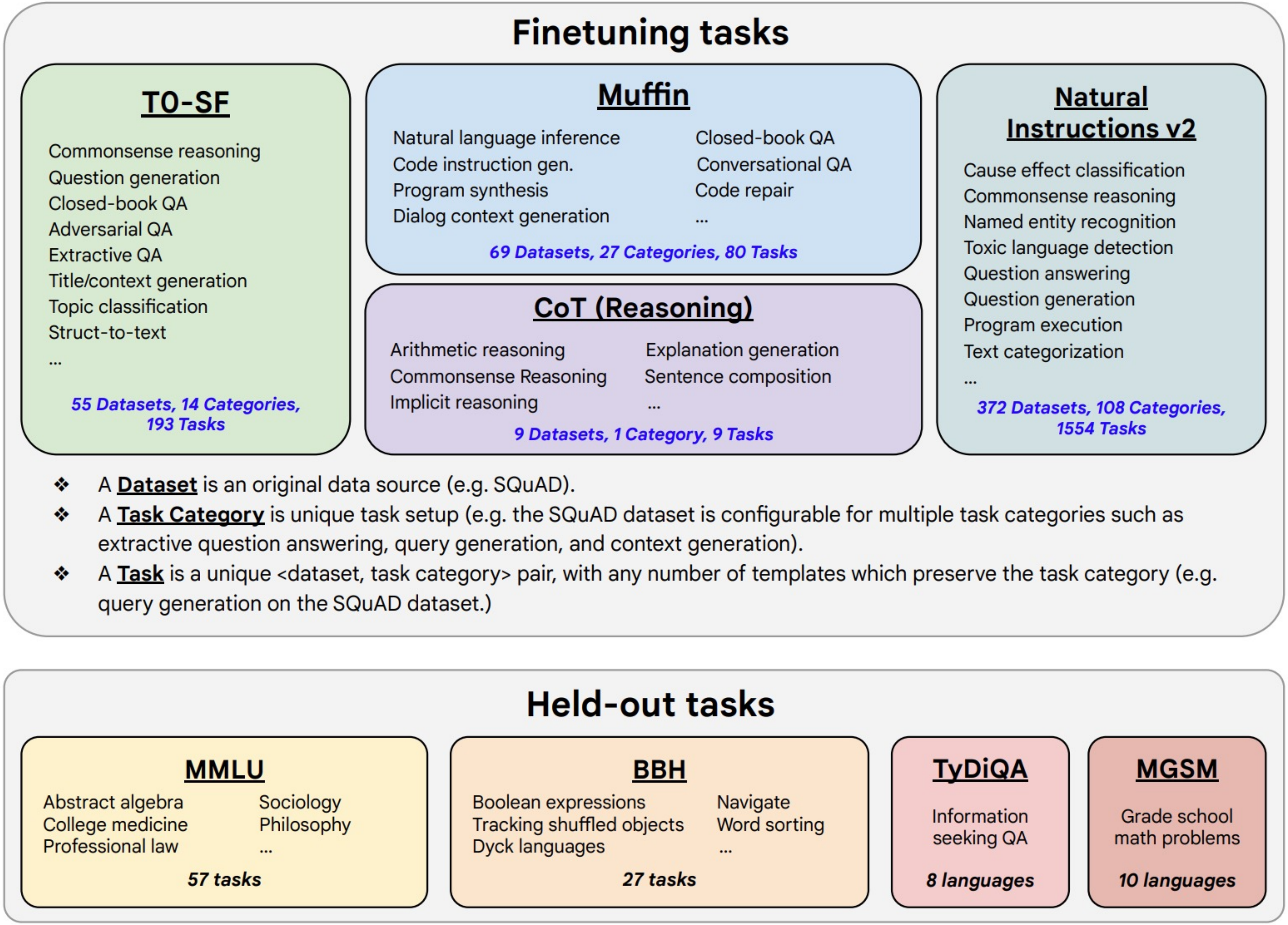


Figure 1: We finetune various language models on 1.8K tasks phrased as instructions, and evaluate them on unseen tasks. We finetune both with and without exemplars (i.e., zero-shot and few-shot) and with and without chain-of-thought, enabling generalization across a range of evaluation scenarios.


Finetuning tasks
T0-SF
Commonsense reasoning
Question generation
Closed-book QA
Adversarial QA
Extractive QA
Title/context generation
Topic classification
Struct-to-text
...
55 Datasets, 14 Categories, 193 Tasks
Muffin
Natural language inference
Code instruction gen.
Program synthesis
Dialog context generation
Closed-book QA
Conversational QA
Code repair
...
69 Datasets, 27 Categories, 80 Tasks
CoT (Reasoning)
Arithmetic reasoning
Commonsense Reasoning
Implicit reasoning
Explanation generation
Sentence composition
...
9 Datasets, 1 Category, 9 Tasks
Natural Instructions v2
Cause effect classification
Commonsense reasoning
Named entity recognition
Toxic language detection
Question answering
Question generation
Program execution
Text categorization
...
372 Datasets, 108 Categories, 1554 Tasks
A Dataset is an original data source (e.g. SQuAD).
A Task Category is unique task setup (e.g. the SQuAD dataset is configurable for multiple task categories such as extractive question answering, query generation, and context generation).
A Task is a unique <dataset, task category> pair, with any number of templates which preserve the task category (e.g. query generation on the SQuAD dataset.)
Held-out tasks
MMLU
Abstract algebra
College medicine
Professional law
Sociology
Philosophy
...
57 tasks
BBH
Boolean expressions
Tracking shuffled objects
Dyck languages
Navigate
Word sorting
...
27 tasks
TyDiQA
Information seeking QA
8 languages
MGSM
Grade school math problems
10 languages

Params	Model	Arhitecture	pre-training Objective	Pretrain FLOPs	Finetune FLOPs	% Finetune Compute
80M	Flan-T5-Small	encoder-decoder	span corruption	1.8E+20	2.9E+18	1.6%
250M	Flan-T5-Base	encoder-decoder	span corruption	6.6E+20	9.1E+18	1.4%
780M	Flan-T5-Large	encoder-decoder	span corruption	2.3E+21	2.4E+19	1.1%
3B	Flan-T5-XL	encoder-decoder	span corruption	9.0E+21	5.6E+19	0.6%
11B	Flan-T5-XXL	encoder-decoder	span corruption	3.3E+22	7.6E+19	0.2%
8B	Flan-PaLM	decoder-only	causal LM	3.7E+22	1.6E+20	0.4%
62B	Flan-PaLM	decoder-only	causal LM	2.9E+23	1.2E+21	0.4%
540B	Flan-PaLM	decoder-only	causal LM	2.5E+24	5.6E+21	0.2%
62B	Flan-cont-PaLM	decoder-only	causal LM	4.8E+23	1.8E+21	0.4%
540B	Flan-U-PaLM	decoder-only	prefix LM + span corruption	2.5E+23	5.6E+21	0.2%

Table 2: Across several models, instruction finetuning only costs a small amount of compute relative to pre-training. T5: Raffel et al. (2020). PaLM and cont-PaLM (also known as PaLM 62B at 1.3T tokens): Chowdhery et al. (2022). U-PaLM: Tay et al. (2022b).

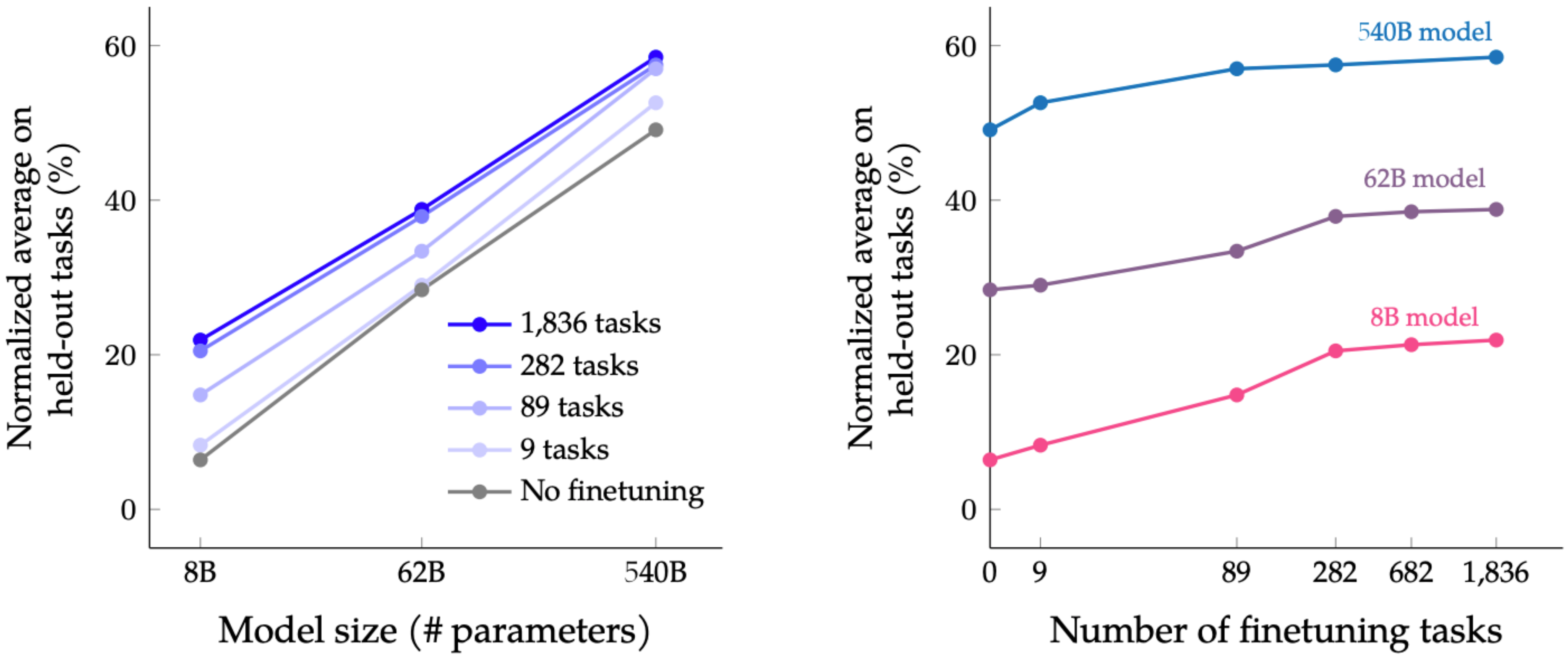


Figure 4: Scaling behavior of multi-task instruction finetuning with respect to model size (# parameters) and number of finetuning tasks. The x-axes are log scale. The benchmark suites are MMLU (57 tasks), BBH (23 tasks), TyDiQA (8 languages), and MGSM (10 languages). The evaluation metric on all four benchmark suites is few-shot prompted accuracy (exact match), where we take an unweighted average over all tasks. As an aggregate metric we report the normalized average of MMLU-direct, MMLU-CoT, BBH-direct, BBH-CoT, TyDiQA, and MGSM. These evaluation benchmarks are held-out (not included in the finetuning data).

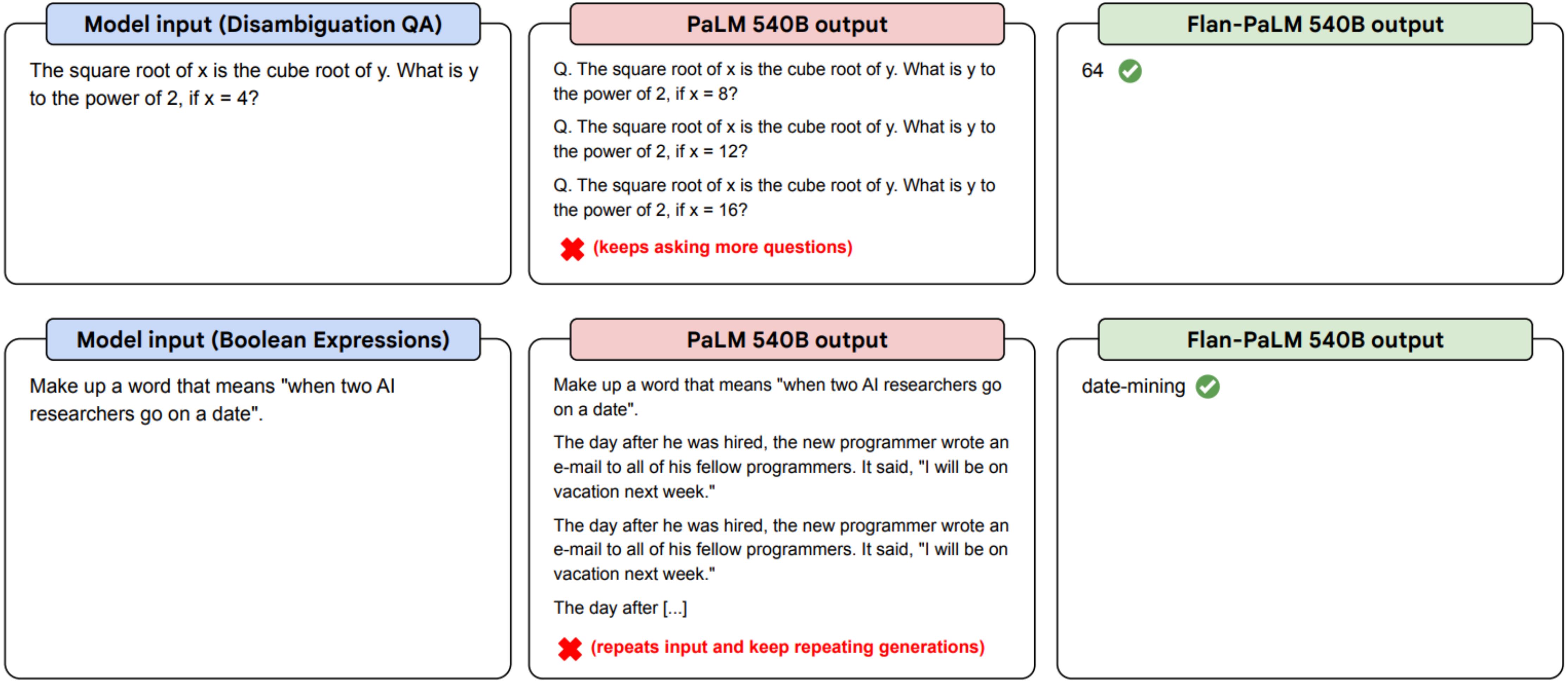


Figure 9: Some examples for zero-shot prompting, comparing PaLM and Flan-PaLM. PaLM struggles with repetitions and not replying to instructions in the zero-shot setting (though these errors can be mitigated by using few-shot exemplars).

[Submitted on 16 Aug 2021 (v1), last revised 12 Jul 2022 (this version, v3)]

On the Opportunities and Risks of Foundation Models

Authors: Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh , et al. (89 additional authors not shown)

Abstract:AI is undergoing a paradigm shift with the rise of models (e.g., BERT, DALL-E, GPT-3) that are trained on broad data at scale and are adaptable to a wide range of downstream tasks. We call these models foundation models to underscore their critically central yet incomplete character. This report provides a thorough account of the opportunities and risks of foundation models, ranging from their capabilities (e.g., language, vision, robotics, reasoning, human interaction) and technical principles(e.g., model architectures, training procedures, data, systems, security, evaluation, theory) to their applications (e.g., law, healthcare, education) and societal impact (e.g., inequity, misuse, economic and environmental impact, legal and ethical considerations). Though foundation models are based on standard deep learning and transfer learning, their scale results in new emergent capabilities,and their effectiveness across so many tasks incentivizes homogenization. Homogenization provides powerful leverage but demands caution, as the defects of the foundation model are inherited by all the adapted models downstream. Despite the impending widespread deployment of foundation models, we currently lack a clear understanding of how they work, when they fail, and what they are even capable of due to their emergent properties. To tackle these questions, we believe much of the critical research on foundation models will require deep interdisciplinary collaboration commensurate with their fundamentally sociotechnical nature.

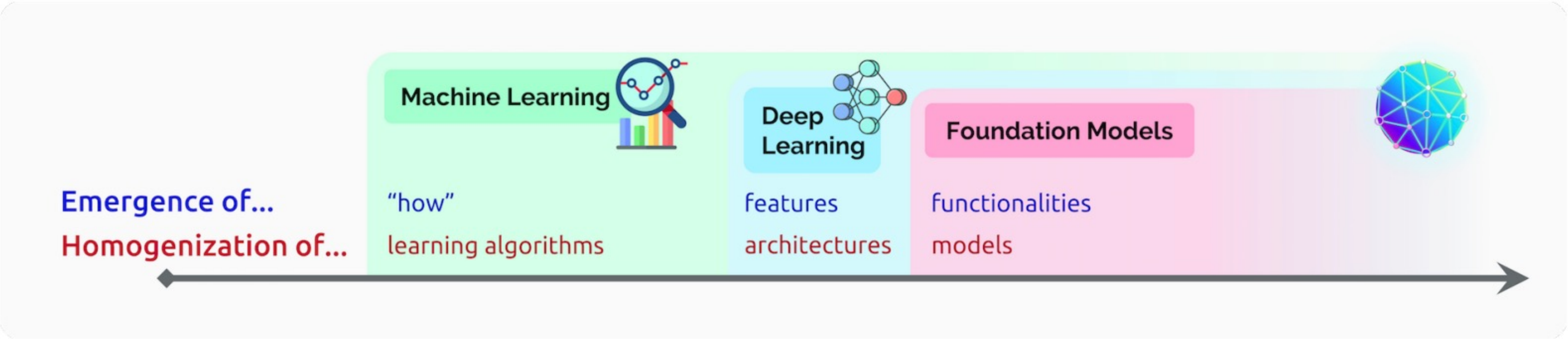
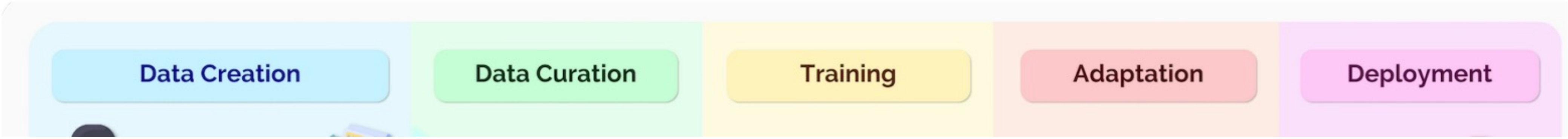
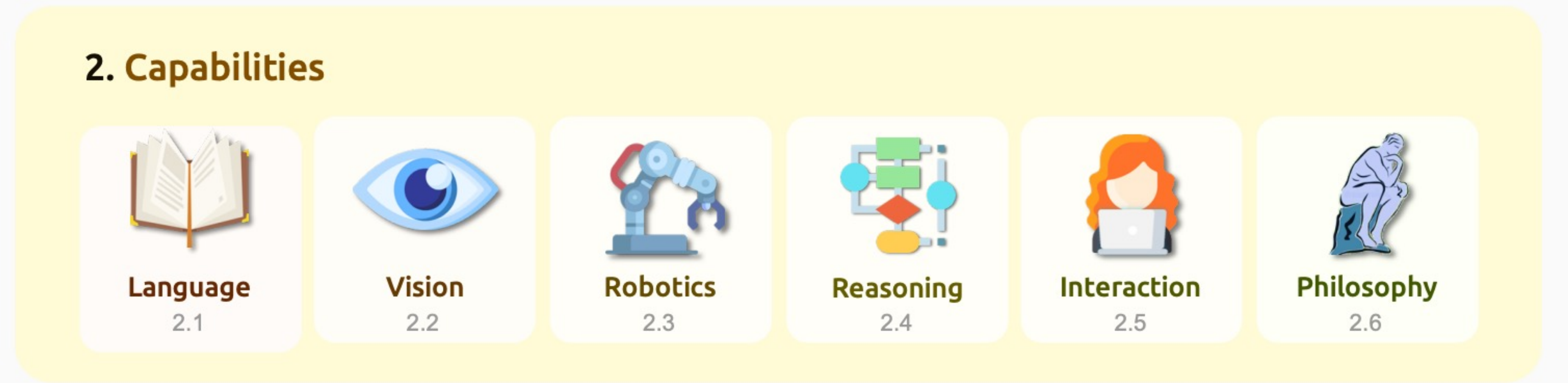


Fig. 1. The story of AI has been one of increasing *emergence* and *homogenization*. With the introduction of machine learning, *how* a task is performed emerges (is inferred automatically) from examples; with deep learning, the high-level features used for prediction emerge; and with foundation models, even advanced functionalities such as in-context learning emerge. At the same time, machine learning homogenizes learning algorithms (e.g., logistic regression), deep learning homogenizes model architectures (e.g., Convolutional Neural Networks), and foundation models homogenizes the model itself (e.g., GPT-3).



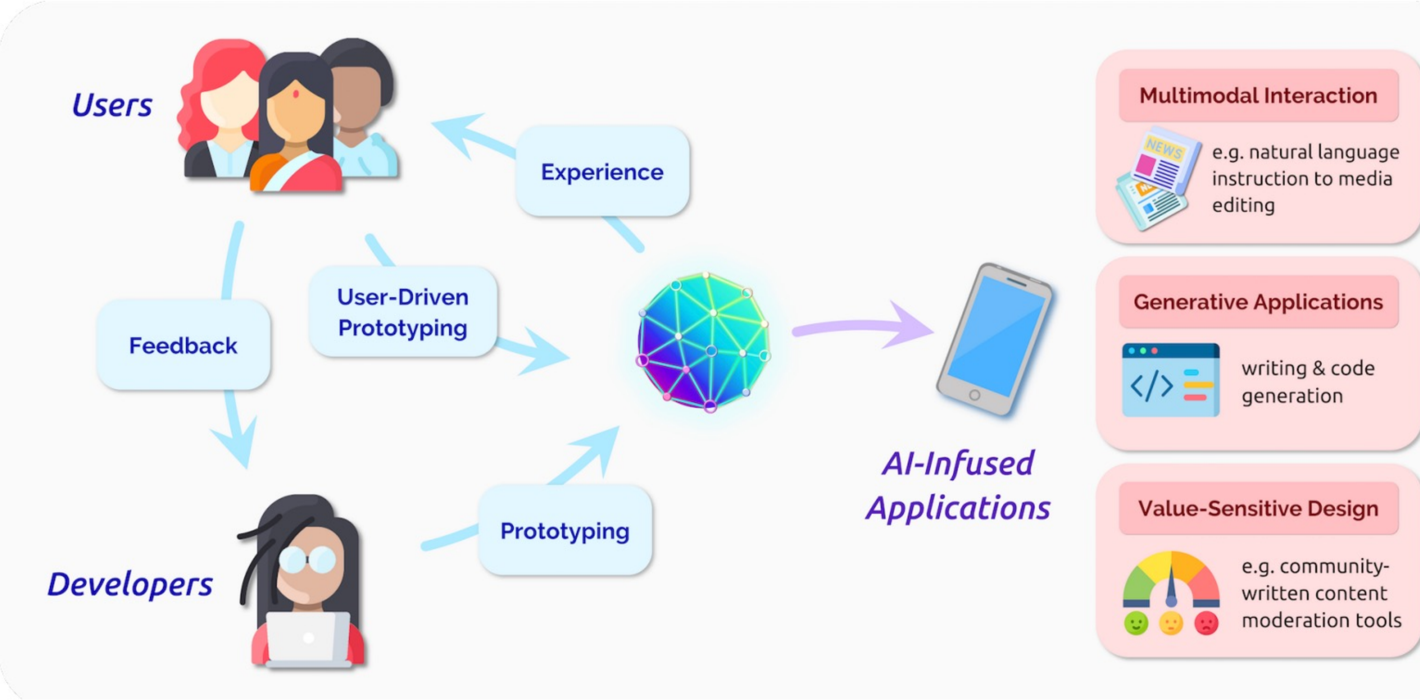

2. Capabilities
Language
2.1
Vision
2.2
Robotics
2.3
Reasoning
2.4
Interaction
2.5
Philosophy
2.6

For example: 2.5

- 5: Interaction. Foundation models show clear potential to transform the developer and user experience for AI systems: foundation models lower the difficulty threshold for prototyping and building AI applications due to their sample efficiency in adaptation, and raise the ceiling for novel user interaction due to their multimodal and generative capabilities.

- This provides a synergy we encourage going forward: developers can provide applications that better fit the user's needs and values, while introducing far more dynamic forms of interaction and opportunities for feedback.

- E.g. low-code / code-completion



4. Technology

- Modeling 4.1
- Training 4.2
- Adaptation 4.3
- Evaluation 4.4
- Systems 4.5
- Data 4.6
- Security 4.7
- Robustness 4.8
- AI Safety & Alignment 4.9
- Theory 4.10
- Interpretability 4.11

5. Society

- Inequity 5.1
- Misuse 5.2
- Environment 5.3
- Legality 5.4
- Economics 5.5
- Ethics 5.6