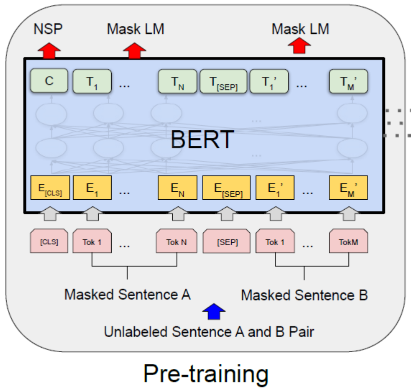
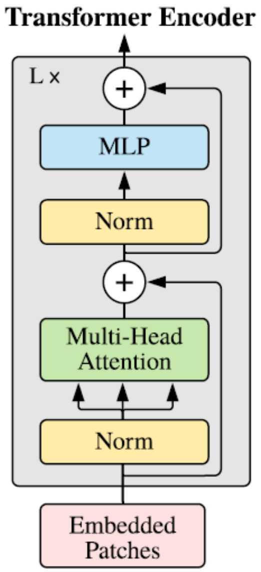
Dall-E

Foundation Models

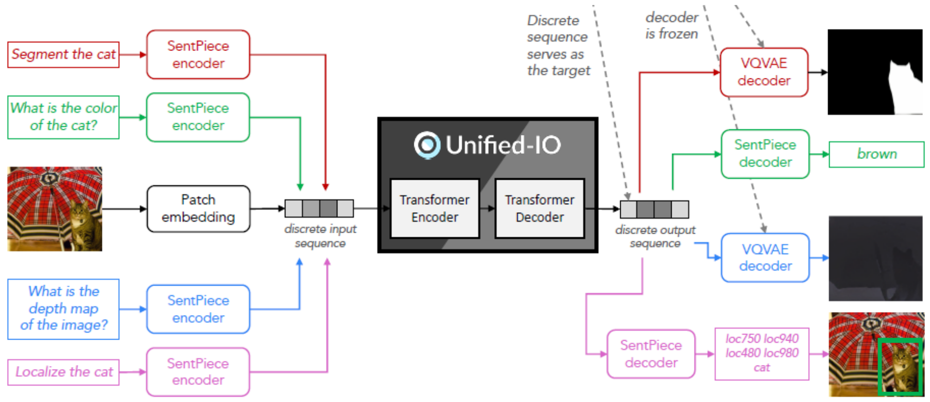
Applied Machine Learning
Derek Hoiem

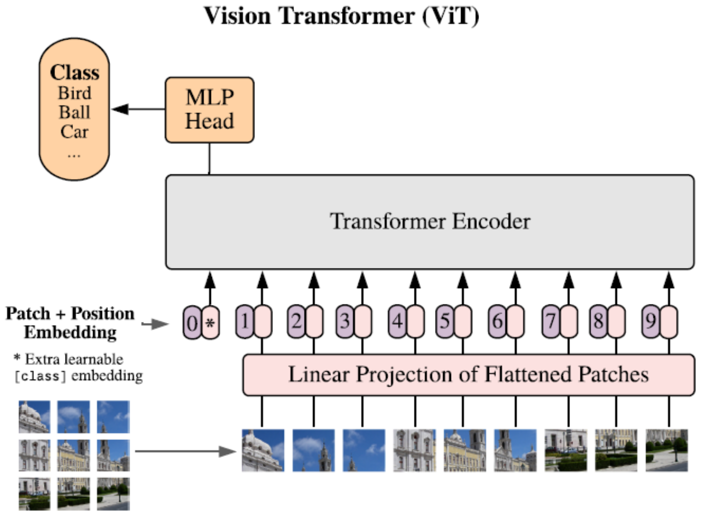
Last class: Transformer Models





Transformers are efficient, multi-modal data processors





This lecture

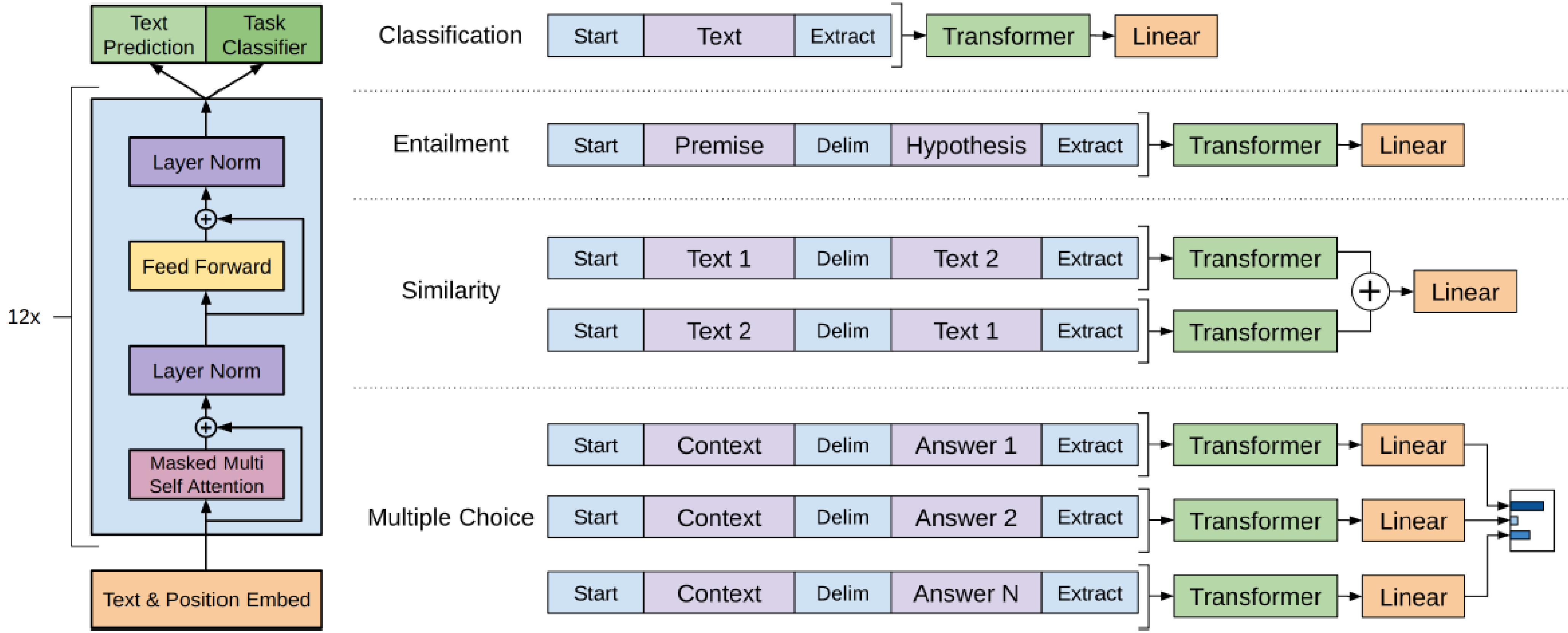
- **Foundation models**: Models that are trained on exorbitant data and compute on a broad task, often intended as a starting point for specialized models

- Key questions for foundation models are
  - How to train them (what architecture, what data, what objective)
  - How to apply them, e.g.
    - Zero-shot: apply to new tasks without any training examples for those specific tasks
    - Linear probe: train a linear model on the features
    - Fine-tune: adjust the entire network to perform better in the target task

- We previously saw two examples of foundation models suitable for fine-tuning
  - ImageNet pretrained models for vision
  - BERT for language

- We will now learn about two more famous models that can do zero shot
  - GPT: **G**enerative **P**retraining **M**odels for Language
  - CLIP: **C**ontrastive **L**anguage-**I**mage **P**retraining for Vision

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



GPT1 (2018)

- Pre-cursor to BERT (2019) that we discussed last class
- Similar architecture and training procedures
  - 117M parameters in GPT1 vs. 340M for BERT Large
- 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

$$L_1(\mathcal{U}) = \sum_i \log P(u_i | u_{i-k}, \ldots, u_{i-1}; \Theta)$$

- Fine-tuned for each task while also retaining the generative objective. Some tasks need to be processed in a special way
- Achieved state-of-art in 9 out of 12 tasks

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

Aims to create a 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.

The dominant approach to creating ML systems is to collect a dataset of training examples demonstrating correct behavior for a desired task, train a system to imitate these behaviors, and then test its performance on independent and identically distributed (IID) held-out examples. This has served well to make progress on narrow experts. But the often erratic behavior of captioning models (Lake et al., 2017), reading comprehension systems (Jia & Liang, 2017), and image classifiers (Alcorn et al., 2018) on the diversity and variety of possible inputs highlights some of the shortcomings of this approach.

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.”

PDF

GPT-2

- A general system should learn to model $P(output|input, task)$

- The task can be specified in natural language, so language tasks can be framed as sequence-to-sequence text processing

- Sequence-to-sequence: A problem formulated as receiving input in some modality and producing output some modality (instead of e.g. predicting probability for labels in a specific task)

GPT-2: Data and Training

- WebText Dataset: Created a new web scrape of pages linked from Reddit with at least 3 karma, as these should be of reasonable quality
  - Does not require additional manual annotation
  - Yields 8 million documents (40GB text) from before 2018 after de-duplication and cleaning
  - Removed Wikipedia, since it is commonly used in test sets

- GPT-2 is generatively trained on WebText data and not fine-tuned on anything else

GPT-2 Architecture and Model Sizes

- Architecture is basically the same as GPT-1 and BERT

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

GPT-2: Zero shot results

	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).

Perplexity (PPL) is 2^entropy; lower is better

- Achieves state-of-art in many tasks without tuning for them
- Performs much worse than state-of-art in summarization and translation (though can effectively translate word for word)

Question	Generated Answer	Correct	Probability
Who wrote the book the origin of species?	Charles Darwin	✓	83.4%
Who is the founder of the ubuntu project?	Mark Shuttleworth	✓	82.0%
Who is the quarterback for the green bay packers?	Aaron Rodgers	✓	81.1%
Panda is a national animal of which country?	China	✓	76.8%
Who came up with the theory of relativity?	Albert Einstein	✓	76.4%
When was the first star wars film released?	1977	✓	71.4%
What is the most common blood type in sweden?	A	✗	70.6%
Who is regarded as the founder of psychoanalysis?	Sigmund Freud	✓	69.3%
Who took the first steps on the moon in 1969?	Neil Armstrong	✓	66.8%
Who is the largest supermarket chain in the uk?	Tesco	✓	65.3%
What is the meaning of shalom in english?	peace	✓	64.0%
Who was the author of the art of war?	Sun Tzu	✓	59.6%
Largest state in the us by land mass?	California	✗	59.2%
Green algae is an example of which type of reproduction?	parthenogenesis	✗	56.5%
Vikram samvat calender is official in which country?	India	✓	55.6%
Who is mostly responsible for writing the declaration of independence?	Thomas Jefferson	✓	53.3%
What us state forms the western boundary of montana?	Montana	✗	52.3%
Who plays ser davos in game of thrones?	Peter Dinklage	✗	52.1%
Who appoints the chair of the federal reserve system?	Janet Yellen	✗	51.5%
State the process that divides one nucleus into two genetically identical nuclei?	mitosis	✓	50.7%
Who won the most mvp awards in the nba?	Michael Jordan	✗	50.2%
What river is associated with the city of rome?	the Tiber	✓	48.6%
Who is the first president to be impeached?	Andrew Johnson	✓	48.3%
Who is the head of the department of homeland security 2017?	John Kelly	✓	47.0%
What is the name given to the common currency to the european union?	Euro	✓	46.8%
What was the emperor name in star wars?	Palpatine	✓	46.5%
Do you have to have a gun permit to shoot at a range?	No	✓	46.4%
Who proposed evolution in 1859 as the basis of biological development?	Charles Darwin	✓	45.7%
Nuclear power plant that blew up in russia?	Chernobyl	✓	45.7%
Who played john connor in the original terminator?	Arnold Schwarzenegger	✗	45.2%

Table 5. The 30 most confident answers generated by GPT-2 on the development set of Natural Questions sorted by their probability according to GPT-2. None of these questions appear in WebText according to the procedure described in Section 4.

See many more examples in the paper

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science.

Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved.

Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow.

Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez.

Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic."

Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America.

While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, "In South America, such incidents seem to be quite common."

However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. "But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.

Continued log-linear improvement with model size

Conclusion: “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 amount of tasks without the need for explicit supervision.**”

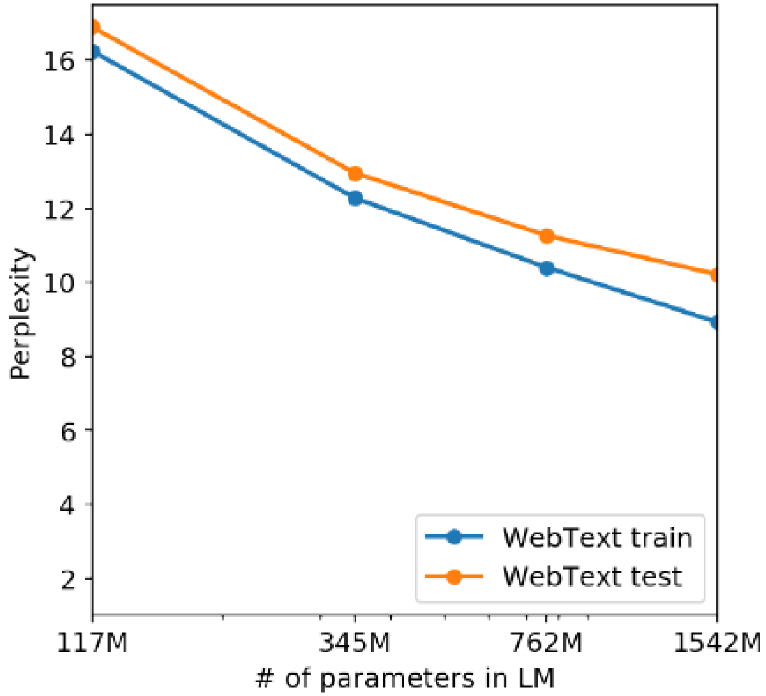
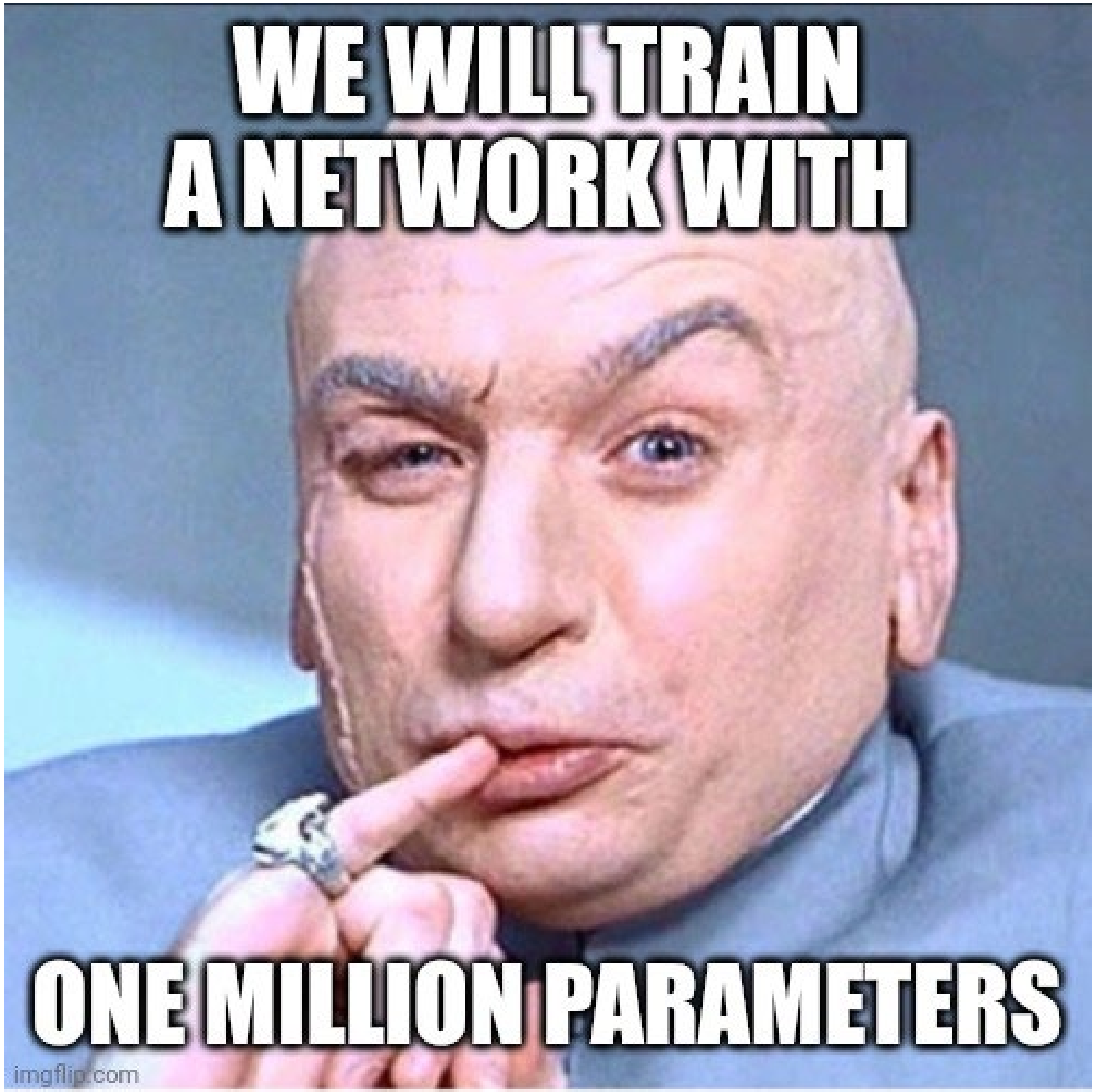
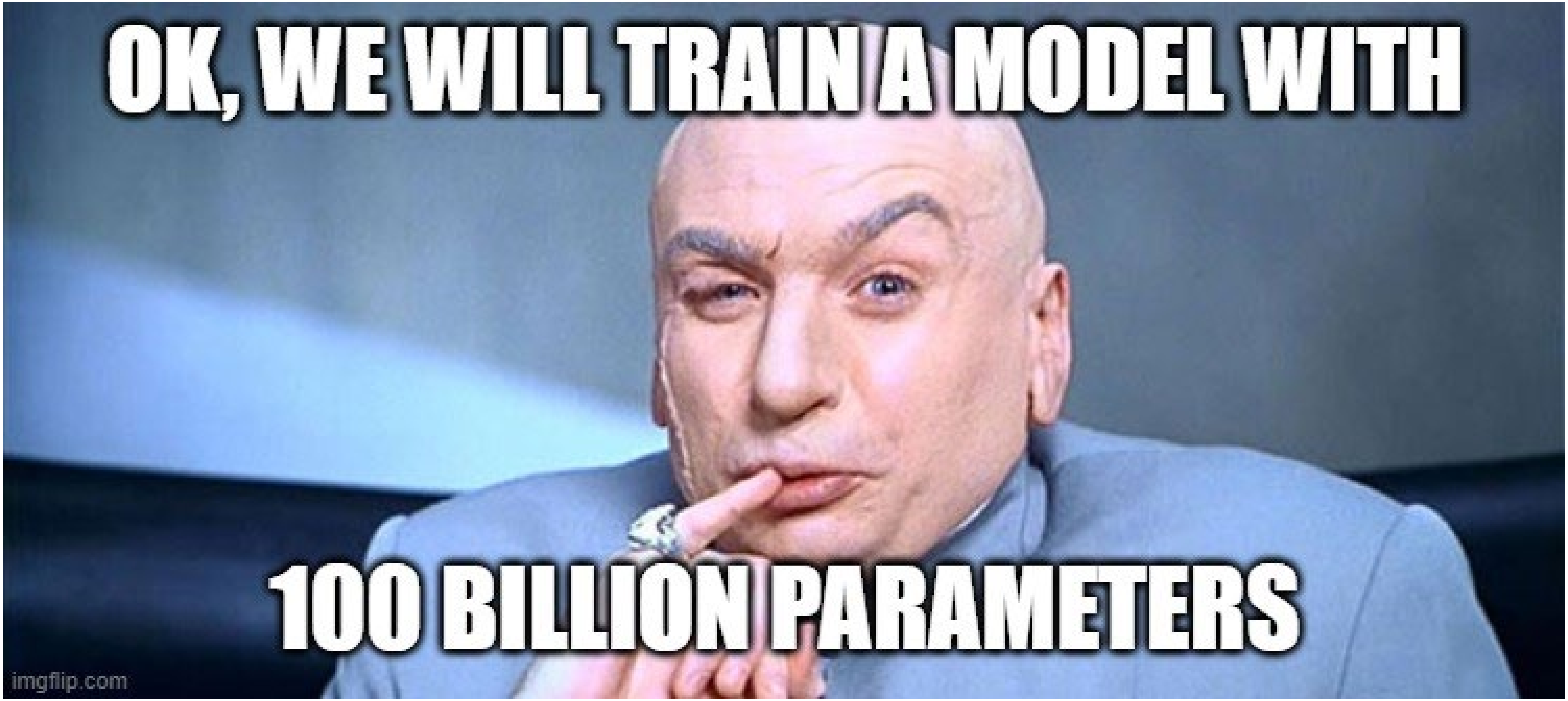


Figure 4. The performance of LMs trained on WebText as a function of model size.

In the OpenAI board room…



https://www.youtube.com/watch?v=EJR1H5tf5wE


OK, WE WILL TRAIN A MODEL WITH
100 BILLION PARAMETERS
imgflip.com

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 data

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

Training compute

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

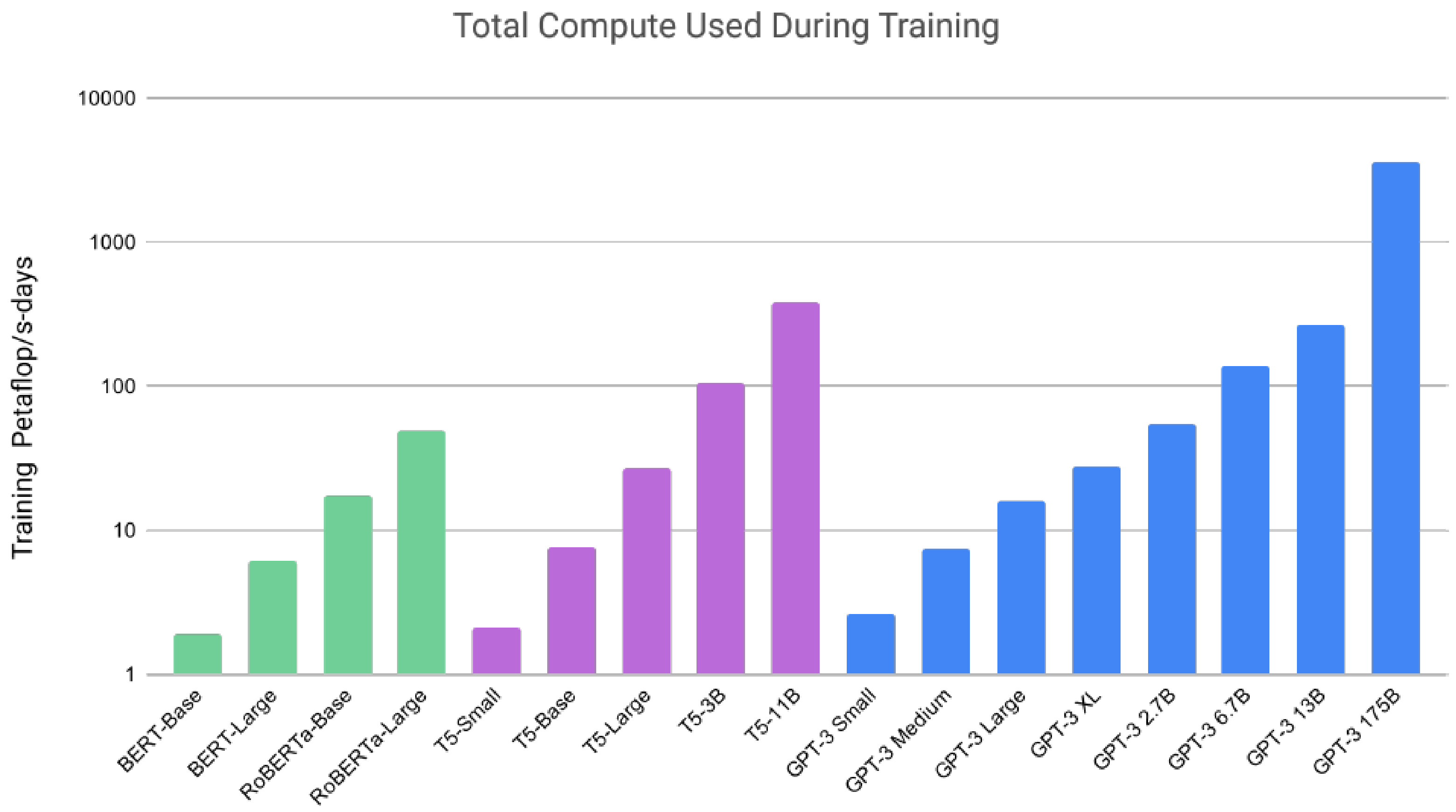
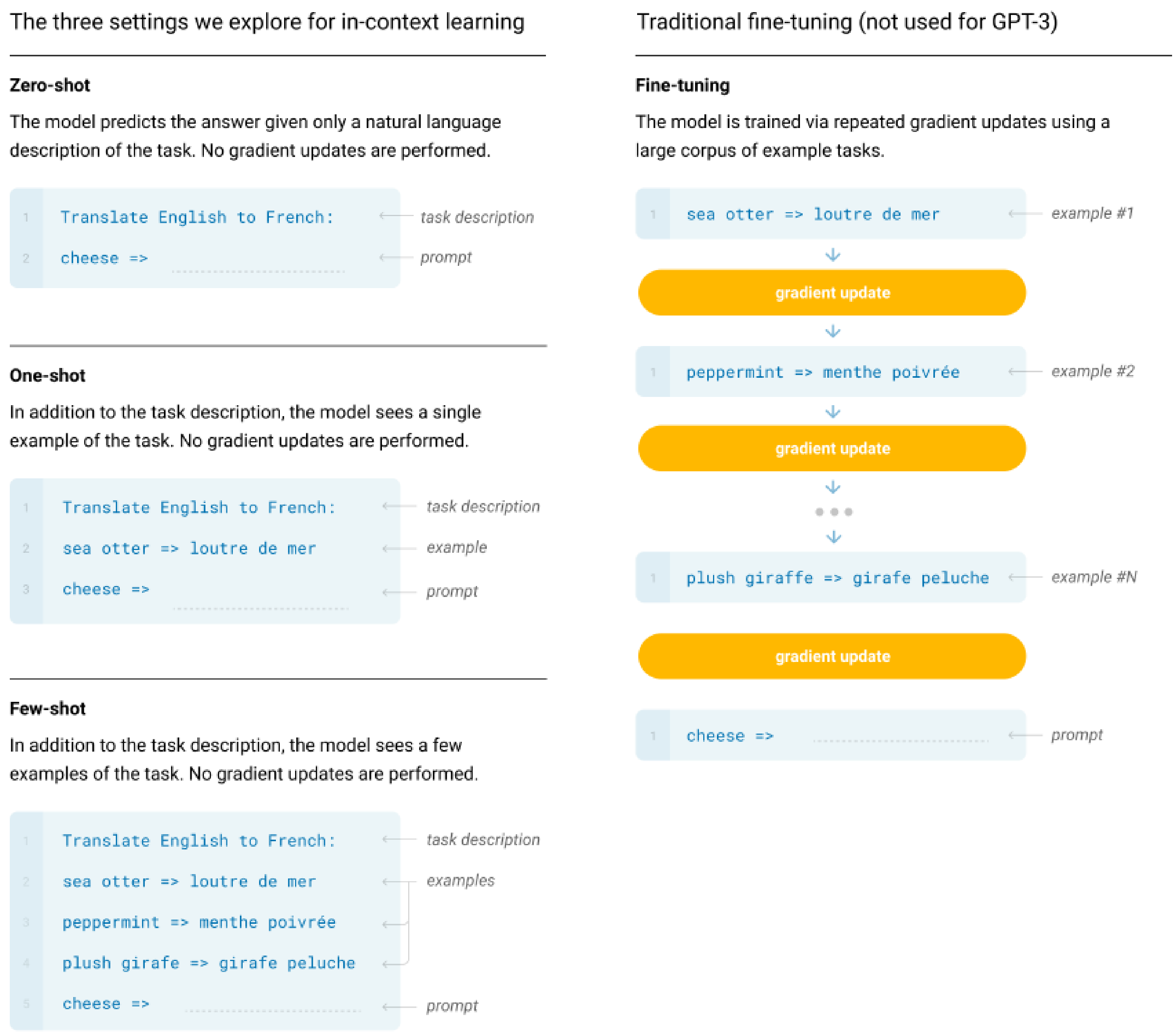
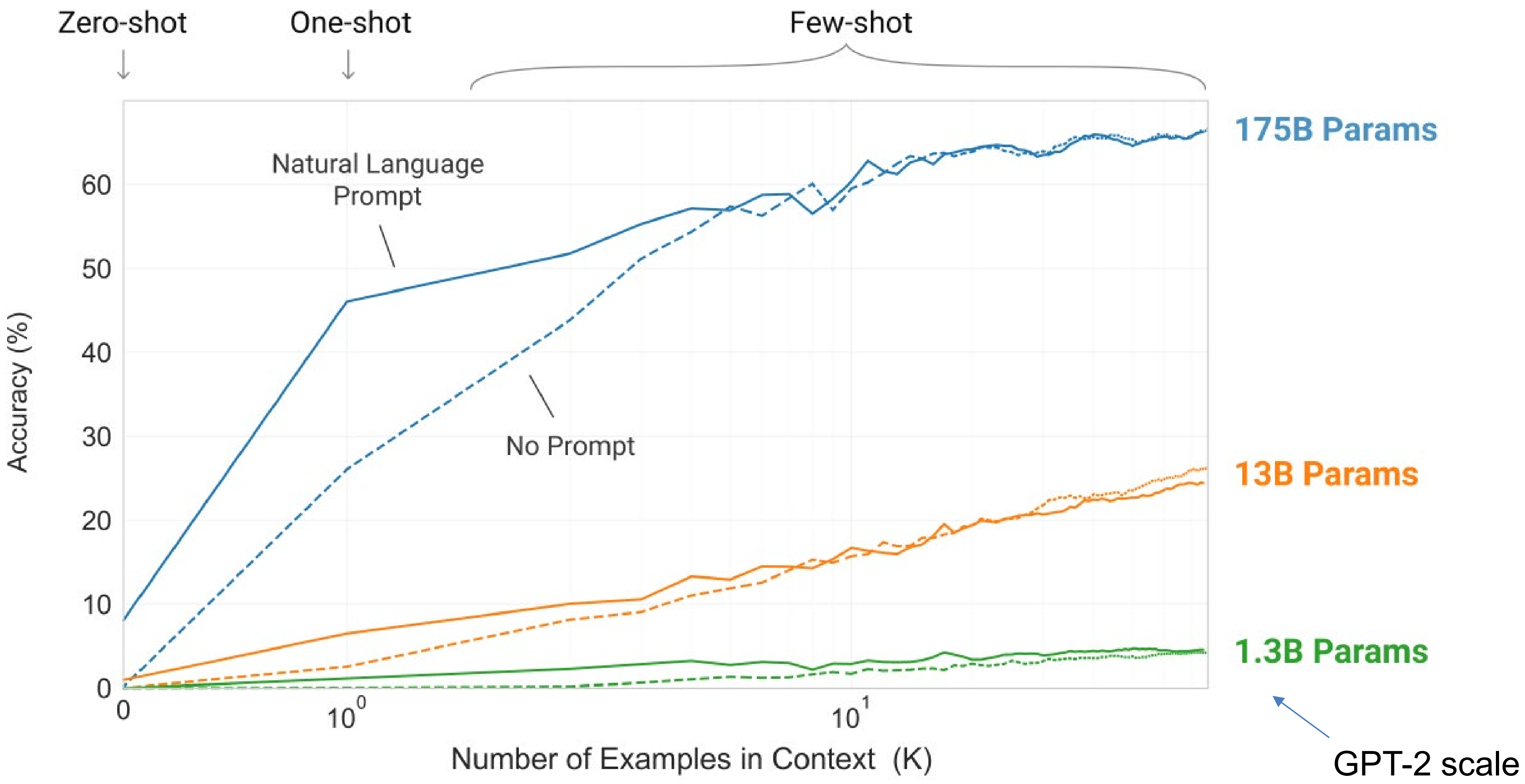


Figure 2.2: Total compute used during training. Based on the analysis in Scaling Laws For Neural Language Models [KMH+20] we train much larger models on many fewer tokens than is typical. As a consequence, although GPT-3 3B is almost 10x larger than RoBERTa-Large (355M params), both models took roughly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

Few-shot “In Context Learning”



GPT-3



Accuracy on a simple task to remove random symbols from a word

GPT-3

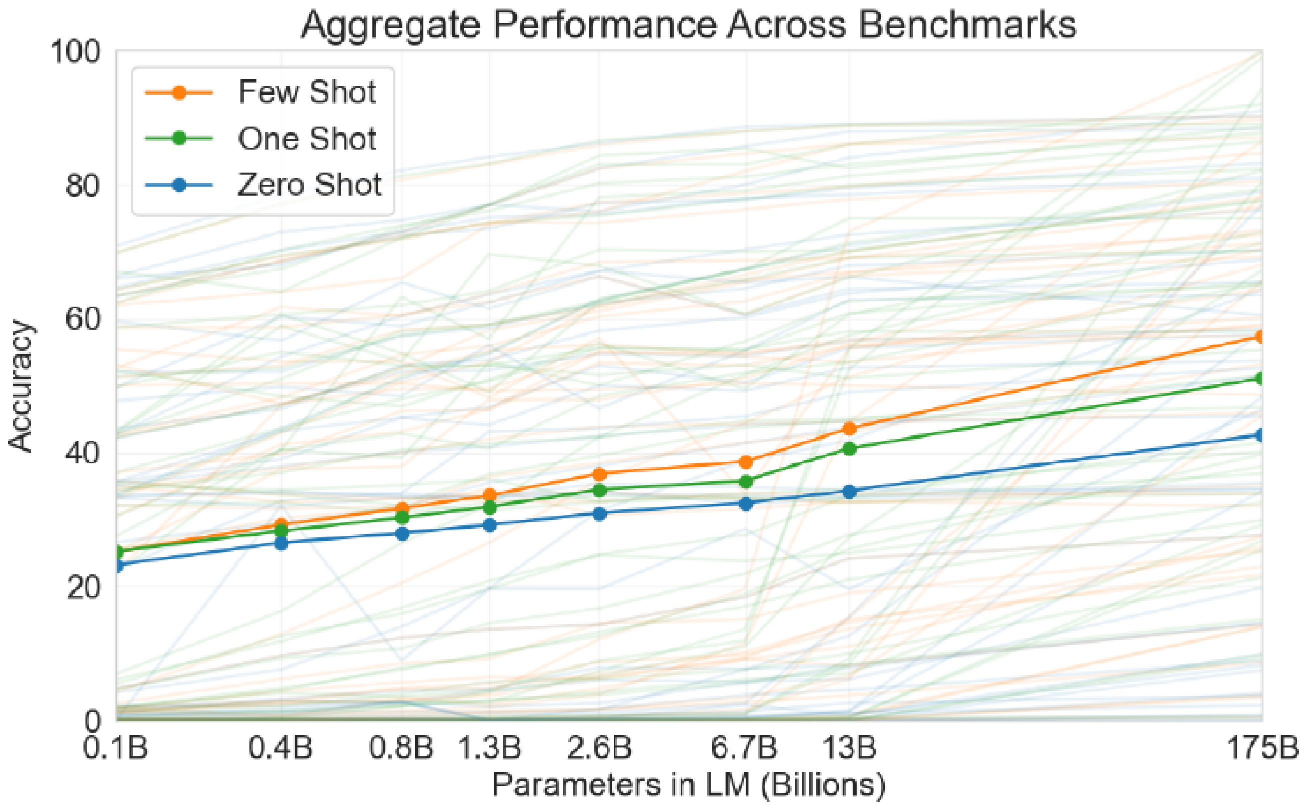


Figure 1.3: Aggregate performance for all 42 accuracy-denominated benchmarks While zero-shot performance improves steadily with model size, few-shot performance increases more rapidly, demonstrating that larger models are more proficient at in-context learning. See Figure 3.8 for a more detailed analysis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

Performance of GPT-3

	SuperGLUE Average	BoolQ Accuracy	CB Accuracy	CB F1	COPA Accuracy	RTE Accuracy
Fine-tuned SOTA	**89.0**	**91.0**	**96.9**	**93.9**	**94.8**	**92.5**
Fine-tuned BERT-Large	69.0	77.4	83.6	75.7	70.6	71.7
GPT-3 Few-Shot	71.8	76.4	75.6	52.0	92.0	69.0

	WiC Accuracy	WSC Accuracy	MultiRC Accuracy	MultiRC F1a	ReCoRD Accuracy	ReCoRD F1
Fine-tuned SOTA	**76.1**	**93.8**	**62.3**	**88.2**	**92.5**	**93.3**
Fine-tuned BERT-Large	69.6	64.6	24.1	70.0	71.3	72.0
GPT-3 Few-Shot	49.4	80.1	30.5	75.4	90.2	91.1

- Average performance of few-shot is about the same as fine-tuned BERT-Large, but varies by task
- Per-task specialized SOTA models are still best

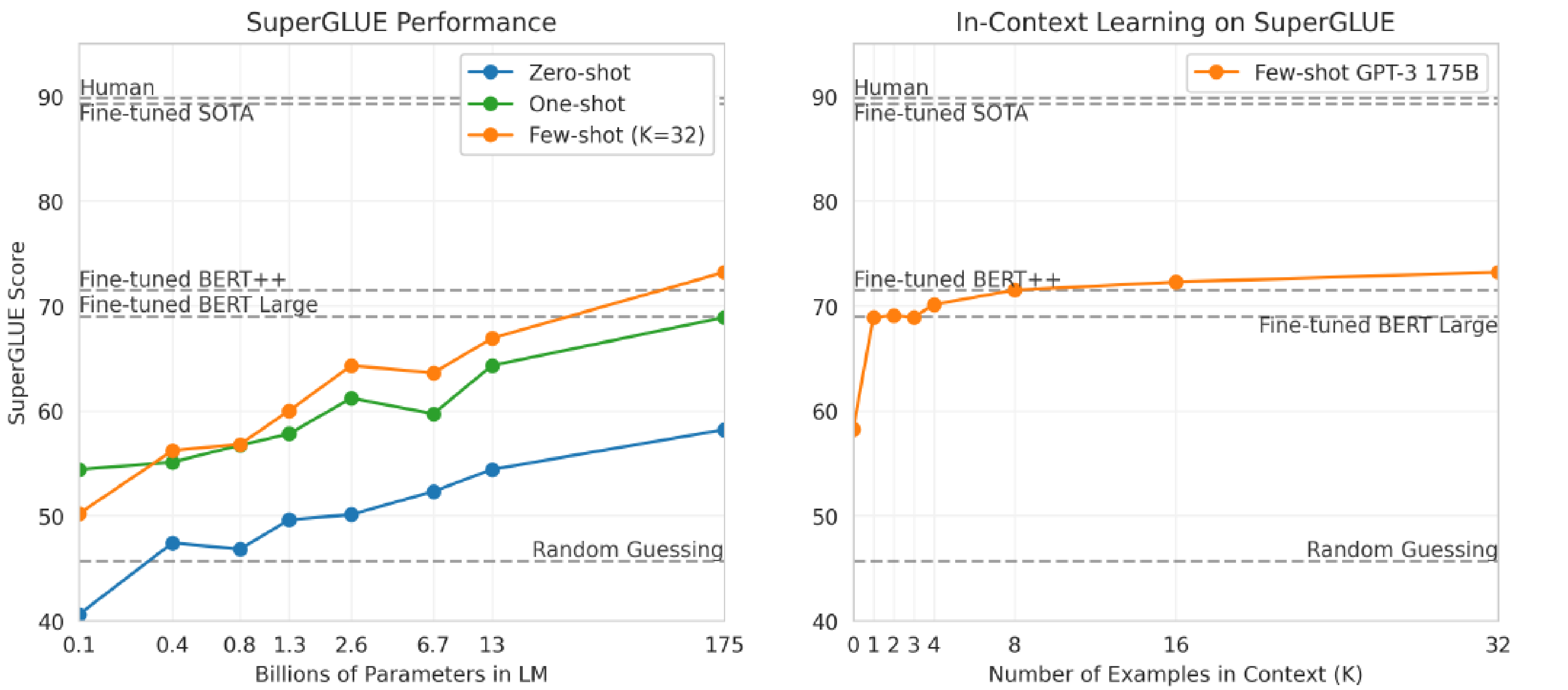
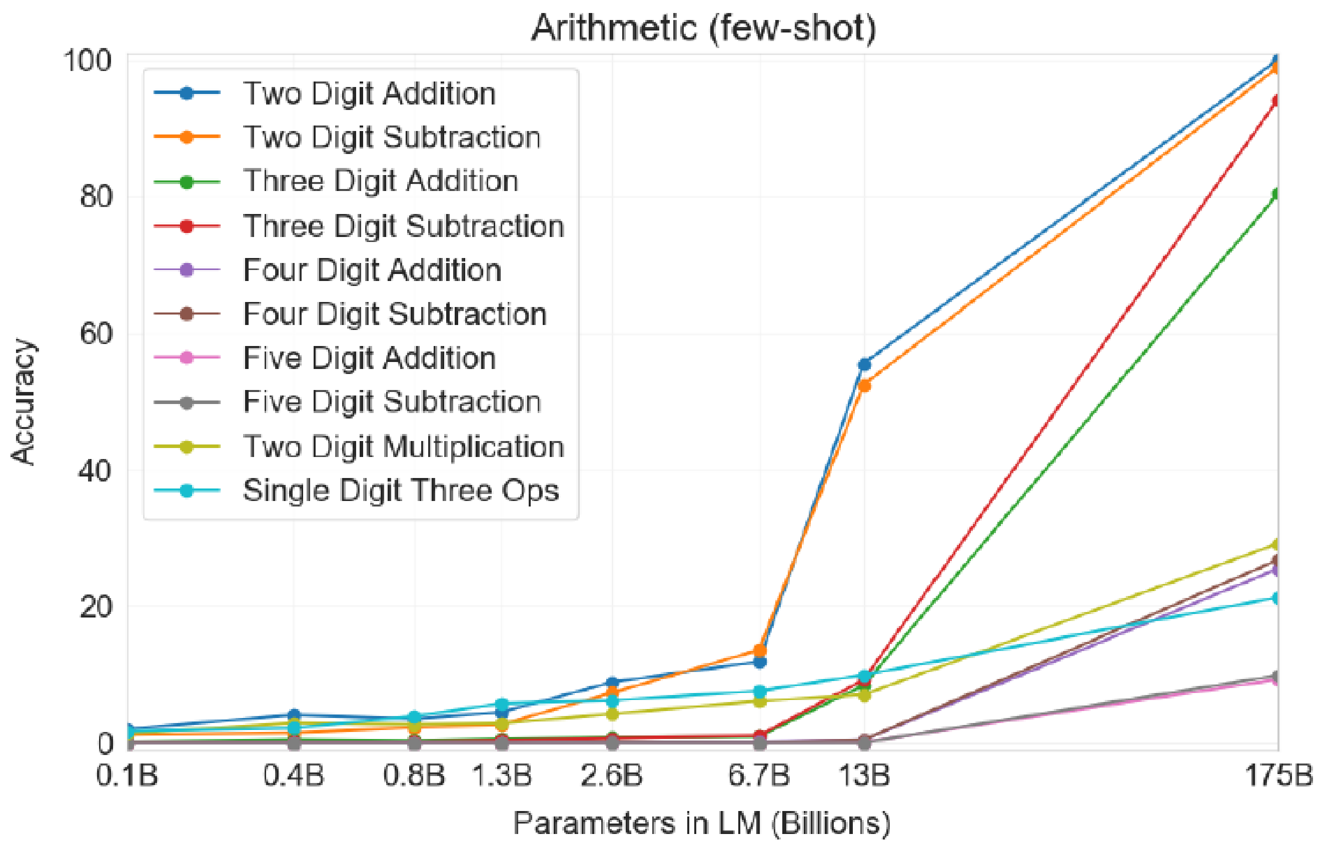


Figure 3.8: Performance on SuperGLUE increases with model size and number of examples in context. A value of $K = 32$ means that our model was shown 32 examples per task, for 256 examples total divided across the 8 tasks in SuperGLUE. We report GPT-3 values on the dev set, so our numbers are not directly 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 set (125K examples), whereas BERT++ was first fine-tuned on MultiNLI (392K examples) and SWAG (113K examples) before further fine-tuning on the SuperGLUE training set (for a total of 630K fine-tuning examples). We find the difference in performance between the BERT-Large and BERT++ to be roughly equivalent to the difference between GPT-3 with one example per context versus eight examples per context.

GPT-3 Arithmetic

- **2 digit addition (2D+)** – The model is asked to add two integers sampled uniformly from $[0, 100)$, phrased in the form of a question, e.g. "Q: What is 48 plus 76? A: 124."
- **2 digit subtraction (2D-)** – The model is asked to subtract two integers sampled uniformly from $[0, 100)$; the answer may be negative. Example: "Q: What is 34 minus 53? A: -19".
- **3 digit addition (3D+)** – Same as 2 digit addition, except numbers are uniformly sampled from $[0, 1000)$.



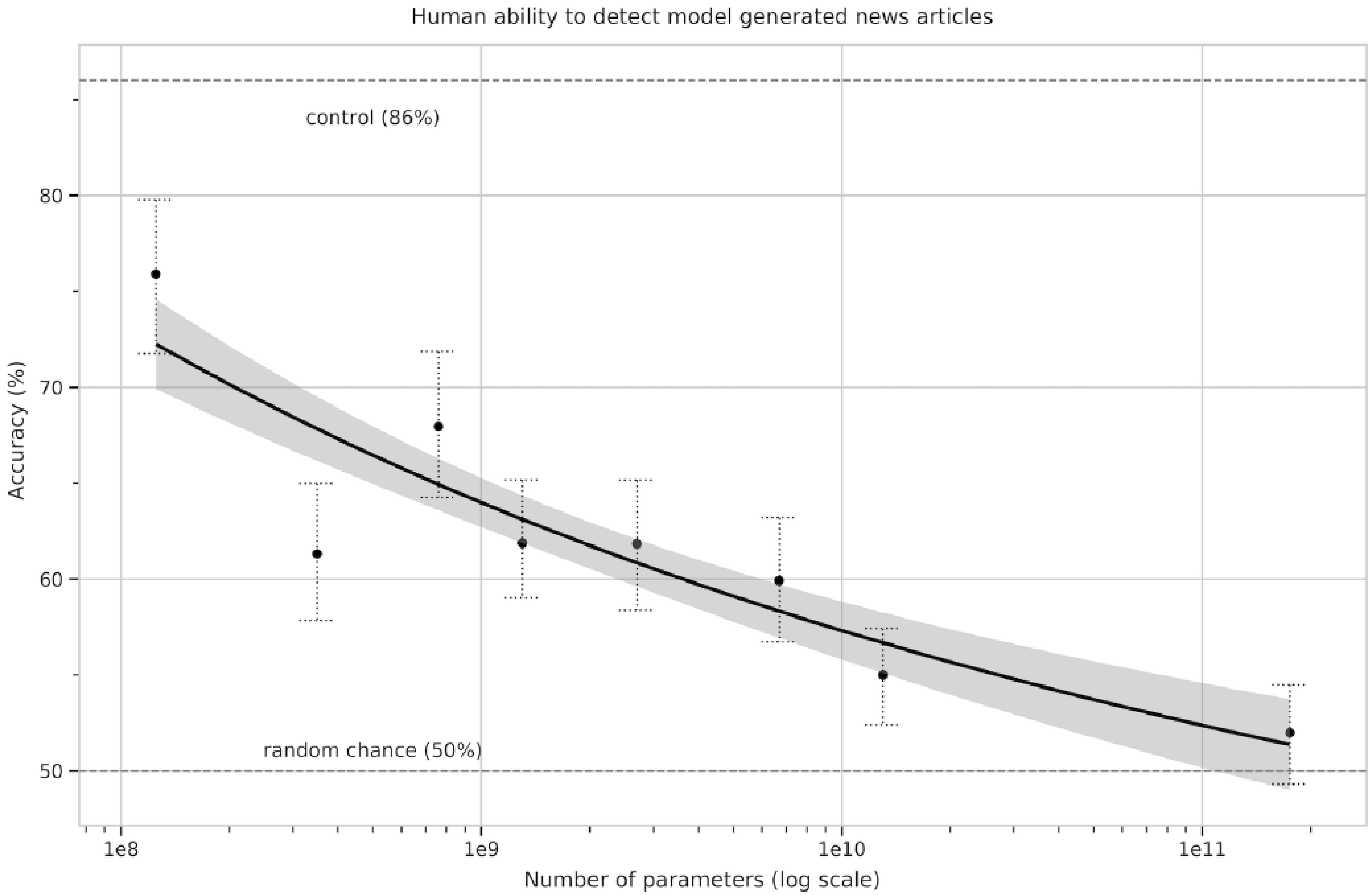


Figure 3.13: People's ability to identify whether news articles are model-generated (measured by the ratio of correct assignments to non-neutral assignments) decreases as model size increases. Accuracy on the outputs on the deliberately-bad control model (an unconditioned GPT-3 Small model with higher 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.

What to learn from the GPT Series

- GPT: generative-pretraining (GPT) is effective for large language models
  - Learns to predict the next word given preceding words

- GPT-2: GPT models can perform reasonable zero-shot task performance with larger models trained on more data

- GPT-3: Even 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

Chat GPT’s opinion of CS 441 (in limerick style)

There once was a class so great
Applied Machine Learning, first-rate
The students all learned
And their skills were discerned
Now their models can predict with high rate!
– Chat GPT

Q1-2

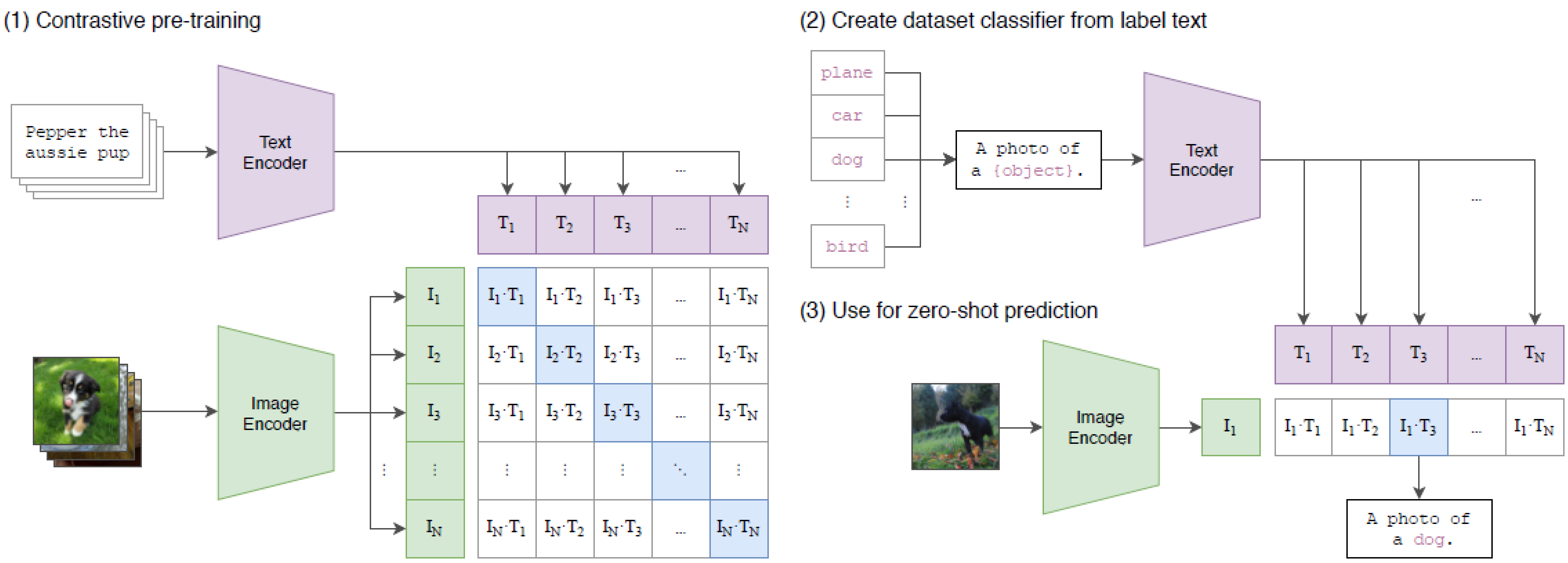
https://tinyurl.com/441-L20-fa24

Since GPT-3

- **Chat GPT** further incorporates RLHF ("reinforcement learning from human feedback") and other tuning
- **VLMs** (vision language models) proliferate, including Flamingo, Florence, LLaVA, BLIP, and Unified IO
- **GPT-4v** has rumored 1.3T params with training cost of $100M and incorporates image models
- **Co-pilot** and other coding assistants emerges as an important application of GPT
- **Other LLMs** proliferate including Mistral, Gemini, and Llama
- **Visual programming**, generating code that calls pretrained models, emerges as an alternative to multimodal/multitask single models

How much of our thoughts and conversation are just next word prediction?

CLIP: Learning Transferrable Models from Natural Language Supervision (Radford et al. 2021)



First key idea: use a text encoder as a classifier

[pdf](pdf)

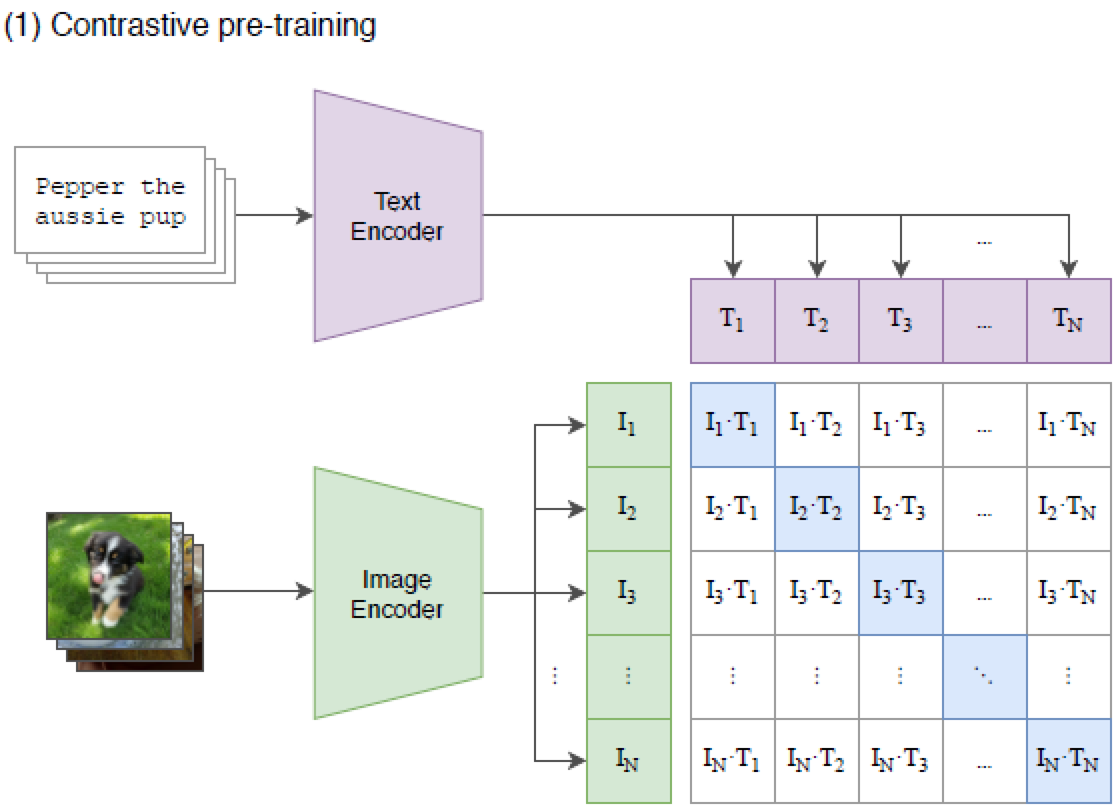
First key idea: **use a text encoder as a classifier**

- This is an old idea – words and pictures work goes back to ~2000, but at a smaller scale

- Main challenge: How to scale?
  - Learn from natural language supervision (not tags or class labels)
  - Scrape 400 million image/text pairs
  - “Bag of words” language representation
  - Contrastive objective, instead of predicting exact language
  - Use transformer architecture

Second key idea(s): **contrastively match gestalt text to image**

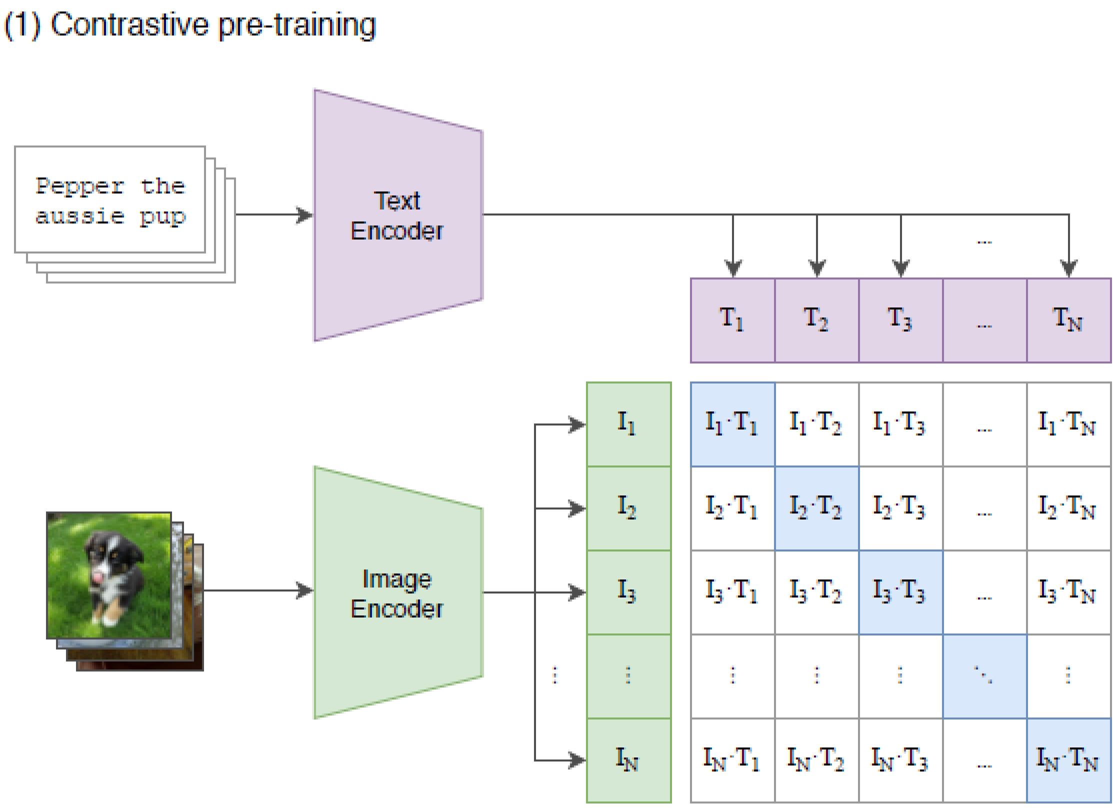
- Use small transformer language model (76M parameters for base)
- Matching task with large batch (size = 32,768)
  - Each image and text from batch is encoded
  - Similarity score obtained for 32K x 32K image-text pairings
  - Loss is cross-entropy on matching each image to its text, and each text to its image



Contrastive task formulations is a good general way to learn when exact target is unpredictable

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

Figure 3. Numpy-like pseudocode for the core of an implementation of CLIP.



Training cost

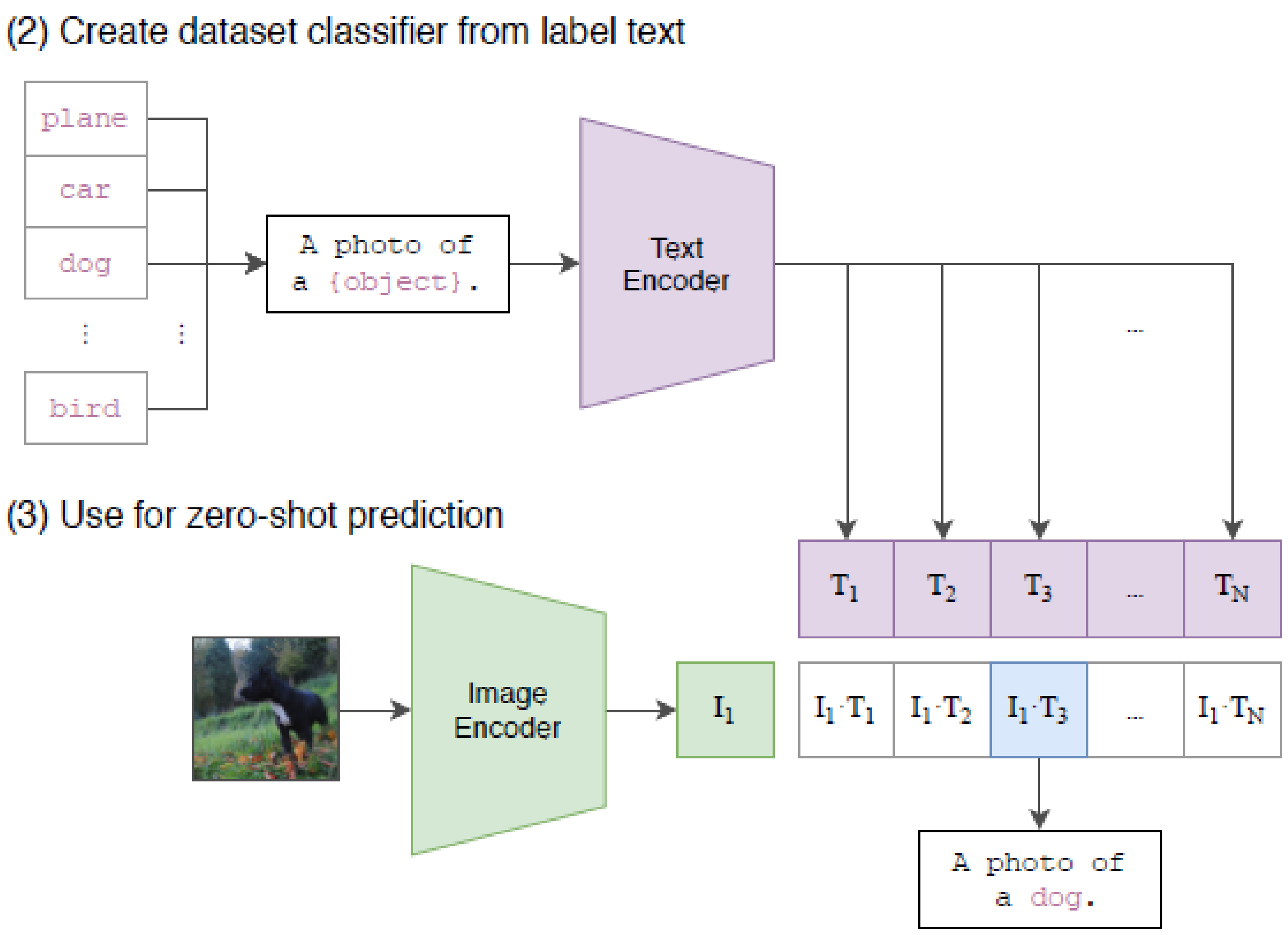
- "The largest ResNet model RN50x64, took 18 days to train on 592 V100 GPUs, while the largest Vision Transformer took 12 days on 256 V100 GPUs"
  - ~$91K for Transformer model; $300K for ResNet model

Key idea 3: **zero-shot classification**

Every batch of training is like a novel classification task, matching 32K classes to 32K images

To create a new classification task:

1. Convert class labels into captions and encode the text
2. Encode the image
3. Assign the image to the label whose caption matches best



Four ways to adapt CLIP to a new task

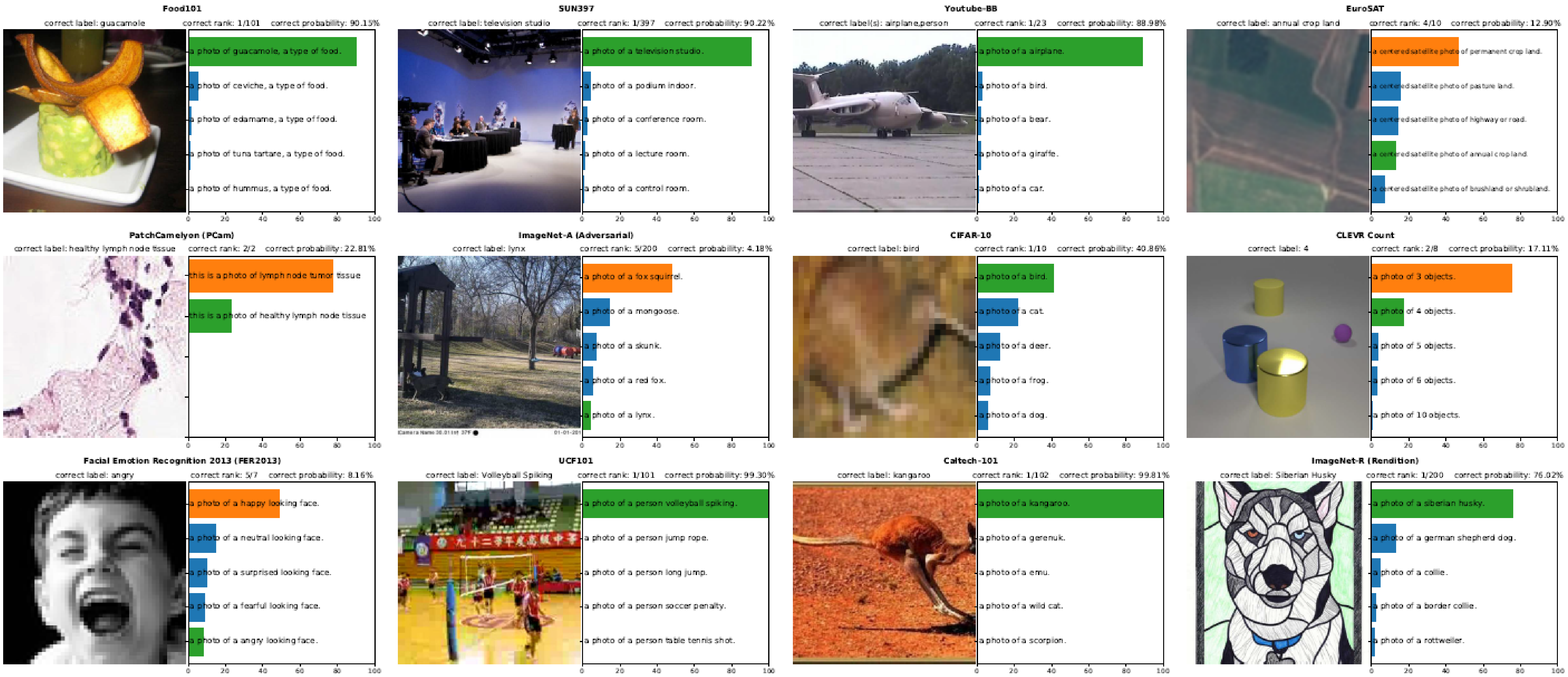
1. Zero-shot: convert labels to text and use text-image similarity

2. Linear probe: freeze the image encoder and train a linear layer on its features

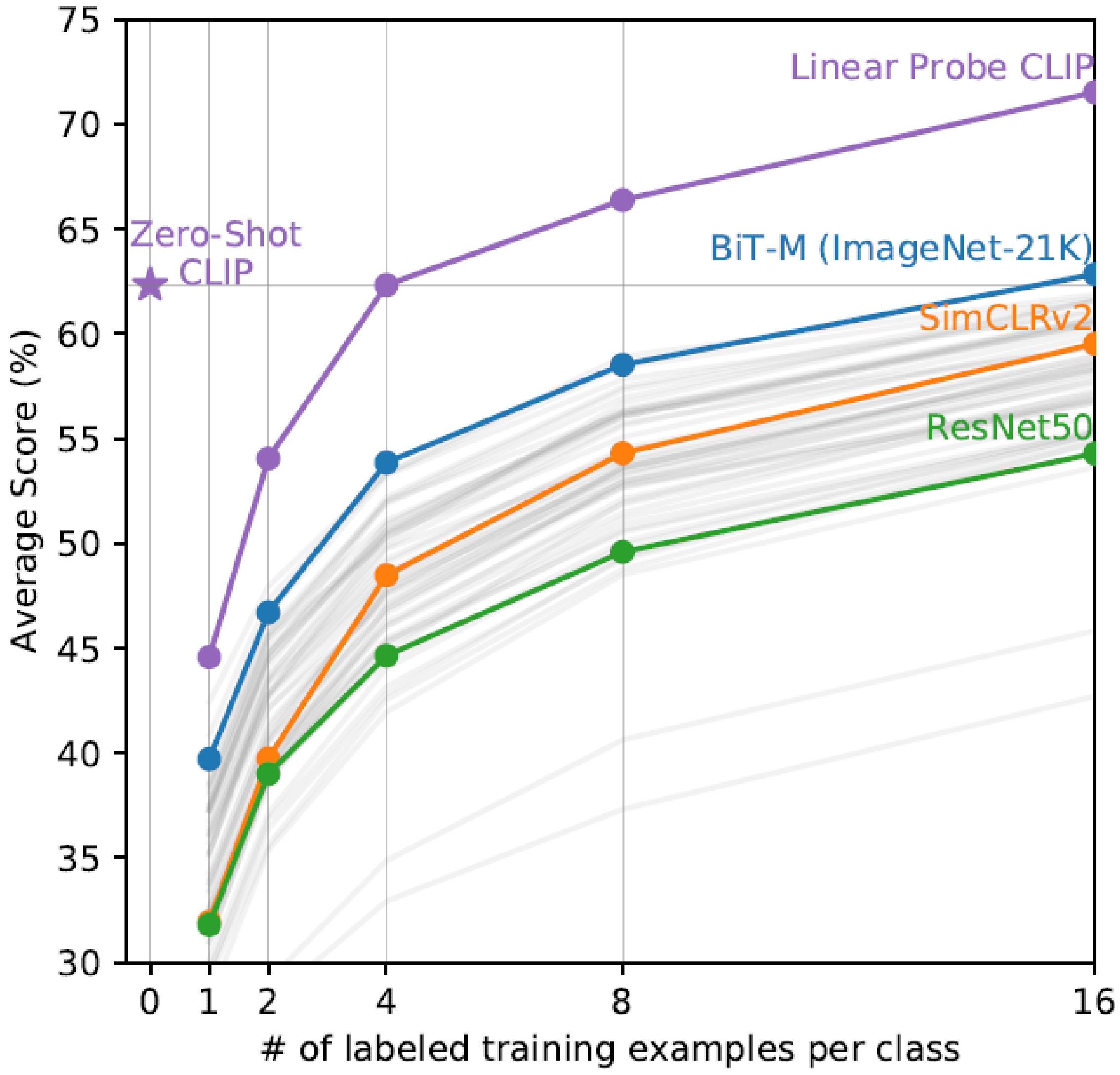
3. Nearest neighbor (not in paper): record features of training examples and use K-NN classifier

4. Fine-tune CLIP encoder for the new task (but then it completely loses its generality)

Zero shot prediction examples (randomly selected)



- Zero-shot clip performs as well as a strong baseline trained on 16 examples per class
- Linear probe needs 4 examples to reach zero-shot performance (on average)



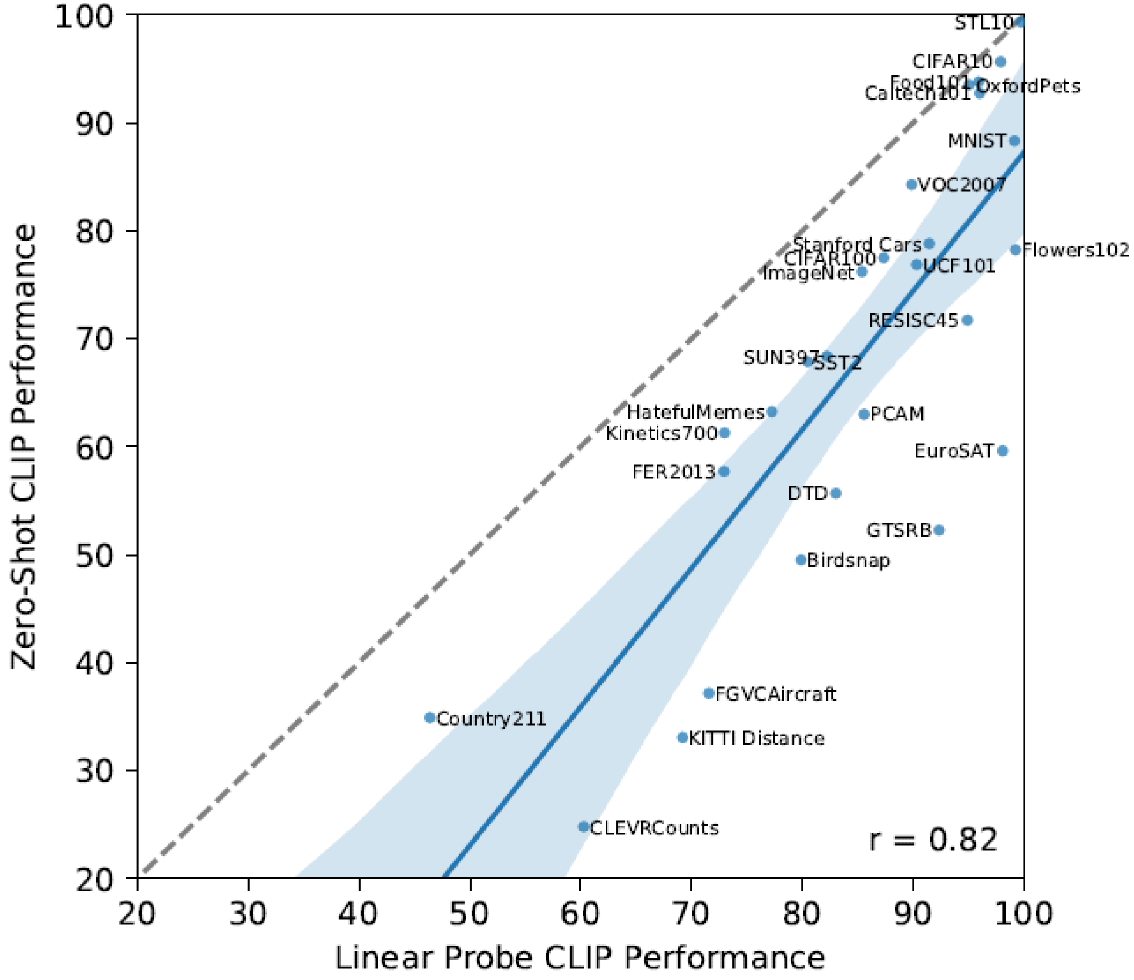


Figure 8. **Zero-shot performance is correlated with linear probe performance but still mostly sub-optimal.** Comparing zero-shot and linear probe performance across datasets shows a strong correlation with zero-shot performance mostly shifted 10 to 25 points lower. On only 5 datasets does zero-shot performance approach linear probe performance (≤3 point difference).

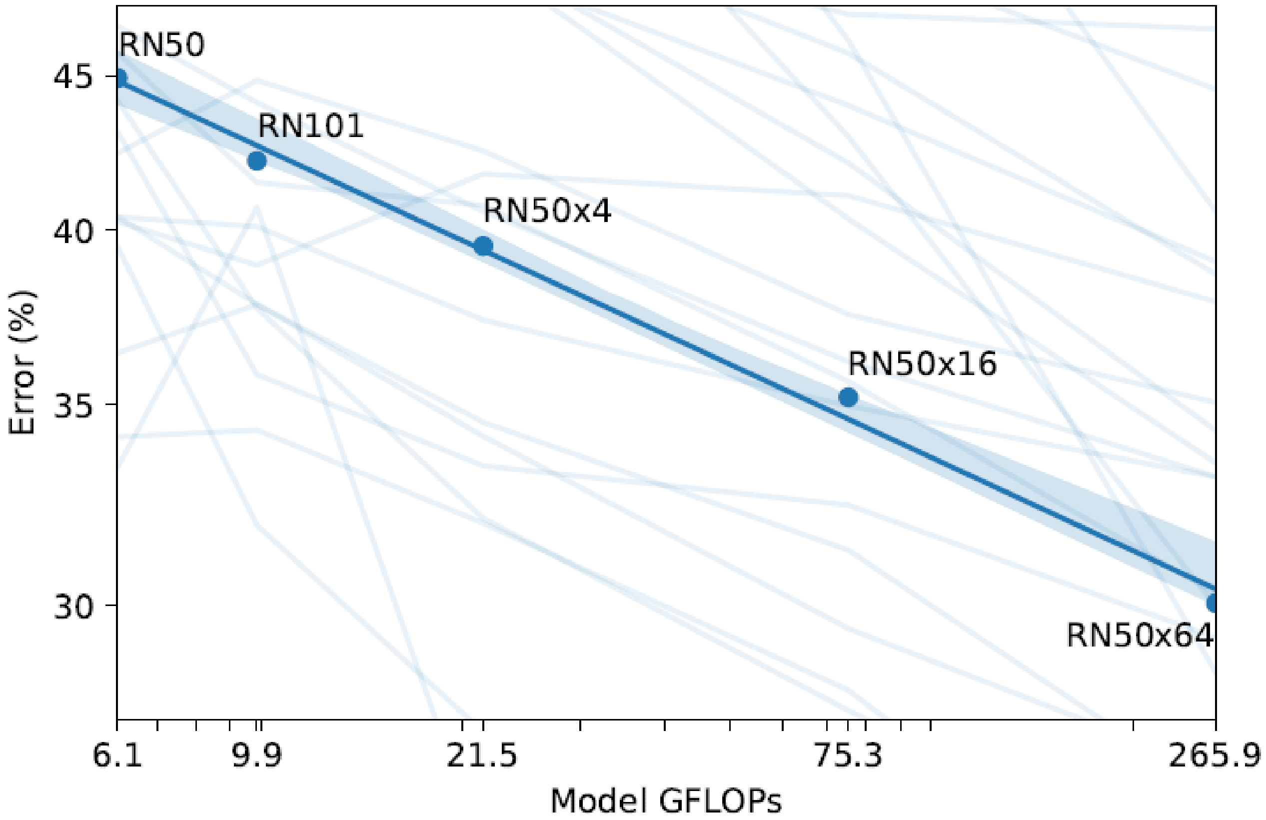


Figure 9. **Zero-shot CLIP performance scales smoothly as a function of model compute.** Across 39 evals on 36 different

The CLIP model is used as the image encoder for many vision-language models and image generators, e.g. DALL-E2, Stable Diffusion, BLIP, Molmo

Q3-4

https://tinyurl.com/441-L20-fa24

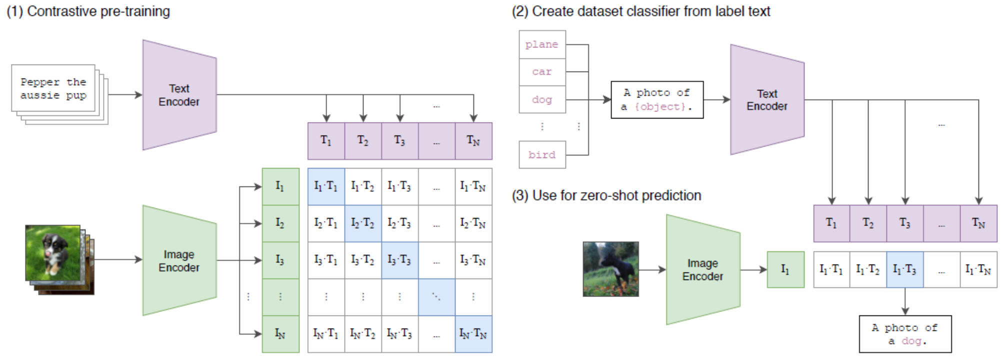
What to remember

- Deep learning models are rarely trained from scratch. Instead,
  - Train on a large supervised dataset and fine-tune on target tasks, e.g. ImageNet-based models
  - Train on a large unsupervised dataset and fine-tune on target tasks, e.g. BERT
  - Train on a large unsupervised dataset and apply to target tasks without fine-tuning, e.g. CLIP and GPT

- With large-scale training and the right formulations, models can perform a range of tasks including those not explicitly trained

- **GPT** demonstrates that learning to predict the next word produces a flexible zero-shot and few-shot general language task performer

- **CLIP** shows that learning to match images to text produces a good zero-shot classifier and an excellent image encoder



Coming up

- Applications
  - Ethics and Impact of AI
  - Bias and Fairness
  - Real-world Applications in ML, with guest speaker Chenxi Yu
  - Audio and 1D signals