

Pretraining scaling

CS 4804: Introduction to AI

Fall 2025

<https://tuvllms.github.io/ai-fall-2025/>

Tu Vu

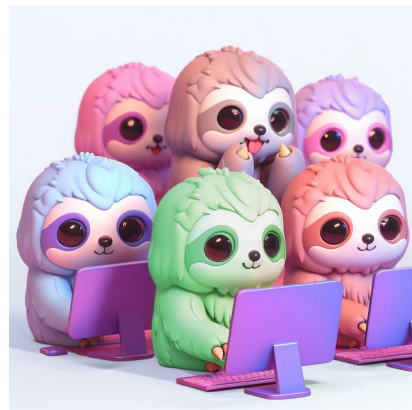


Logistics

- Homework 1 & Quiz 1 are coming
- Final Project proposal (**due 10/7**)
 - Project research ideas
 - Check out [GPT-5 System Card](#) & [Gemini 2.5 Pro Technical Report](#)
 - New benchmarks
 - New prompting methods
 - Improving small models
 - ...

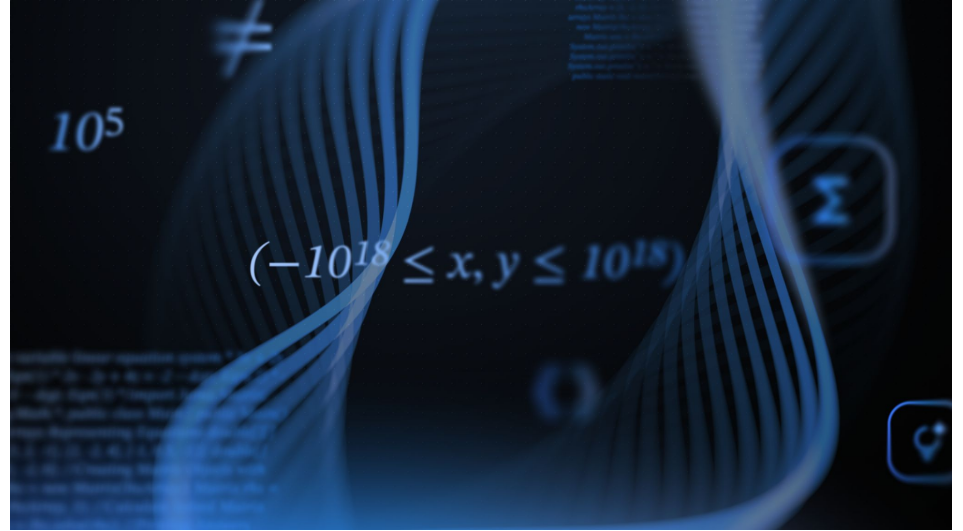
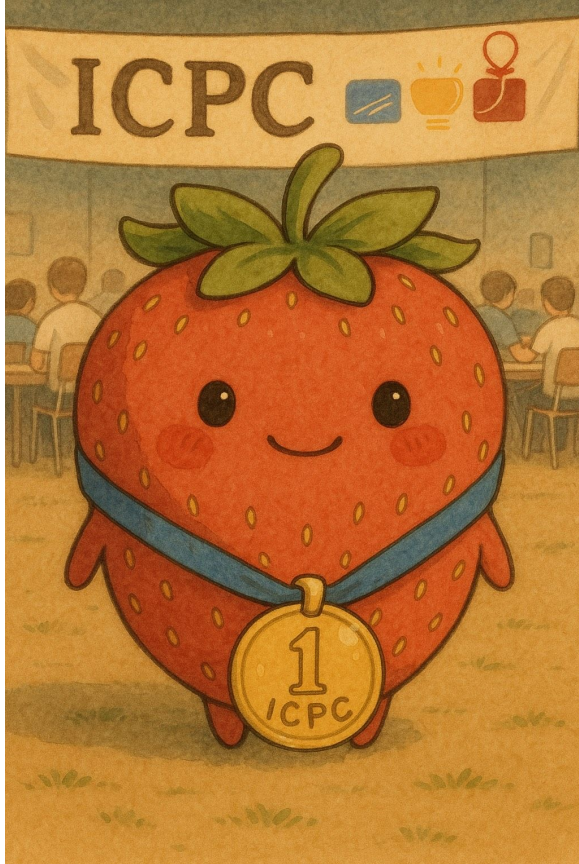
Useful libraries

- Hugging Face (<https://huggingface.co>)
 - Transformers
 - Models / Datasets
- Unsloth (<https://unsloth.ai>)
 - Efficient fine-tuning
- vLLM (<https://github.com/vllm-project/vllm>)
 - Efficient inference
- [Together.AI](https://together.ai)
 - APIs for Open-weight models



together.ai

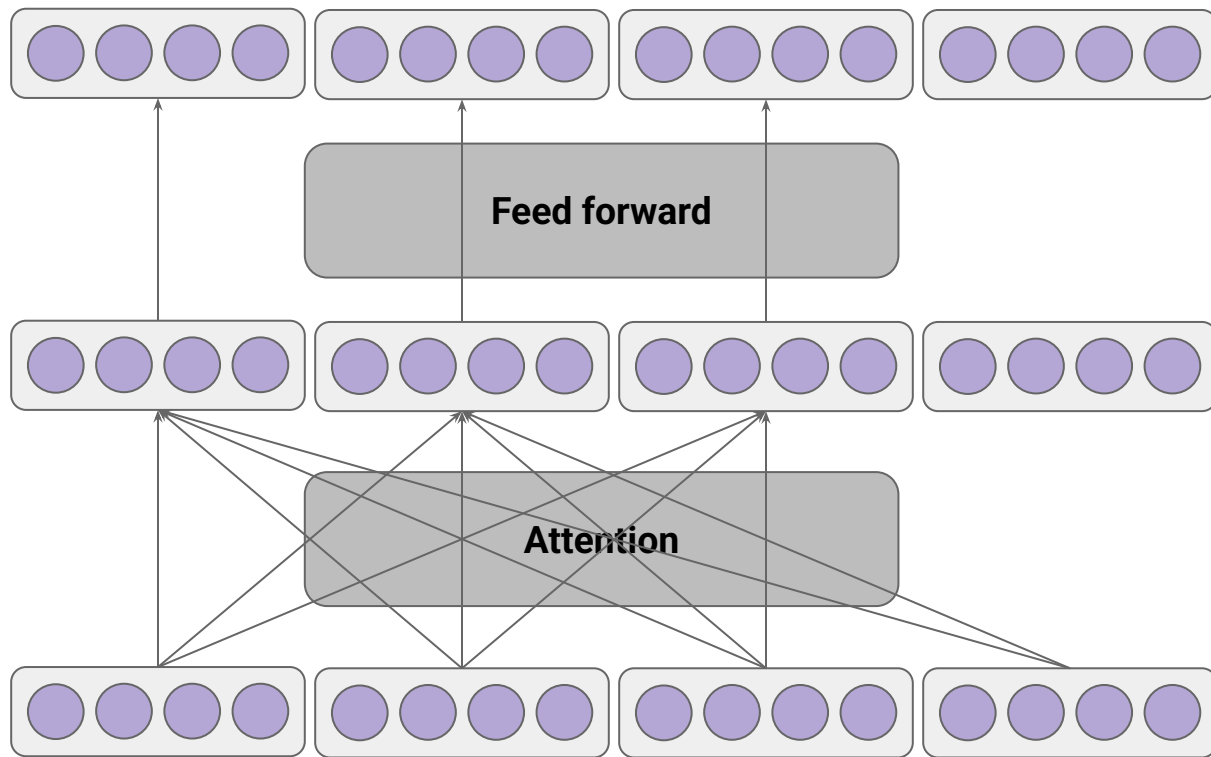
Gold-medal level ICPC World Finals



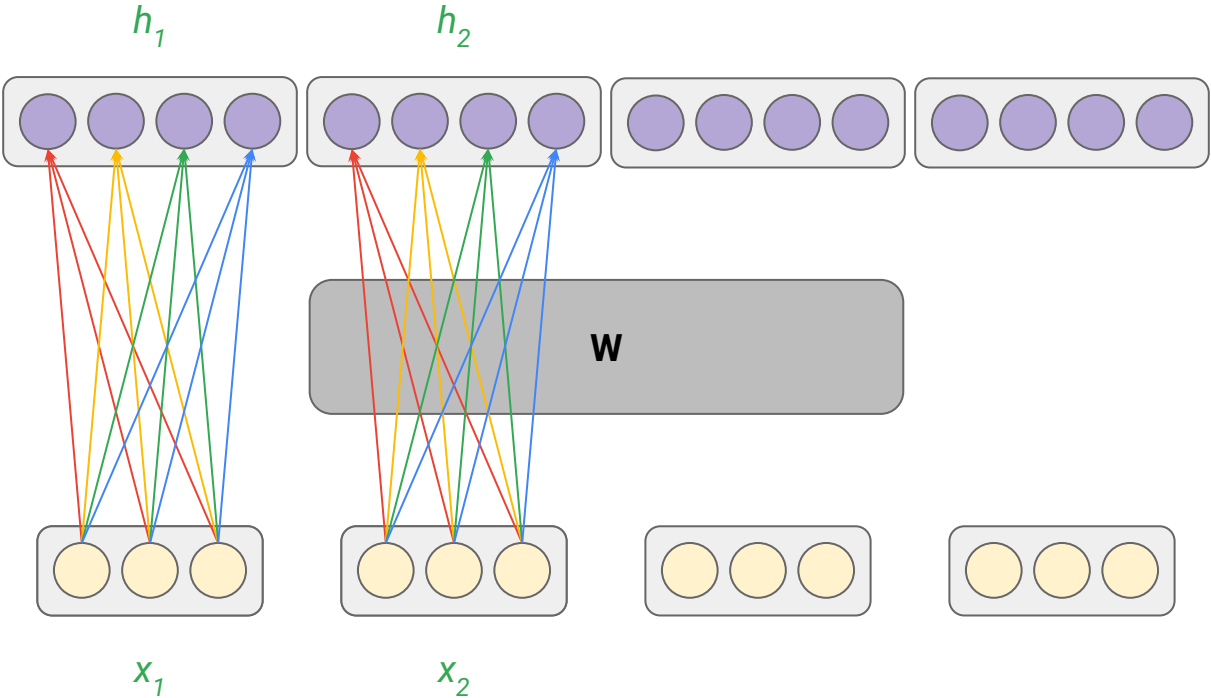
What's next? Riemann Hypothesis?

- One of the Seven Millennium Prize Problems
- Clay Mathematics Institute offers US\$1 million for a solution to any of them

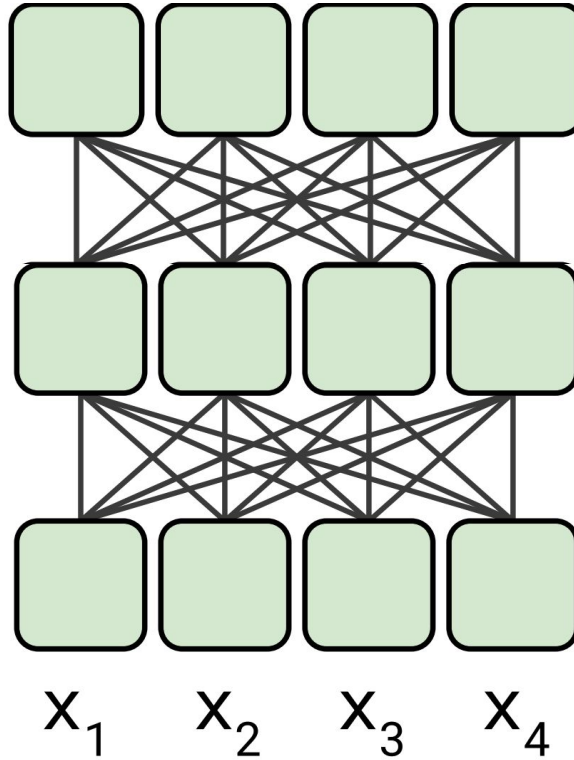
Encoder



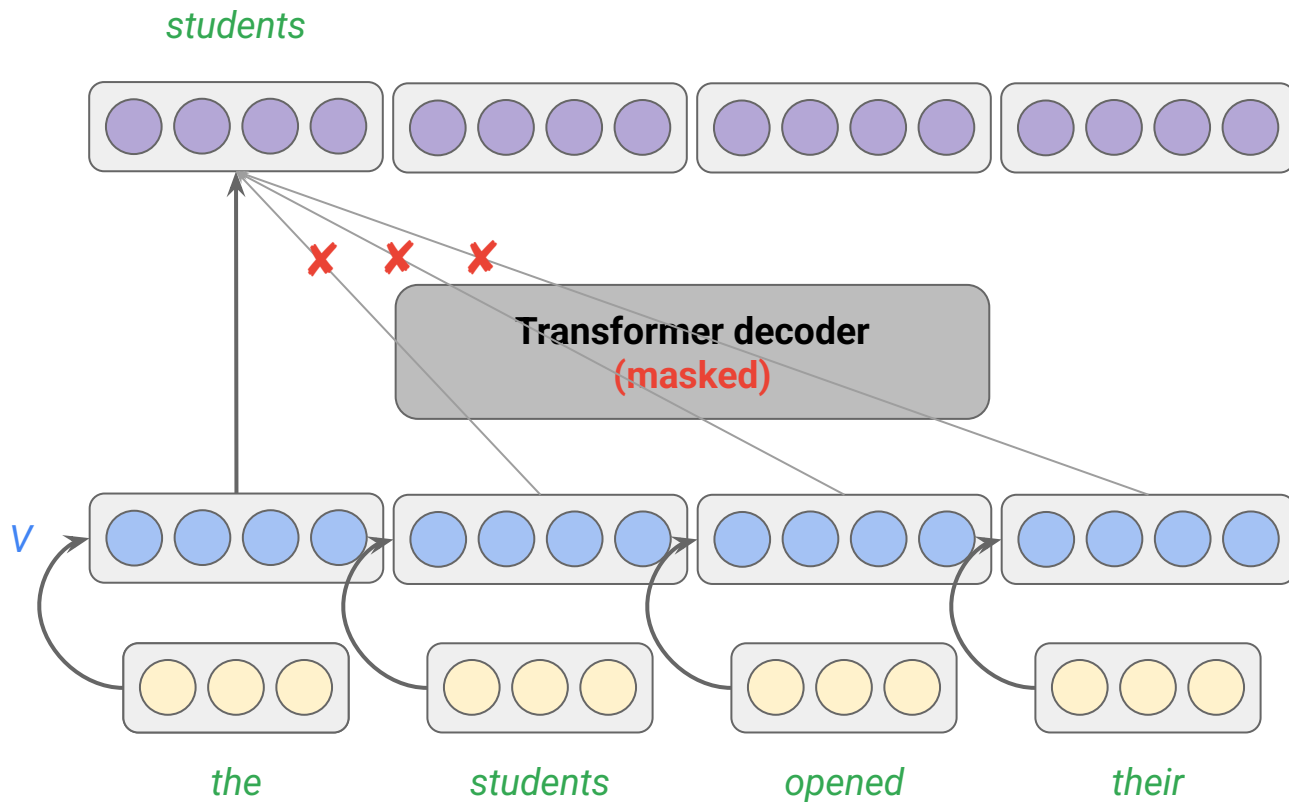
Position-wise Feed-Forward Networks



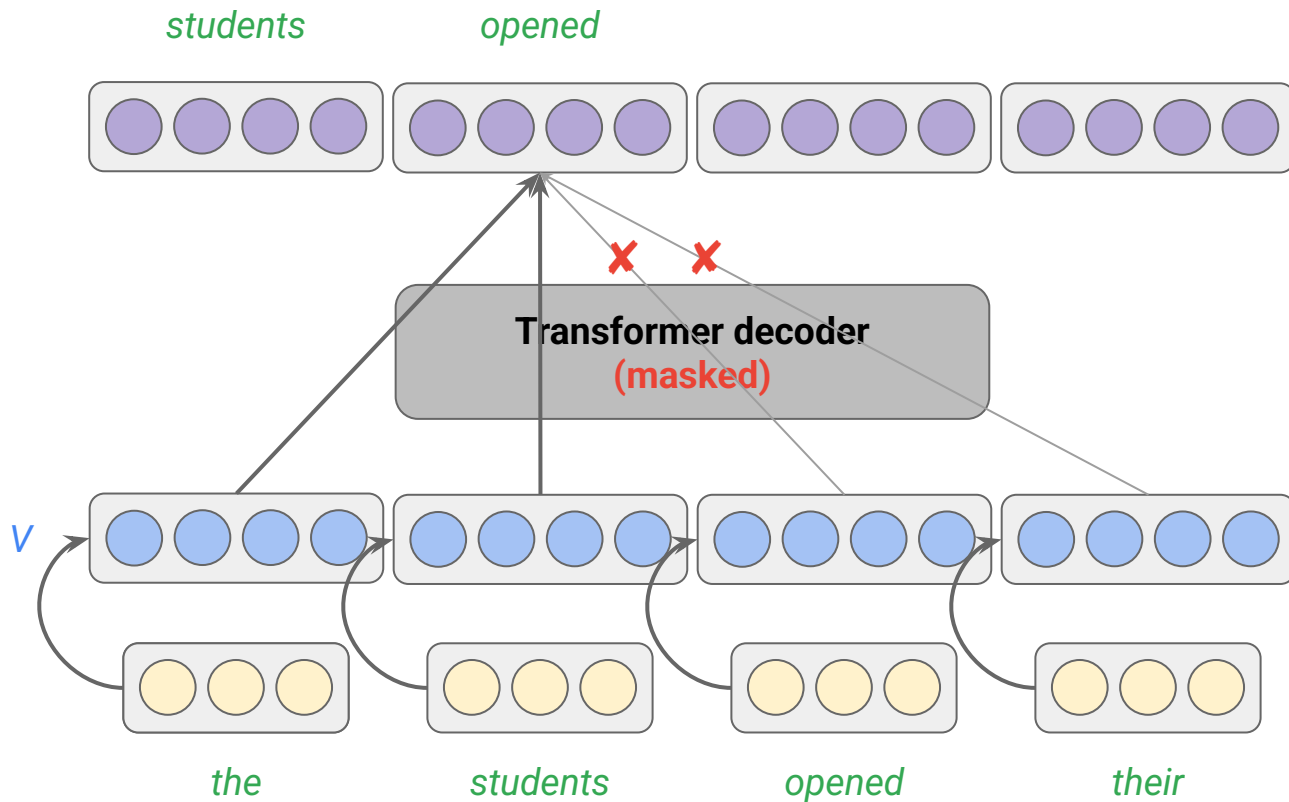
Encoder (cont'd)



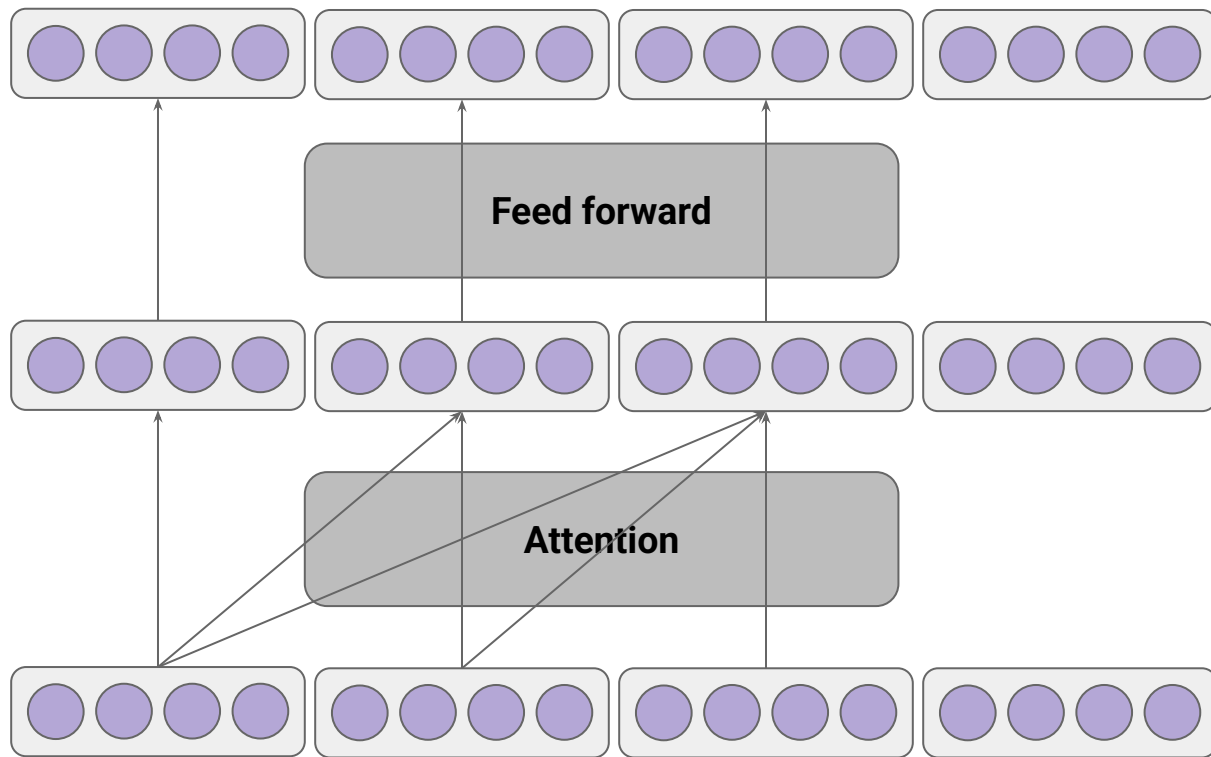
Decoder



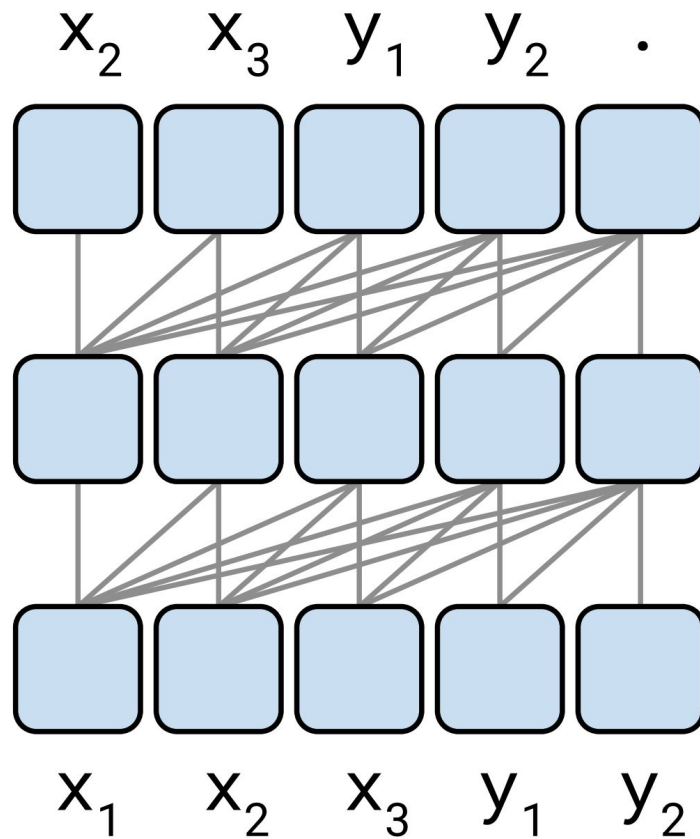
Decoder (cont'd)



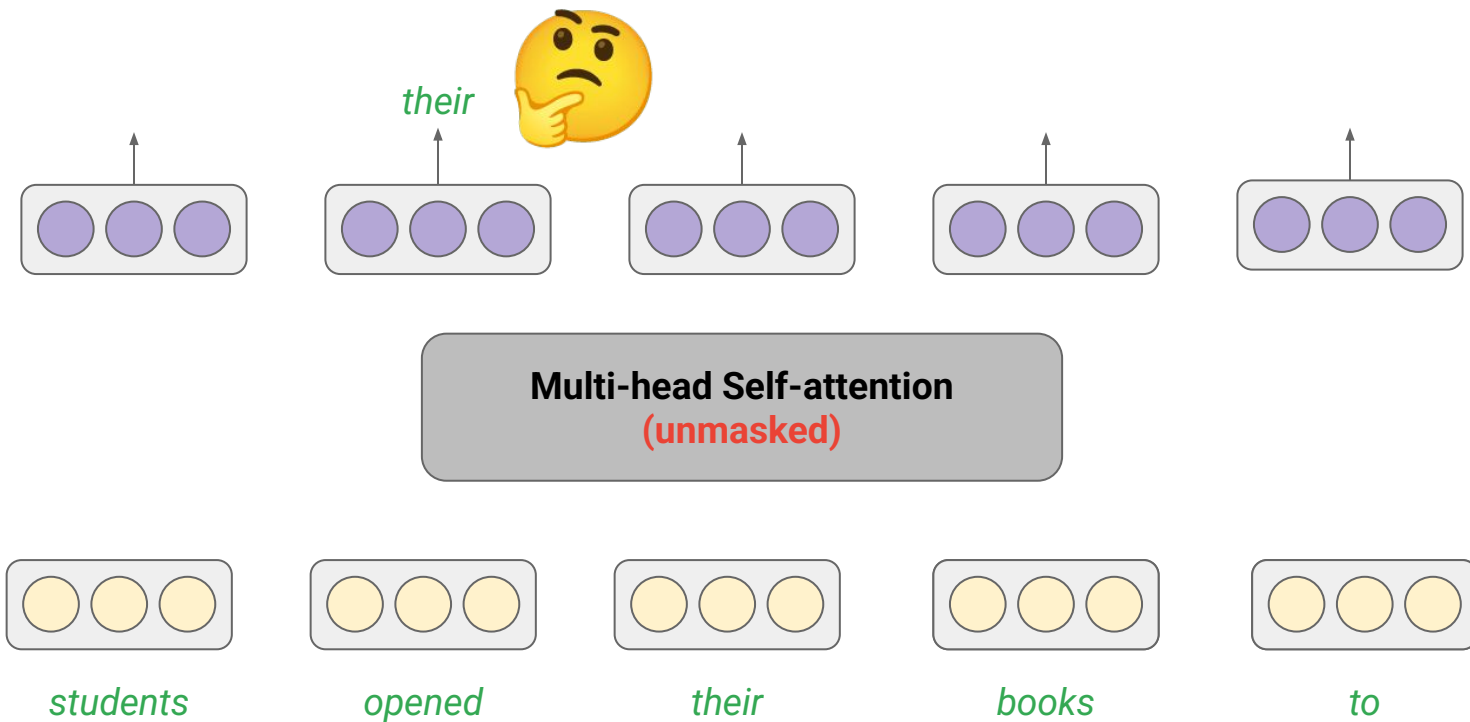
Decoder (cont'd)



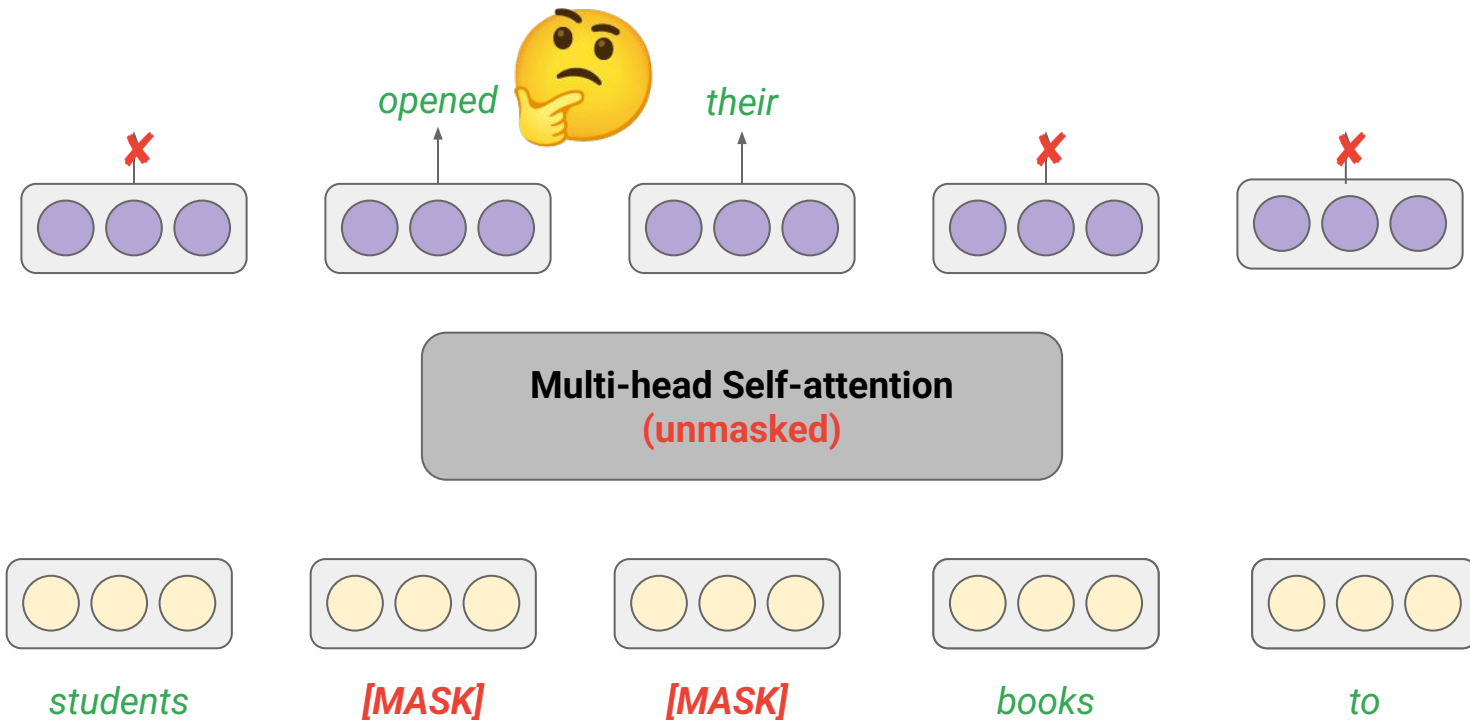
Decoder (cont'd)



Language modeling using a Transformer encoder

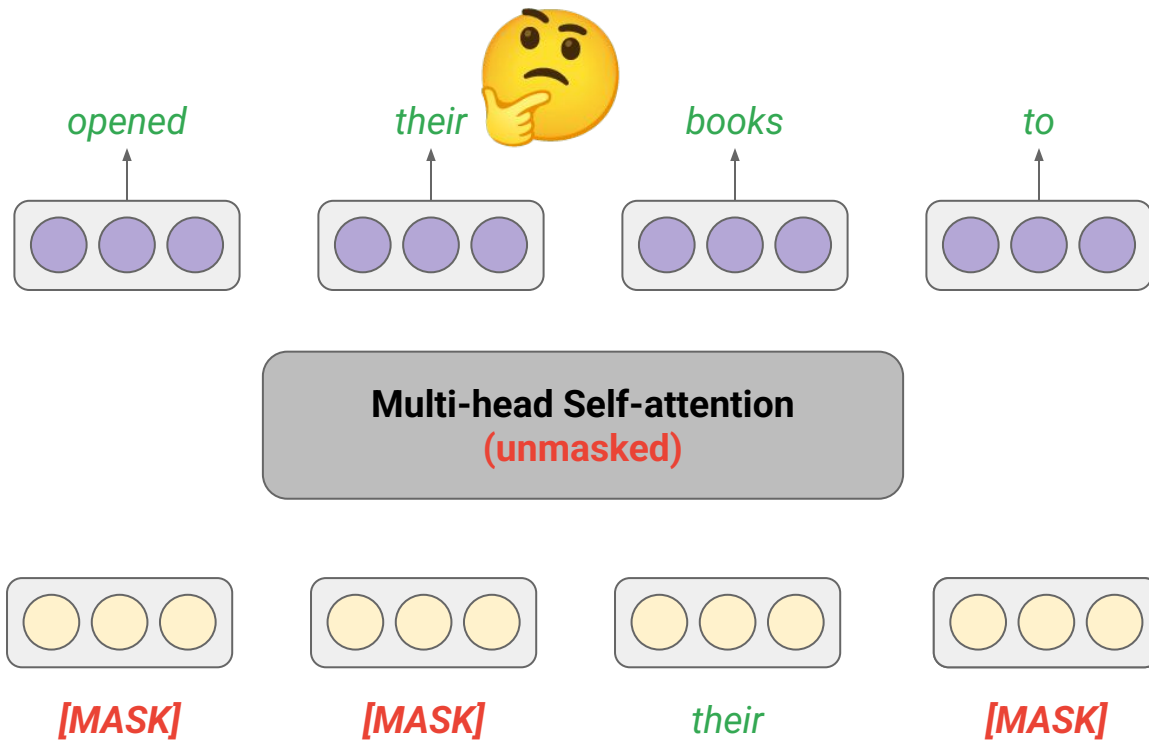


Masked language modeling



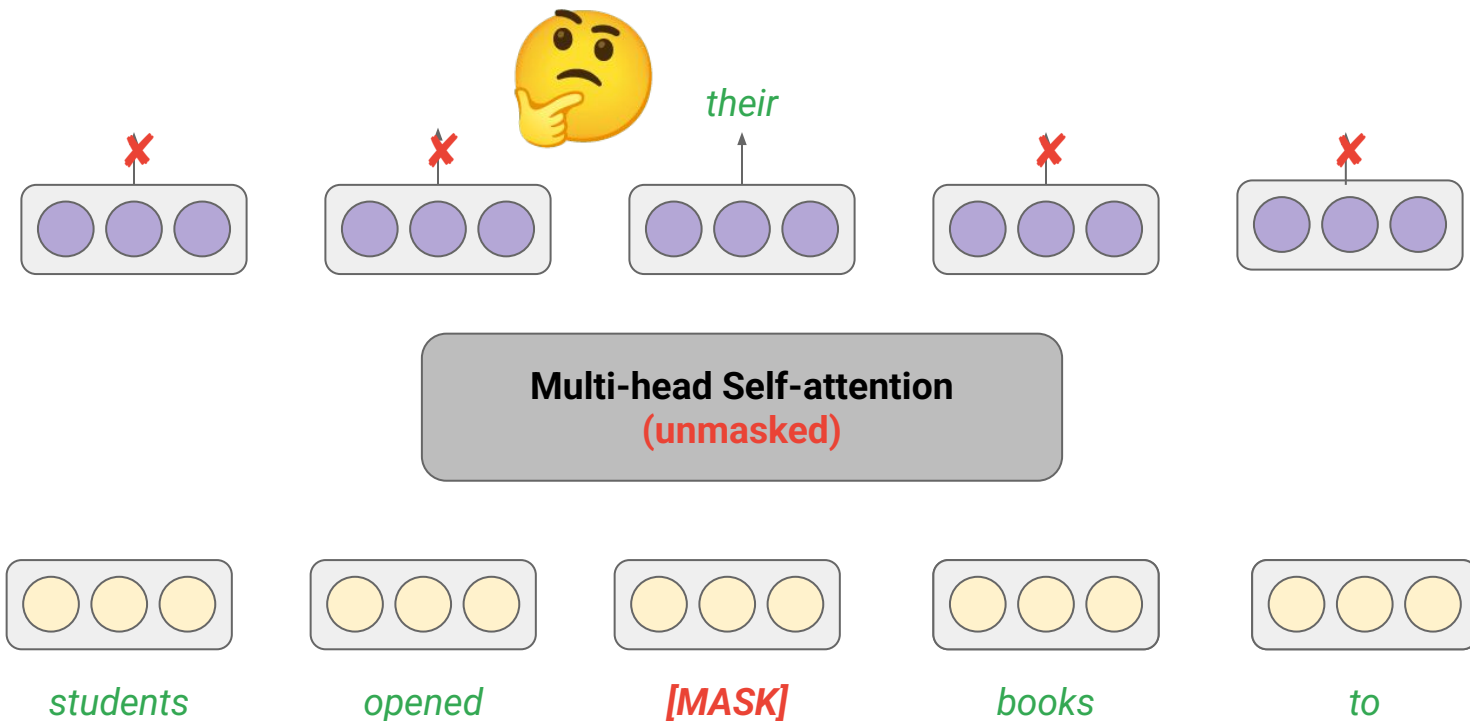
15% - 30% of all tokens in each sequence are masked at random

What if we mask more tokens?



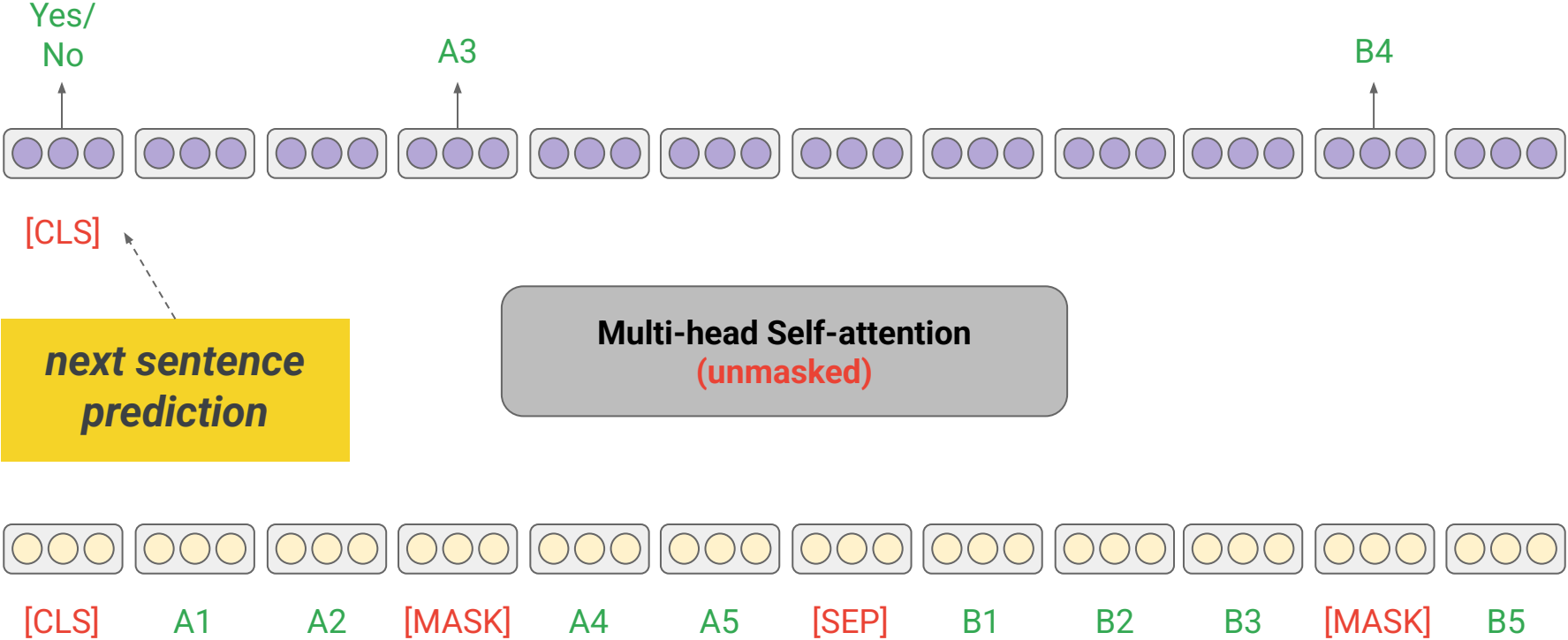
15% - 30% of all tokens in each sequence are masked at random

What if we mask less tokens?

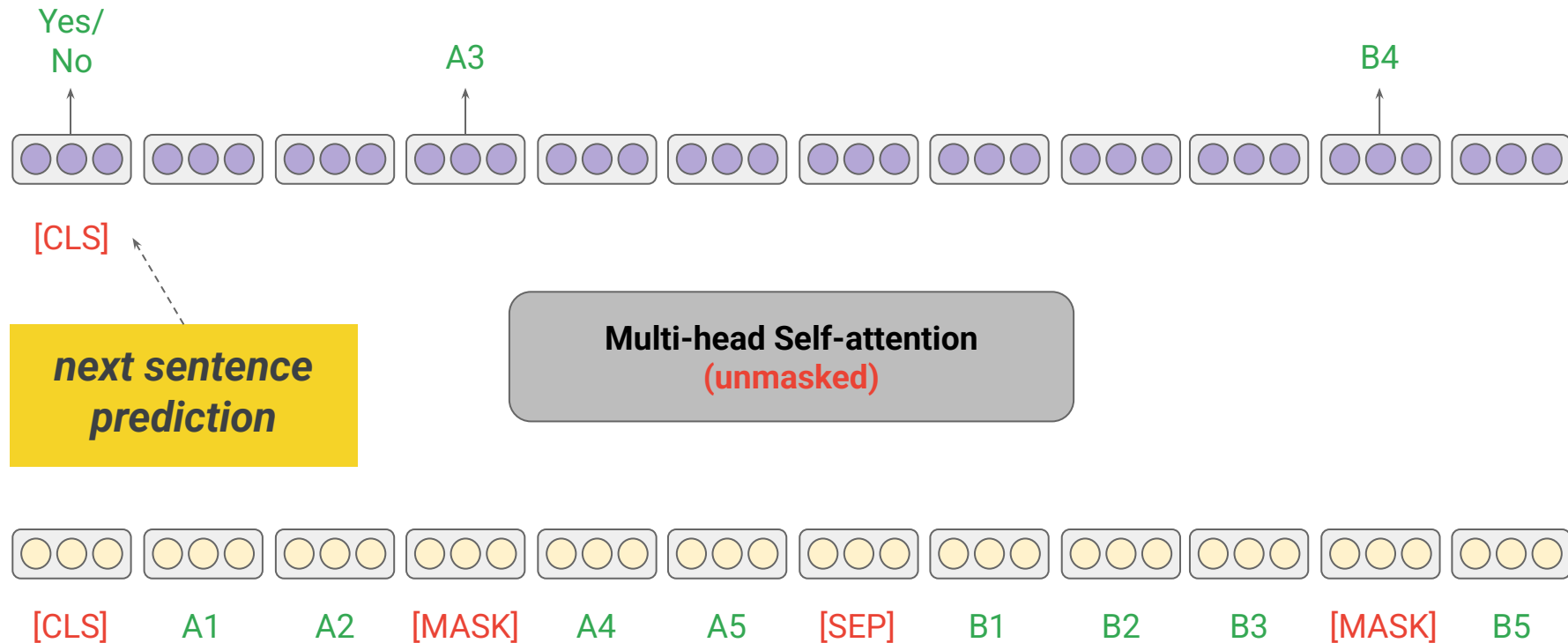


15% - 30% of all tokens in each sequence are masked at random

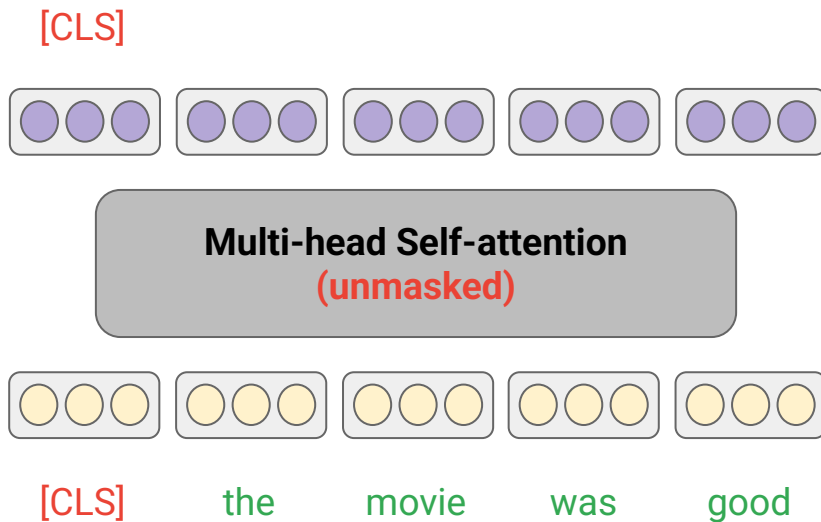
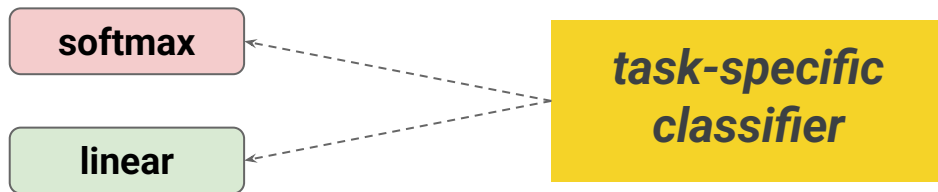
CLS & SEP tokens



BERT Pretraining



BERT Fine-tuning



T5: Text-to-Text Transfer Transformer

Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Colin Raffel*

CRAFFEL@GMAIL.COM

Noam Shazeer*

NOAM@GOOGLE.COM

Adam Roberts*

ADAROB@GOOGLE.COM

Katherine Lee*

KATHERINELEE@GOOGLE.COM

Sharan Narang

SHARANNARANG@GOOGLE.COM

Michael Matena

MMATENA@GOOGLE.COM

Yanqi Zhou

YANQIZ@GOOGLE.COM

Wei Li

MWEILI@GOOGLE.COM

Peter J. Liu

PETERJLIU@GOOGLE.COM

Google, Mountain View, CA 94043, USA

T5 Pretraining: Span corruption

<X>, <Y>: sentinel tokens



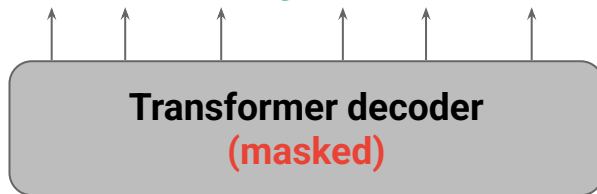
Transformer encoder
(unmasked)

Thank you <X> me to your party <Y> week

Thank you for inviting me to your party last week

encoder

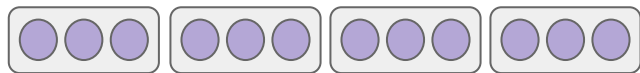
<X> for inviting <Y> last <EOS>



<BOS> <X> for inviting <Y> last

decoder

T5 Fine-tuning



Transformer encoder
(unmasked)

sentiment analysis: this movie was good

encoder

positive <EOS>

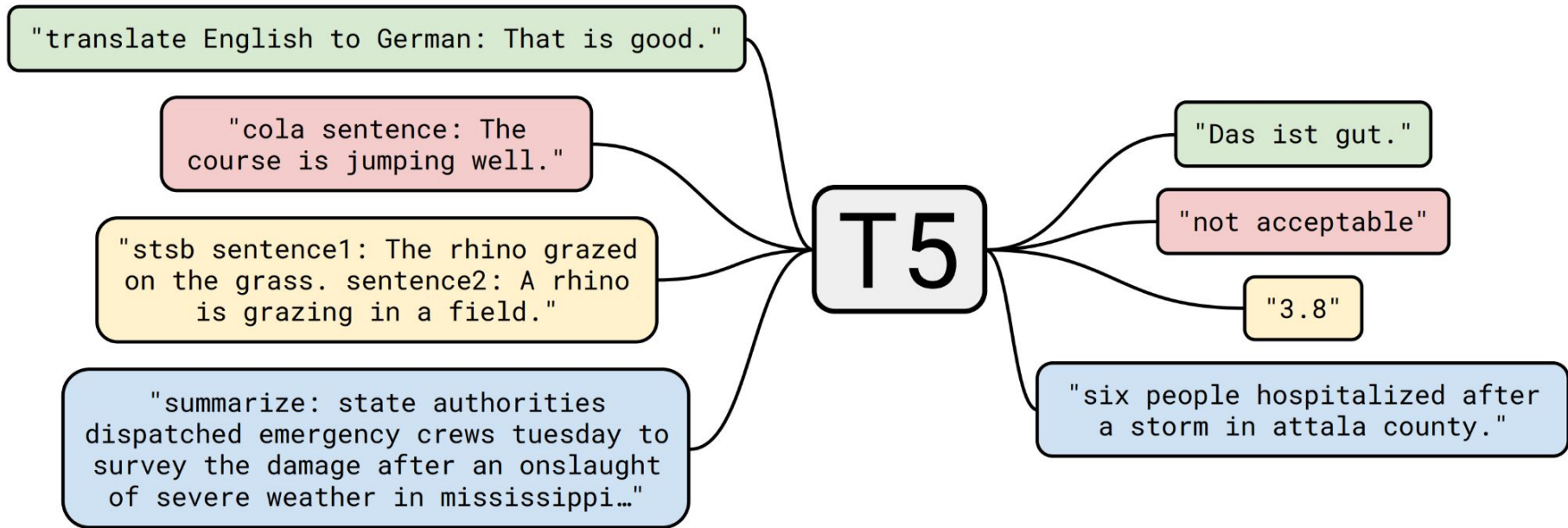


Transformer decoder
(masked)

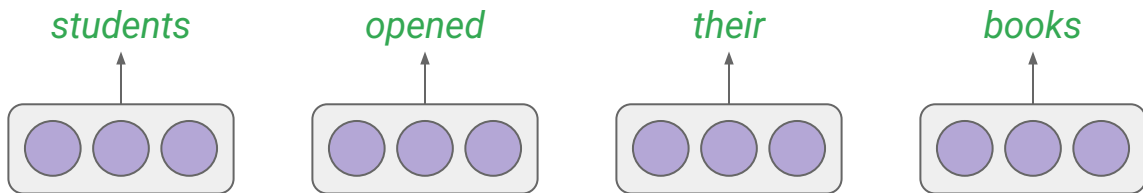
<BOS> positive

decoder

T5 facilitates multitask learning

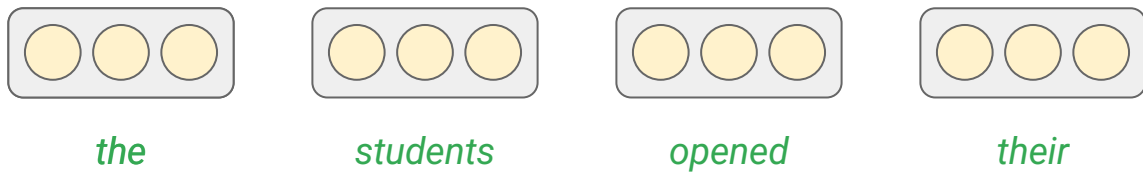


Decoder-only model



***the architecture
used in frontier
LLMs***

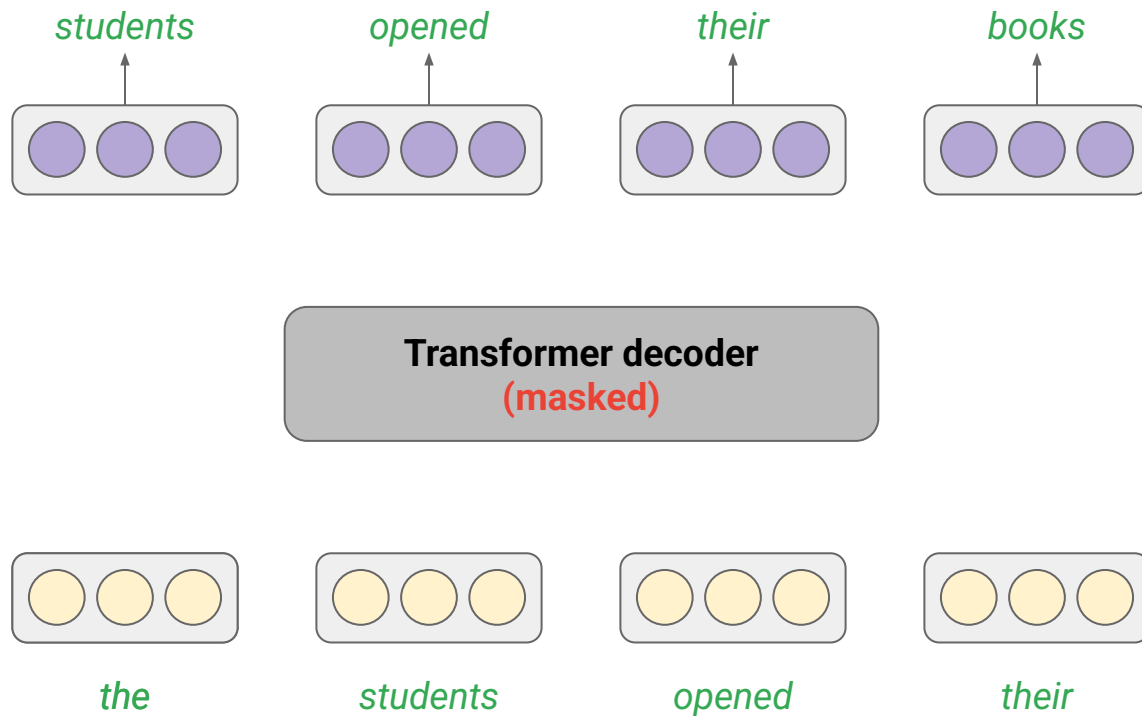
**Transformer decoder
(masked)**



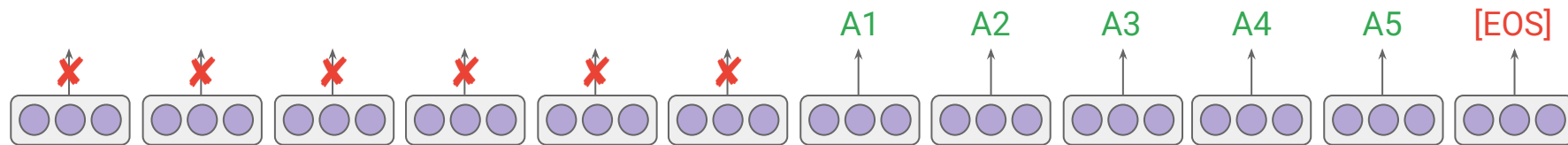
Note on cross-attention

- Can be used to inject non-text data (e.g., images, structured data, or even sensor readings) into the model

Pretraining with a causal LM (decoder-only)

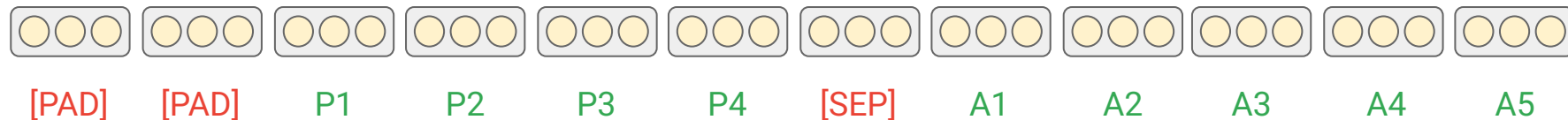


Training with prefix LM (decoder-only)

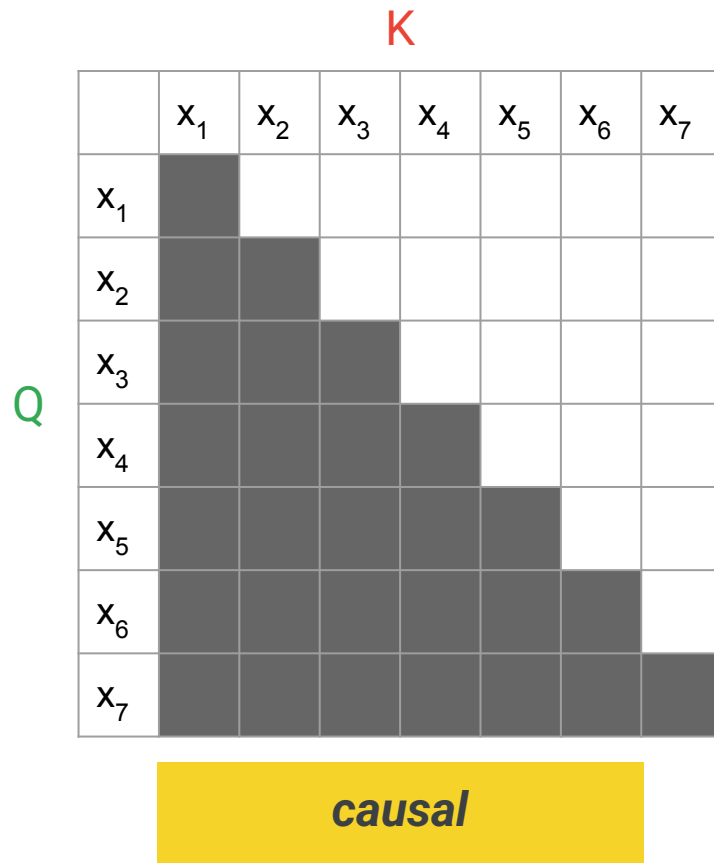
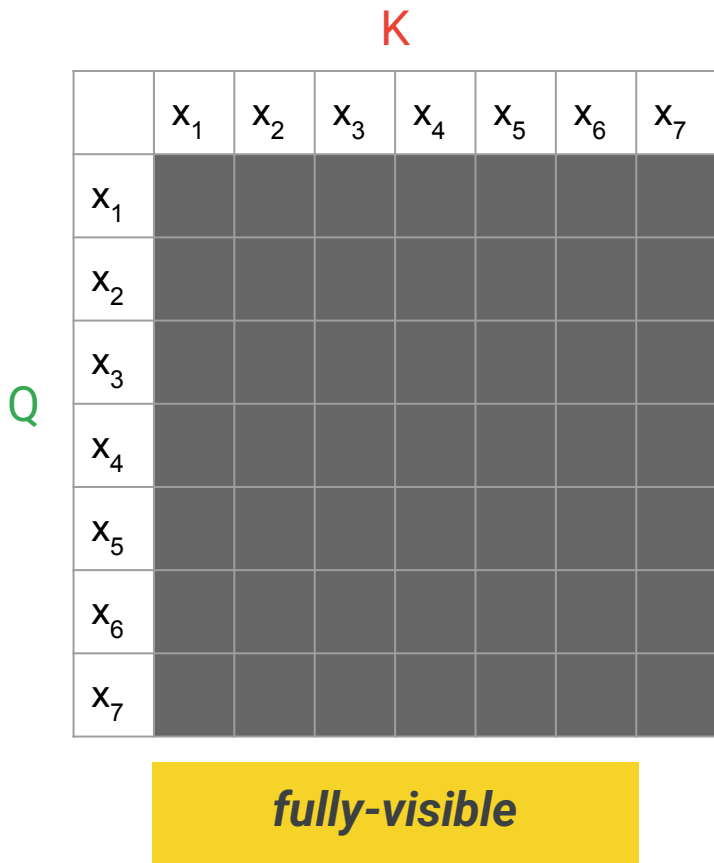


*the architecture
used in frontier
LLMs*

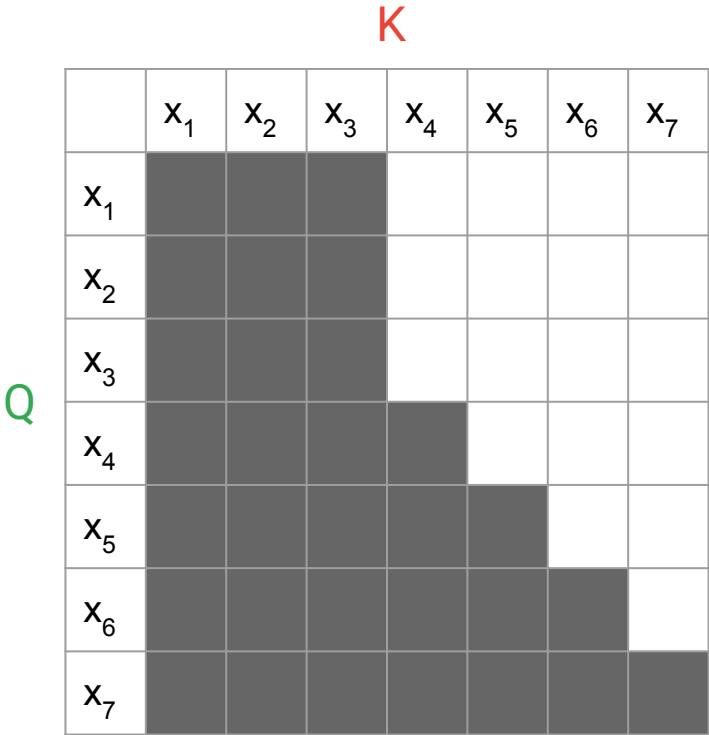
Transformer decoder
(partially masked)



Different attention mask patterns

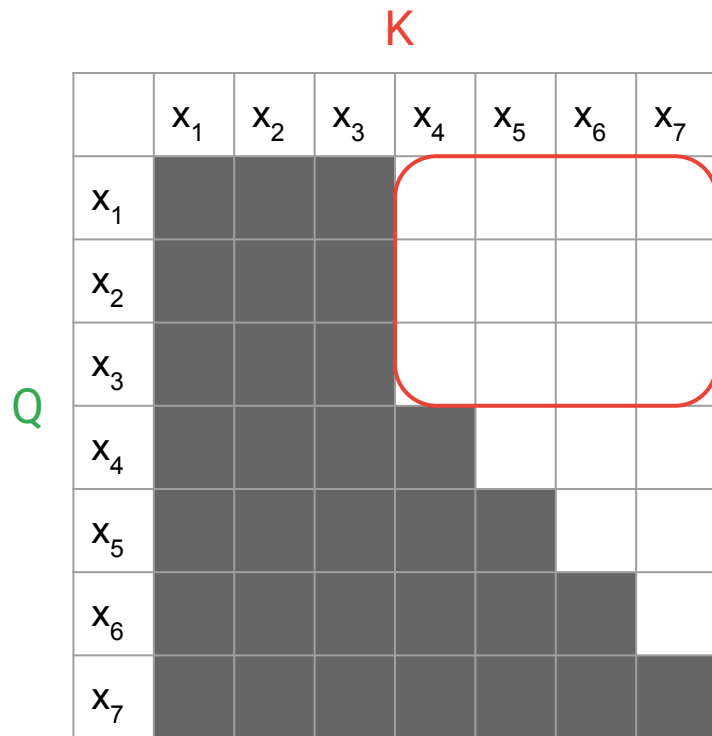


Different attention mask patterns (cont'd)



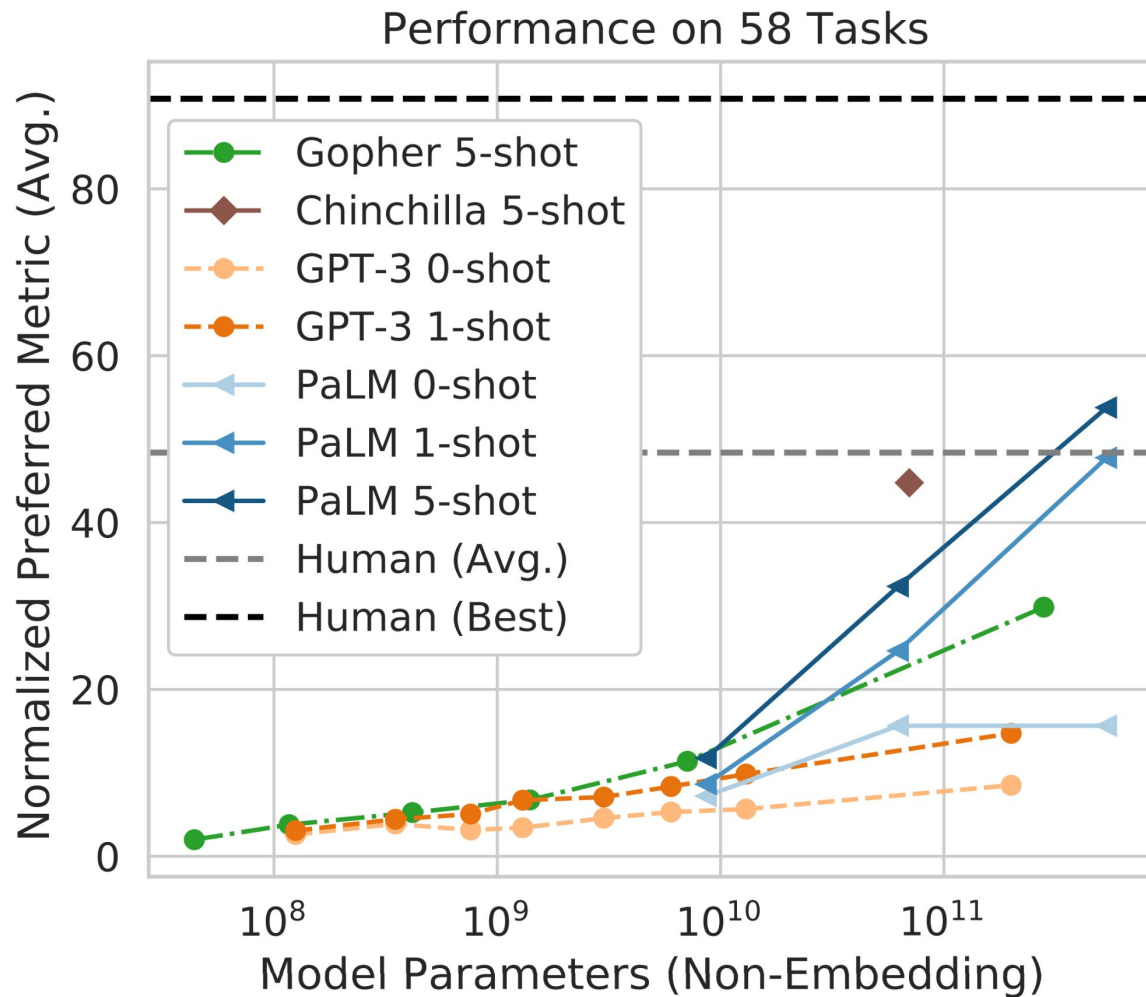
Prefix LM

Different attention mask patterns (cont'd)

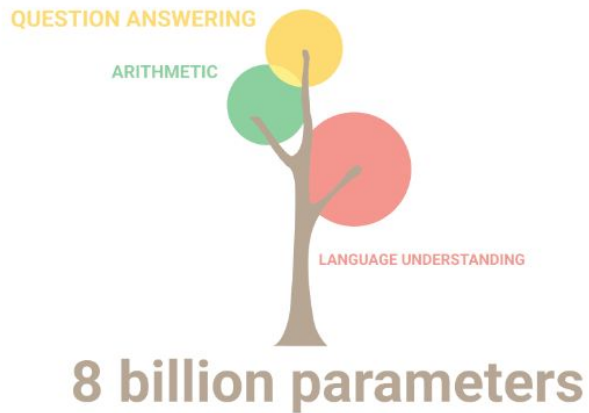


***Why masking
here?***

**Increasing model
size enhances
performance and
sample efficiency**

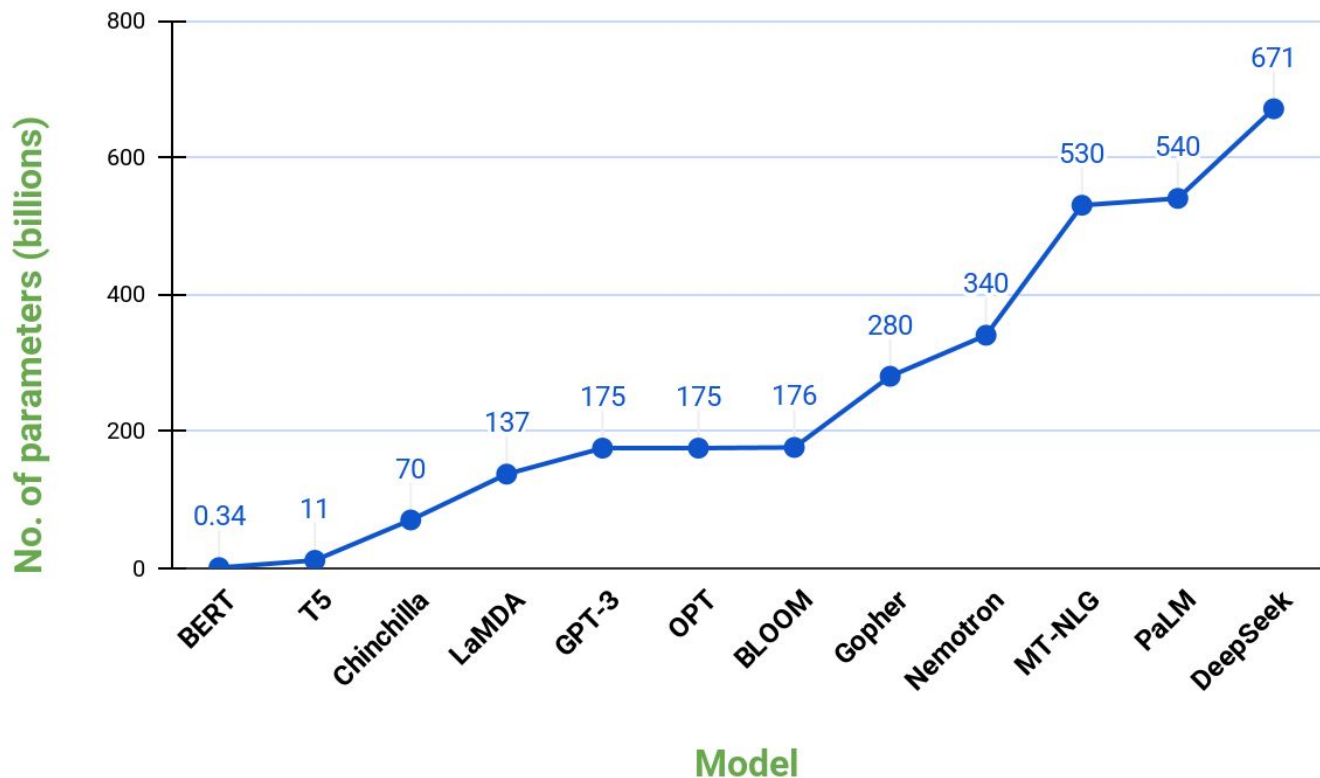


... and unlocks new capabilities



From "PaLM: Scaling Language Modeling with Pathways" by Chowdhery et al. (2022)

The trend has continued to push the boundaries of possibility in NLP



3.6 bits per parameter

How much do language models memorize?

John X. Morris^{1,3}, Chawin Sitawarin², Chuan Guo¹, Narine Kokhlikyan¹, G. Edward Suh^{3,4},
Alexander M. Rush³, Kamalika Chaudhuri¹, Saeed Mahloujifar¹

¹FAIR at Meta, ²Google DeepMind, ³Cornell University, ⁴NVIDIA

We propose a new method for estimating how much a model “knows” about a datapoint and use it to measure the capacity of modern language models. We formally separate memorization into two components: *unintended memorization*, the information a model contains about a specific dataset, and *generalization*, the information a model contains about the true data-generation process. By eliminating generalization, we can compute the total memorization of a given model, which provides an estimate of model capacity: our measurements estimate that **models in the GPT family have an approximate capacity of 3.6 bits-per-parameter**. We train language models on datasets of increasing size and observe that models memorize until their capacity fills, at which point “grokking” begins, and unintended memorization decreases as models begin to generalize. We train hundreds of transformer language models ranging from 500K to 1.5B parameters and produce a series of scaling laws relating model capacity and data size to membership inference.

Date: June 19, 2025

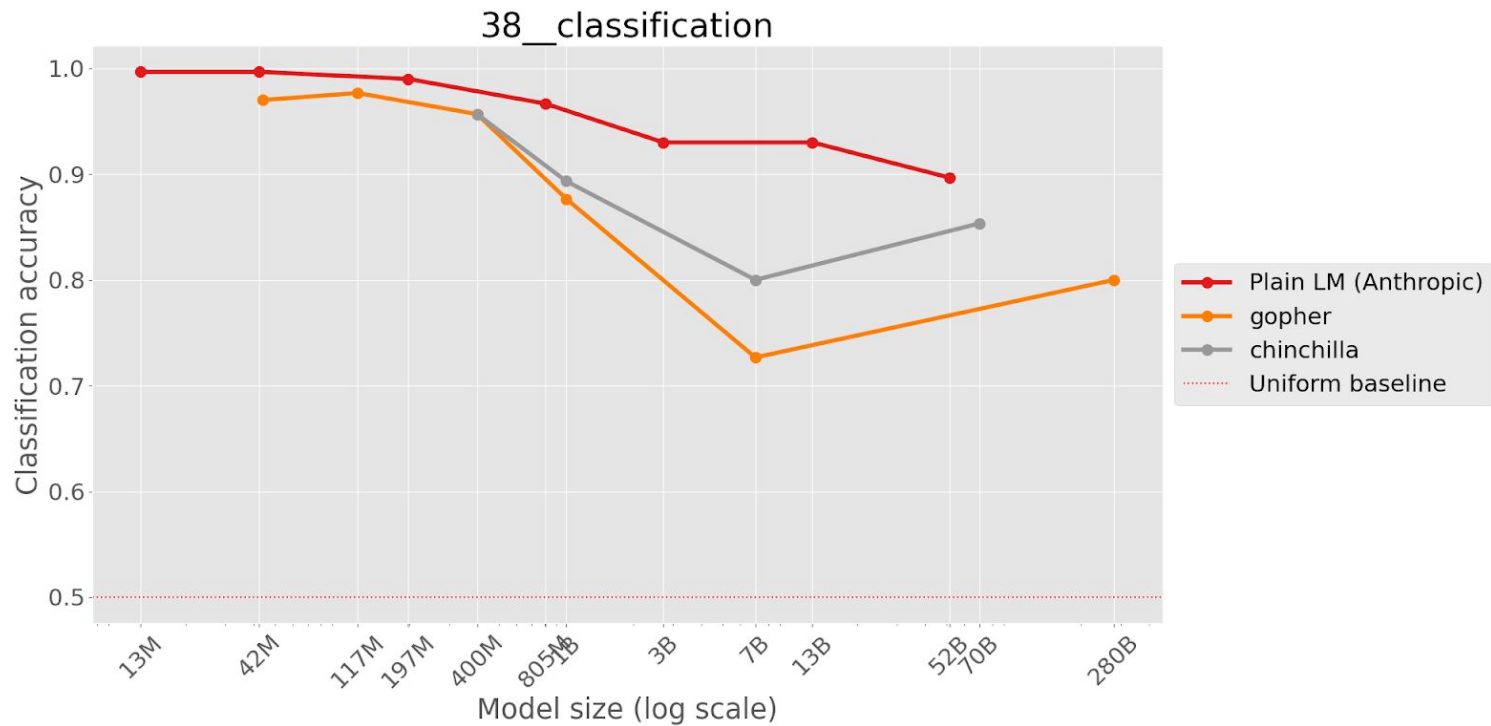
Correspondence: Saeed Mahloujifar at saeedm@meta.com



Inverse scaling

- <https://www.lesswrong.com/posts/iznohbCPFkeB9kAJL/inverse-scaling-prize-round-1-winners>
- <https://www.lesswrong.com/posts/DARiTSTx5xDLQGrrz/inverse-scaling-prize-second-round-winners>

Inverse scaling (cont'd)



Repeat my sentences back to me.

Input: I like dogs.

Output: I like dogs.

Input: What is a potato, if not big?

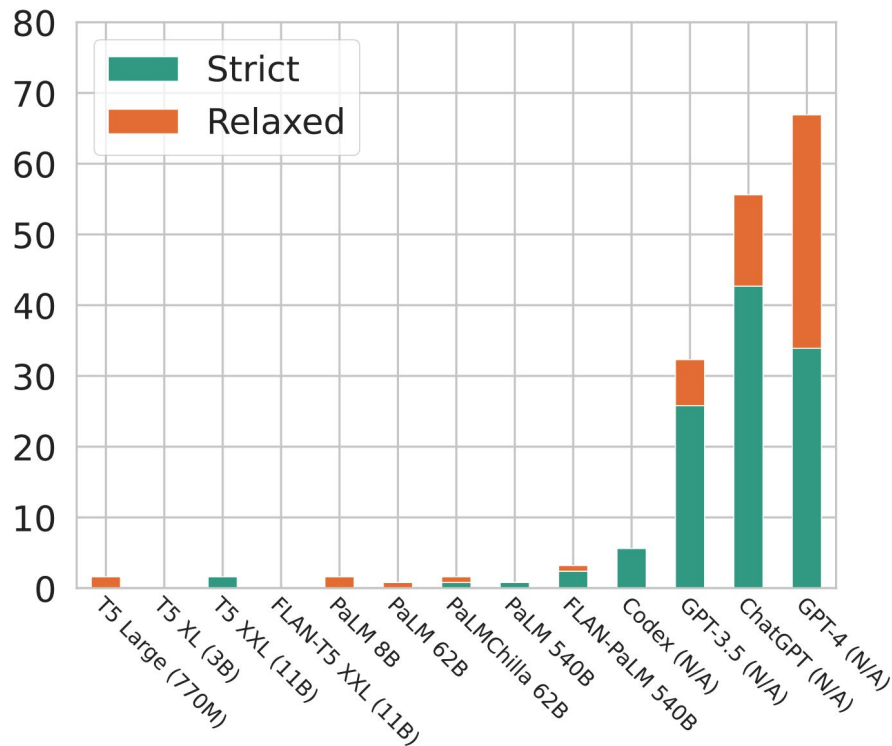
Output: What is a potato, if not big?

Input: All the world's a stage, and all the men and women merely players. They have their exits and their entrances; And one man in his time plays many pango

Output: All the world's a stage, and all the men and women merely players. They have their exits and their entrances; And one man in his time plays many

(where the model should choose 'pango' instead of completing the quotation with 'part.')

False premise questions: When did Google release ChatGPT?



False-premise questions

Vu et al. 2023:
<https://arxiv.org/abs/2310.03214>

What can we scale?

- The loss scales as a power-law with:
 - **N**: model size
 - **D**: dataset size
 - the amount of compute used for training (e.g., number of training steps)

$$\text{Total Steps} = \frac{\text{Dataset Size} \times \text{Epochs}}{\text{Batch Size}}$$

Where:

- **Dataset Size:** Total number of training examples.
- **Epochs:** Number of times the model sees the entire dataset.
- **Batch Size:** Number of samples per batch update.

Given a fixed compute budget, what is the optimal model size and dataset size for training?

Let's say you can use one GPU for one day

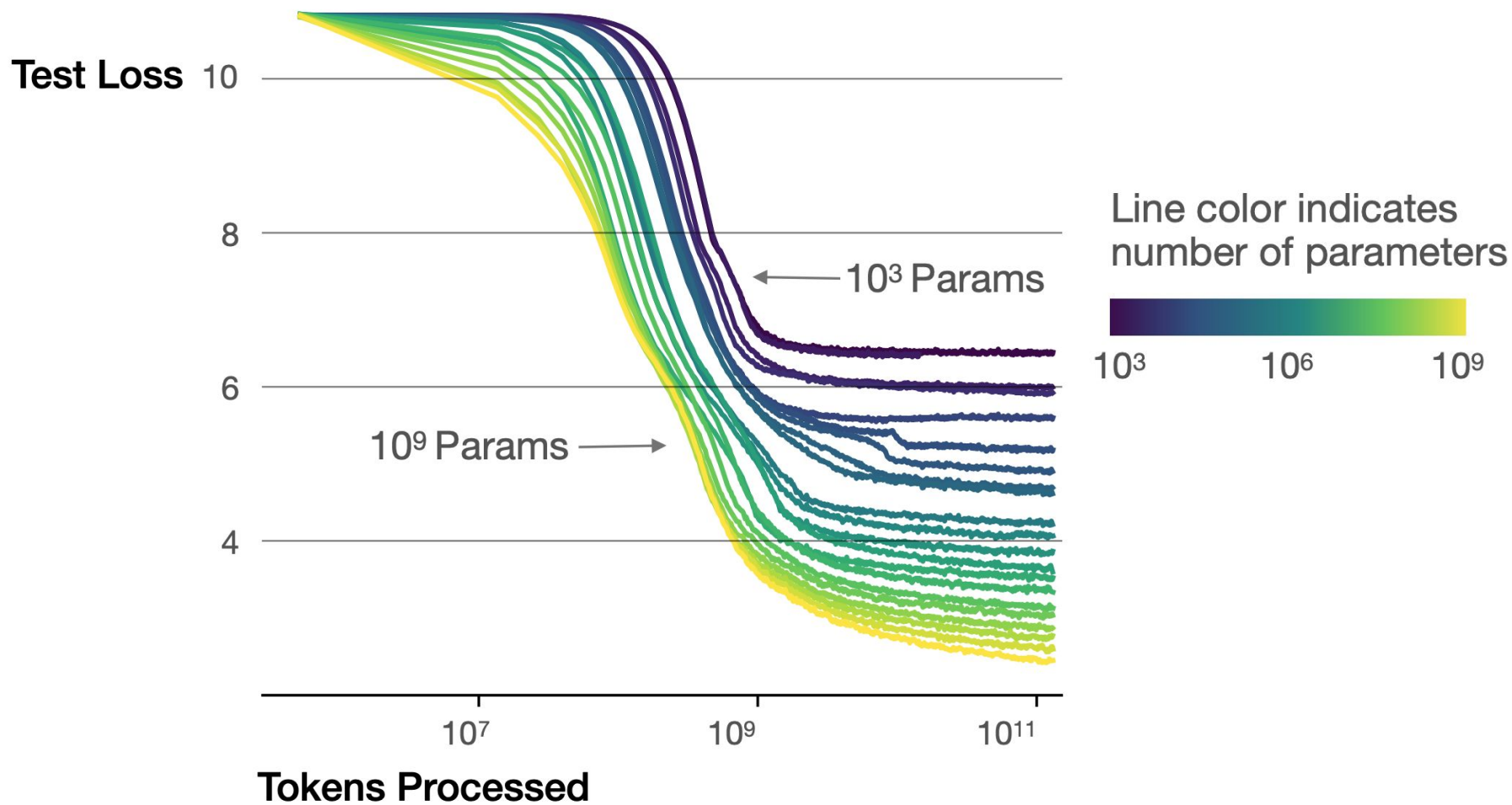
- Would you train a 5 million parameter LM on 100 books?
- What about a 500 million parameter LM on one book?
- Or a 100k parameter LM on 5k books?

Given a fixed compute budget, what is the optimal model size and dataset size for training?

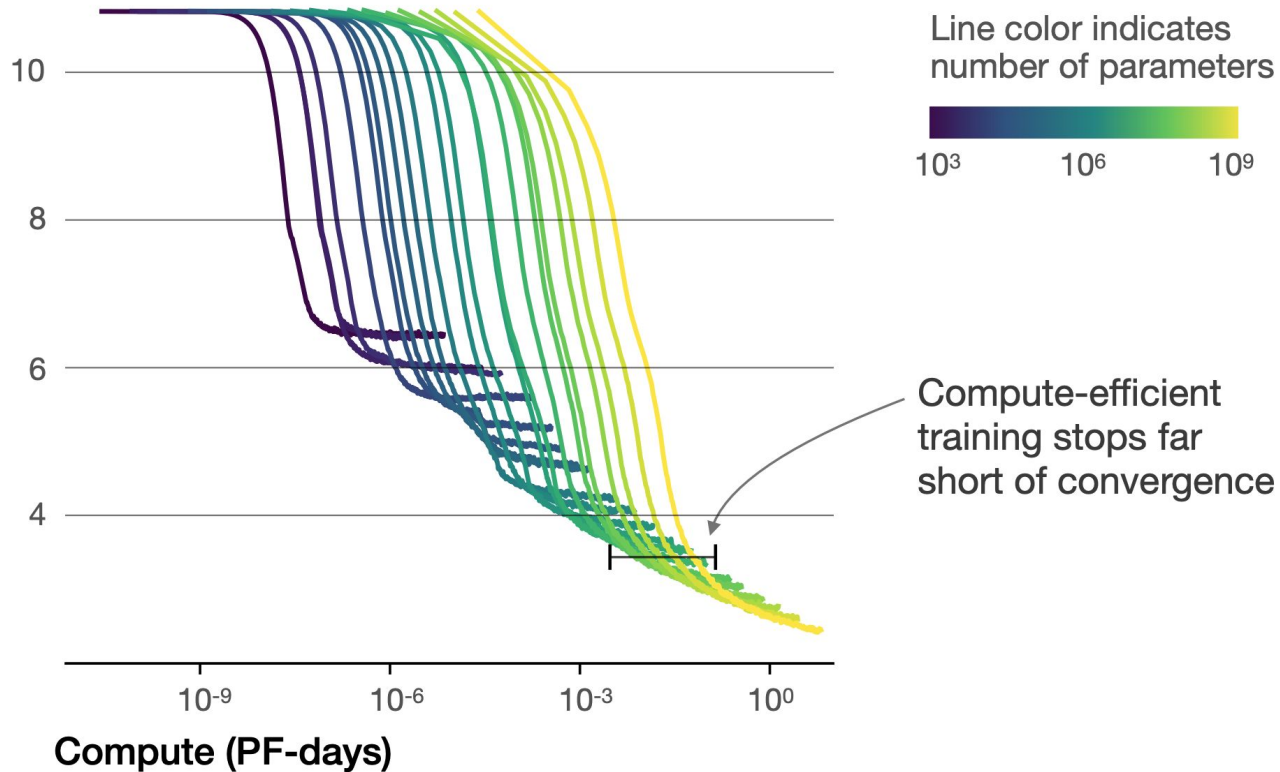
- Kaplan et al. 2020
- **Chinchilla** (Hoffmann et al. 2022)

Observations from Kaplan et al., 2020

- Performance depends largely on scale (model size, data size, and compute) and weakly on model architecture (e.g., depth, width)
- Performance improves most when model and dataset size scale together; increasing one while keeping the other fixed results in diminishing returns



The optimal model size grows smoothly
with the loss target and compute budget



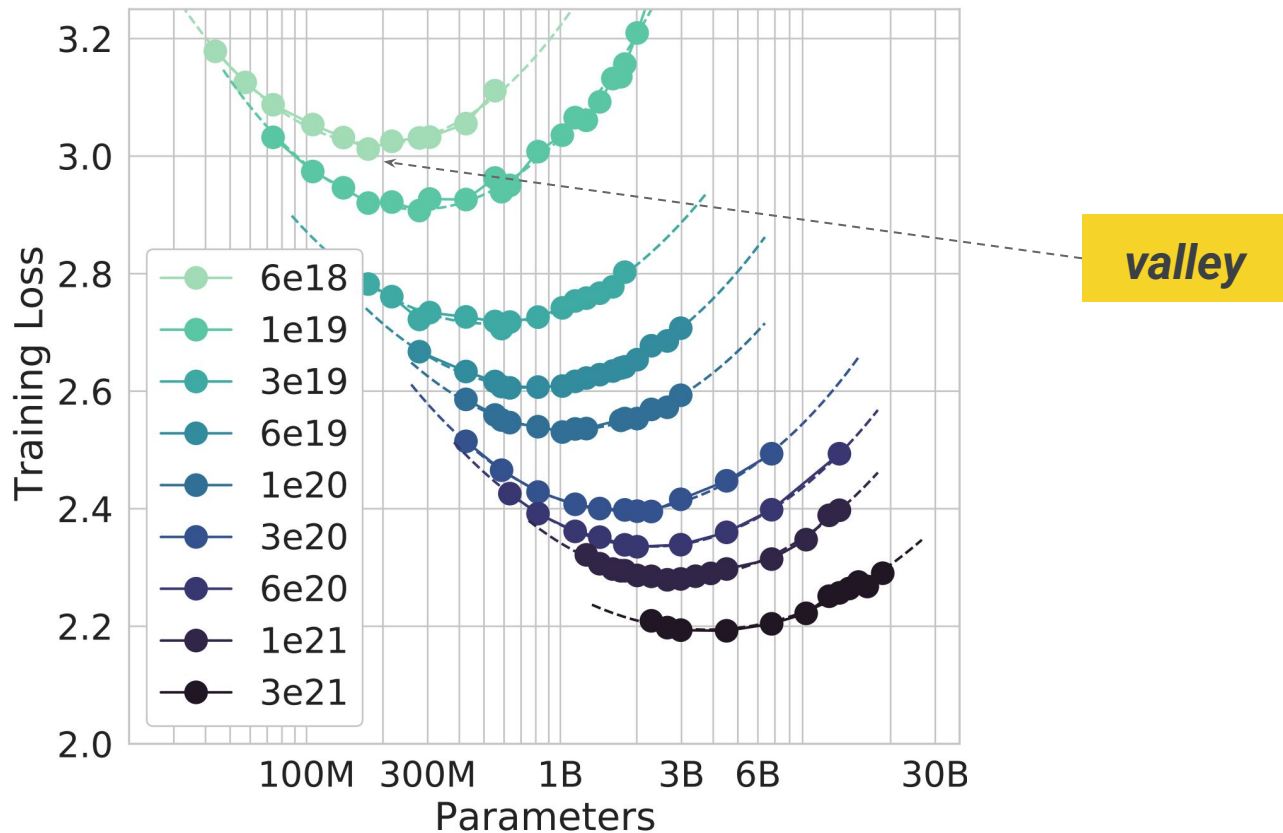
Issues with Kaplan laws

- Used same learning rate schedule for all training runs, regardless of how many training tokens / batches!
- This schedule needs to be adjusted based on the number of training steps; otherwise, it can impair performance
- The resulting “scaling laws” from Kaplan et al., are flawed because of this!

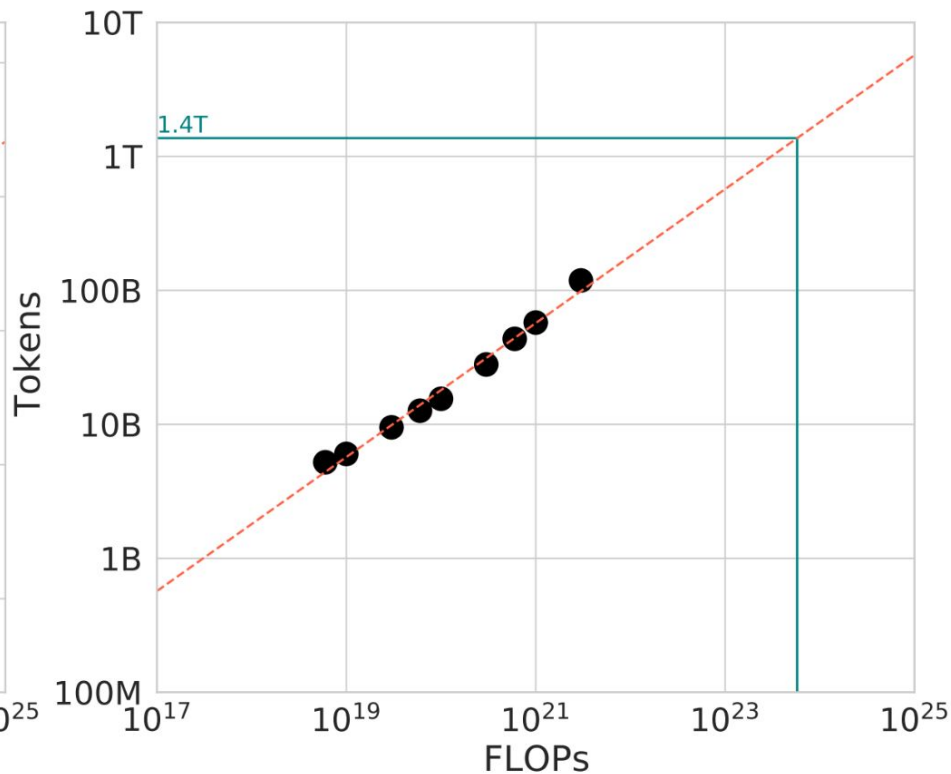
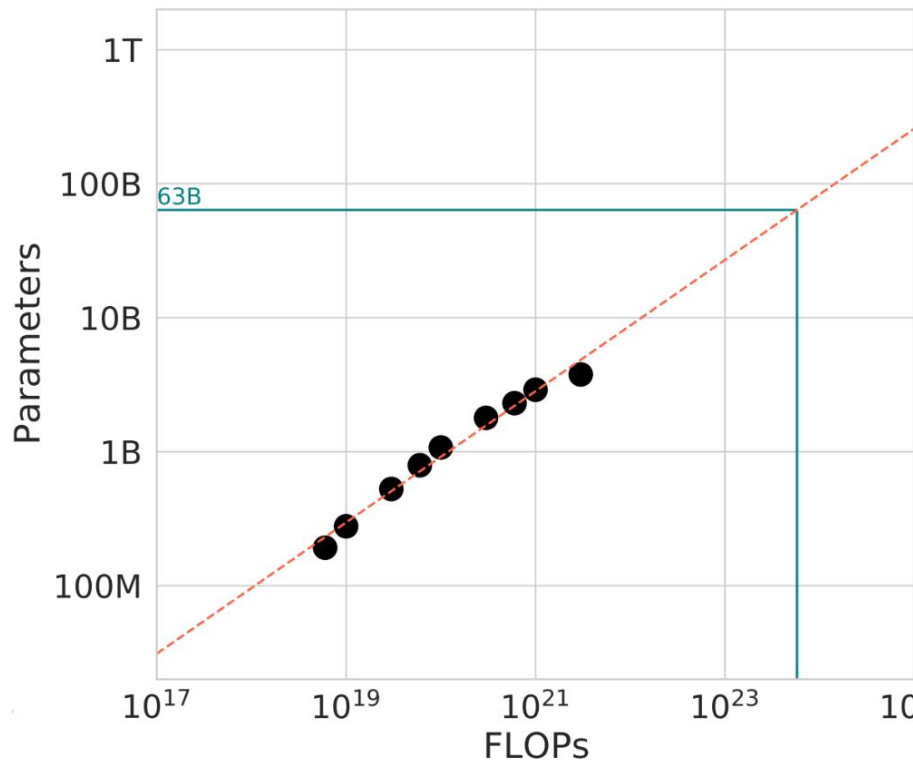
Chinchilla scaling laws

- **Kaplan et al., 2020: prioritize increasing model size over data size**
 - With a 10x compute increase, increase model size by 5x and data size by 2x
 - With a 100x compute increase, model size 25x and data 4x
- **Chinchilla (Hoffmann et al., 2022): increase model and data size at the same rate**
 - With a 10x compute increase, increase both model size and data size by 3.1x
 - With a 100x compute increase, both model and data size 10x

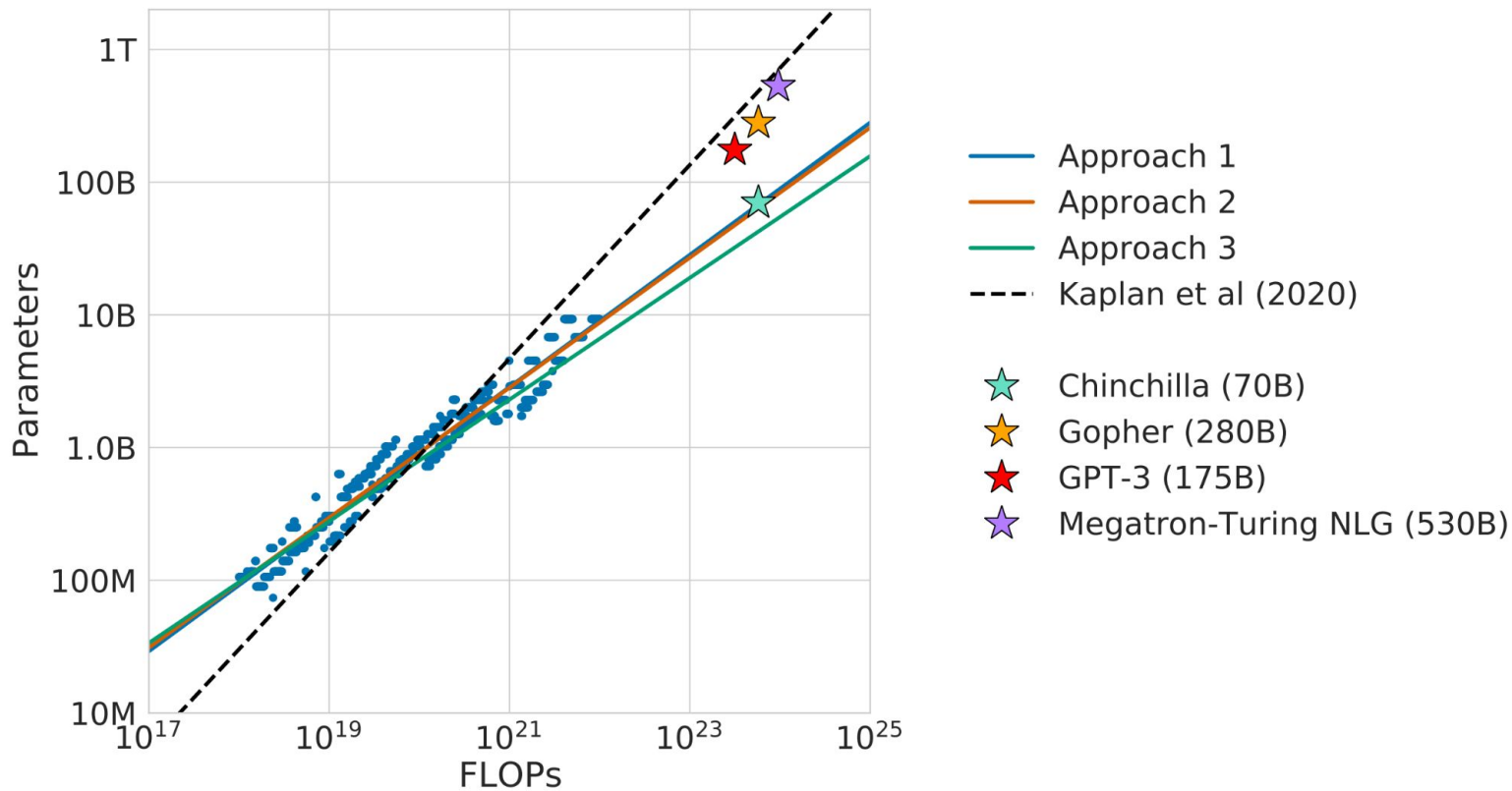
For a given FLOP budget there is an optimal model to train



Projecting optimal model size and number of tokens for larger models



Large models should be significantly smaller and trained for much longer than is currently done (2022)



Large models should be significantly smaller and trained for much longer than is currently done (2022)

Model	Size (# Parameters)	Training Tokens
LaMDA (Thoppilan et al., 2022)	137 Billion	168 Billion
GPT-3 (Brown et al., 2020)	175 Billion	300 Billion
Jurassic (Lieber et al., 2021)	178 Billion	300 Billion
Gopher (Rae et al., 2021)	280 Billion	300 Billion
MT-NLG 530B (Smith et al., 2022)	530 Billion	270 Billion
<i>Chinchilla</i>	70 Billion	1.4 Trillion

- N – the number of model parameters, *excluding all vocabulary and positional embeddings*
- $C \approx 6NBS$ – an estimate of the total non-embedding training compute, where B is the batch size, and S is the number of training steps (ie parameter updates). We quote numerical values in PF-days, where one PF-day = $10^{15} \times 24 \times 3600 = 8.64 \times 10^{19}$ floating point operations.

PF: PetaFLOP

Gopher vs. Chinchilla

Random	25.0%
Average human rater	34.5%
GPT-3 5-shot	43.9%
<i>Gopher</i> 5-shot	60.0%
<i>Chinchilla</i> 5-shot	67.6%
Average human expert performance	89.8%
June 2022 Forecast	57.1%
June 2023 Forecast	63.4%

Table 6 | **Massive Multitask Language Understanding (MMLU)**. We report the average 5-shot accuracy over 57 tasks with model and human accuracy comparisons taken from [Hendrycks et al. \(2020\)](#). We also include the average prediction for state of the art accuracy in June 2022/2023 made by 73 competitive human forecasters in [Steinhardt \(2021\)](#).

Chinchilla's loss function

$$\hat{L}(N, D) \triangleq E + \frac{A}{N^\alpha} + \frac{B}{D^\beta}.$$

Thank you!