Deep Learning (1470)

Randall Balestriero

Class 14: Large Language Models

Recap!

Causal Self-Attention

Masked attention for autoregressive models (GPT, LLaMA, etc.)

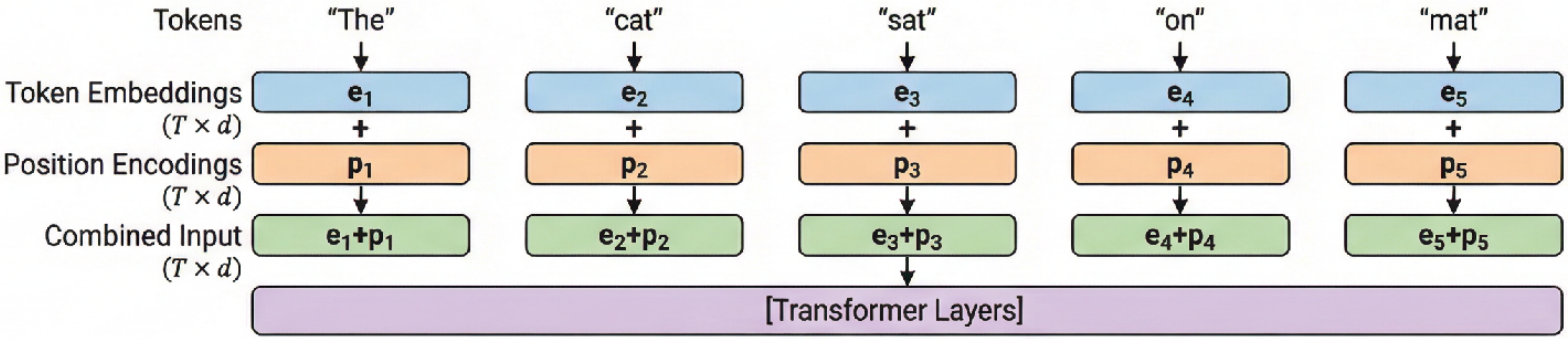
$$\text{Causal Attention:} \quad \mathbf{A} = \text{softmax}\left(\frac{\mathbf{Q}\cdot\mathbf{K}^T + \mathbf{M}}{\sqrt{d_k}}\right) \qquad \mathbf{Z} = \mathbf{A}\cdot\mathbf{V} \qquad \text{where } M_{ij} = 0 \text{ if } j \le i, \text{ else } -\infty$$

How to encode position?

Positional Encodings

Injecting position information into the transformer

SECTION 1: WHERE TO ADD



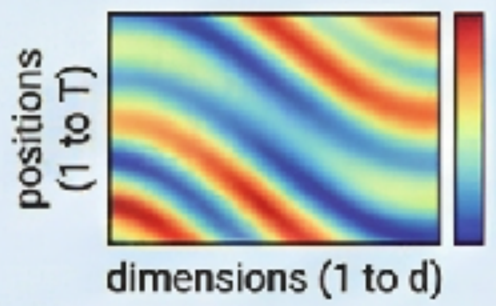
Position encoding added to token embeddings **BEFORE** transformer layers

SECTION 2: TYPES OF POSITIONAL ENCODINGS

Sinusoidal (Original Transformer)

$$\mathrm{PE}(pos, 2i) = \sin\left(\frac{pos}{10000^{2i/d}}\right)$$

$$\mathrm{PE}(pos, 2i{+}1) = \cos\left(\frac{pos}{10000^{2i/d}}\right)$$

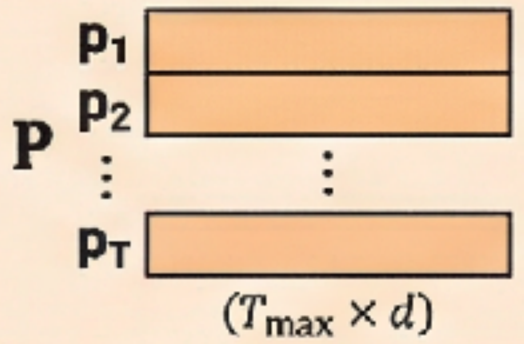


- Fixed (not learned)
- Deterministic
- Can extrapolate to longer sequences

Learned (BERT, GPT-2)

$$\mathbf{P} \in \mathbb{R}^{T_{\max} \times d}$$

"Lookup table of learnable vectors"

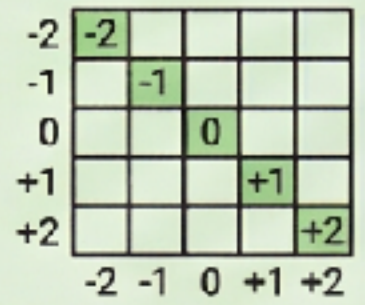


- Learned during training
- More flexible
- Limited to max sequence length T_{max}

Relative (Transformer-XL, T5)

"Encode relative distance $(i - j)$ not absolute position"

a_{ij} depends on $(i - j)$



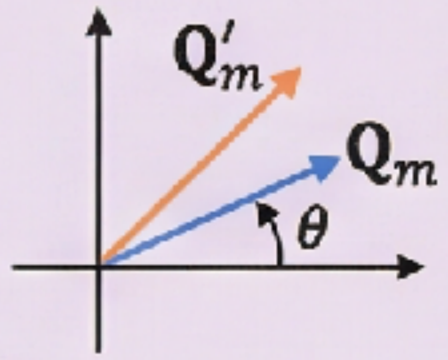
- Captures relative distance
- Better for long sequences
- Added in attention computation

RoPE / Rotary (LLaMA, GPT-NeoX)

"Rotate **Q** and **K** vectors based on position"

$$\mathbf{Q}'_m = \mathbf{R}_m \cdot \mathbf{Q}_m$$

$$\mathbf{K}'_n = \mathbf{R}_n \cdot \mathbf{K}_n$$



- Applied to **Q, K** in attention
- Relative position via rotation
- Extrapolates well

SECTION 3: VISUAL COMPARISON

Method	Where Added	Learned?	Extrapolation
Sinusoidal	Input	No	✓ Good
Learned	Input	Yes	✗ Limited
Relative	Attention	Yes	✓ Good
RoPE	Q, K	No	✓ Good

EQUATIONS BOX

Input to transformer:

$$\mathbf{X} = \text{TokenEmbed(tokens)} + \text{PositionEncode(positions)}$$

For sinusoidal:

$$\mathrm{PE}(t, 2i) = \sin\left(\frac{t}{10000^{2i/d}}\right) \qquad \mathrm{PE}(t, 2i{+}1) = \cos\left(\frac{t}{10000^{2i/d}}\right)$$

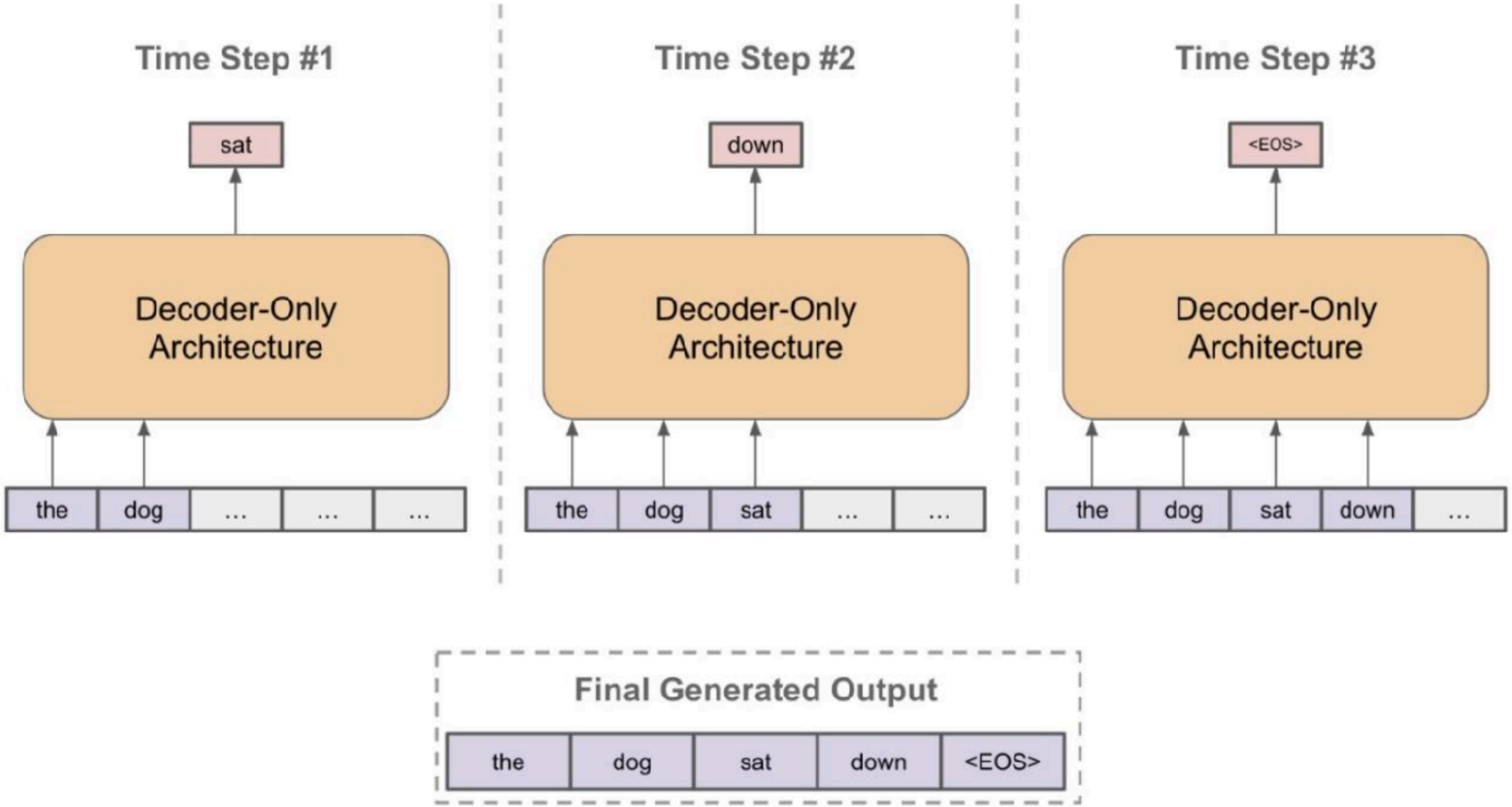
LLM Hyperparameters

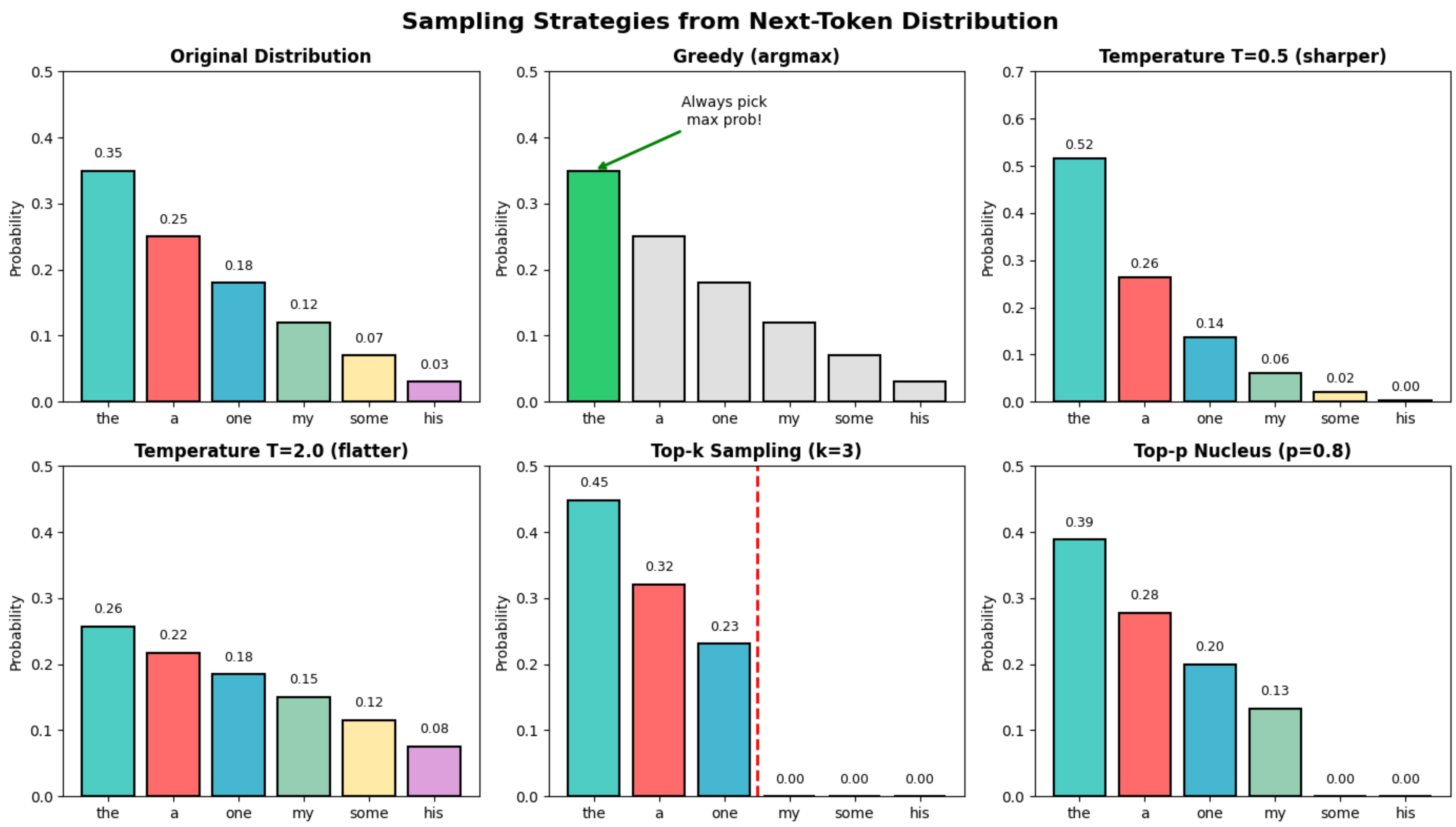
	OLMo-7B	LLaMA2-7B	OpenLM-7B	Falcon-7B	PaLM-8B
Dimension	4096	4096	4096	4544	4096
Num heads	32	32	32	71	16
Num layers	32	32	32	32	32
MLP ratio	~8/3	~8/3	~8/3	4	4
Layer norm type	non-parametric	RMSNorm	parametric	parametric	parametric
Positional embeddings	RoPE	RoPE	RoPE	RoPE	RoPE
Attention variant	full	GQA	full	MQA	MQA
Biases	none	none	in LN only	in LN only	none
Block type	sequential	sequential	sequential	parallel	parallel
Activation	SwiGLU	SwiGLU	SwiGLU	GeLU	SwiGLU
Sequence length	2048	4096	2048	2048	2048
Batch size (instances)	2160	1024	2048	2304	512
Batch size (tokens)	~4M	~4M	~4M	~4M	~1M
Weight tying	no	no	no	no	yes

Table 2: LM architecture comparison at the 7–8B scale. In the "layer norm type" row, "parametric" and "non-parametric" refer to the usual layer norm implementation with and without adaptive gain and bias, respectively.

What do we do after training?

Generating Autoregressive Output




Sampling Strategies from Next-Token Distribution
Original Distribution
Greedy (argmax)
Always pick max prob!
Temperature T=0.5 (sharper)
Temperature T=2.0 (flatter)
Top-k Sampling (k=3)
Top-p Nucleus (p=0.8)
Probability
the
a
one
my
some
his
0.35
0.25
0.18
0.12
0.07
0.03
0.52
0.26
0.14
0.06
0.02
0.00
0.26
0.22
0.18
0.15
0.12
0.08
0.45
0.32
0.23
0.39
0.28
0.20
0.13

Can we train ChatGPT now?

Can we train ChatGPT now?

Yes, but only the poor version

Language Modeling is not Enough

Language Modeling is not Enough

- Model is repetitive

Language Modeling is not Enough

- Model is repetitive
- Does not follow instruction

Language Modeling is not Enough

- Model is repetitive
- Does not follow instruction
- Does not feel interactive

Language Modeling is not Enough

- Model is repetitive
- Does not follow instruction
- Does not feel interactive
- Has unconstrained output (dangerous)

Language Modeling is not Enough

- Model is repetitive
- Does not follow instruction
- Does not feel interactive
- Has unconstrained output (dangerous)
- We need to “teach it” what to say more precisely!

Language Modeling is not Enough

- Model is repetitive
- Does not follow instruction
- Does not feel interactive
- Has unconstrained output (dangerous)
- We need to “teach it” what to say more precisely!
    - Supervised finetuning

Language Modeling is not Enough

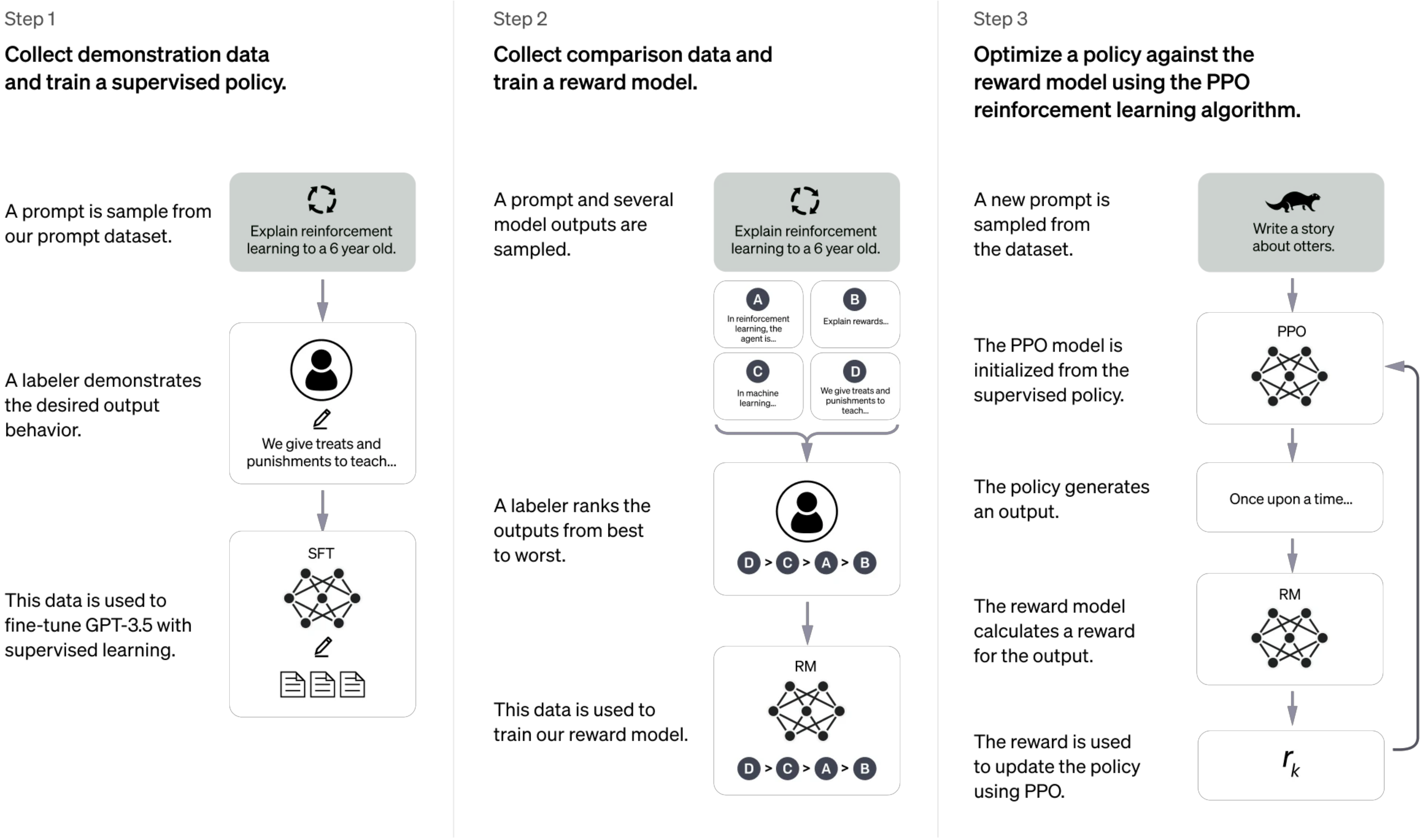
- Model is repetitive
- Does not follow instruction
- Does not feel interactive
- Has unconstrained output (dangerous)
- We need to “teach it” what to say more precisely!
    - Supervised finetuning
    - RLHF

Language Modeling is not Enough

- Model is repetitive
- Does not follow instruction
- Does not feel interactive
- Has unconstrained output (dangerous)
- We need to “teach it” what to say more precisely!
    - Supervised finetuning
    - RLHF
    - …

Turning GPT to Chat-GPT

Step 0: Train GPT



Source: OpenAI

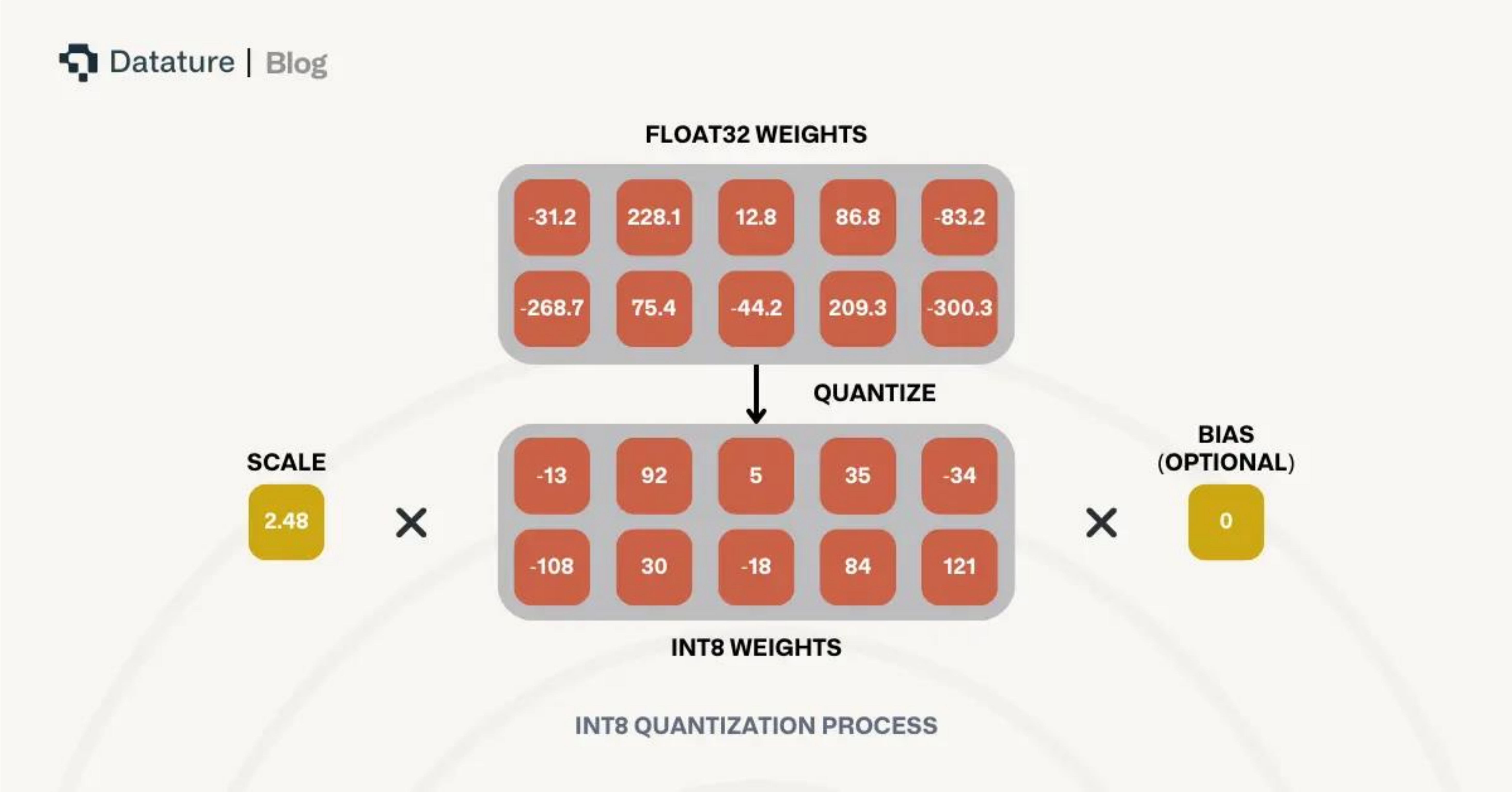
Now that you can generate nice content, how to speed it up?

Quantization

Can we use smaller representation of parameters?

DeepSeek was able to create distilled and quantized models that only used 4 bits per parameter

https://huggingface.co/neuralmagic/DeepSeek-R1-Distill-Llama-8B-quantized.w4a16



But you can do much more

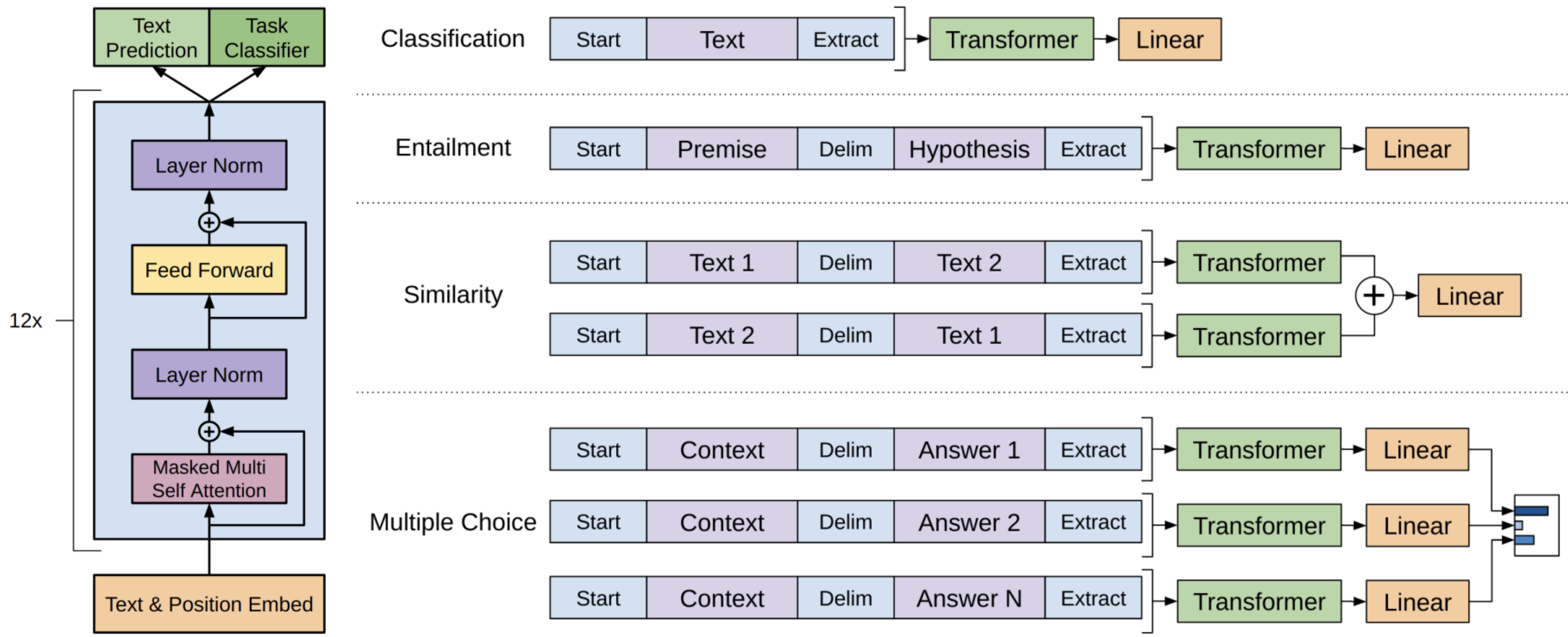


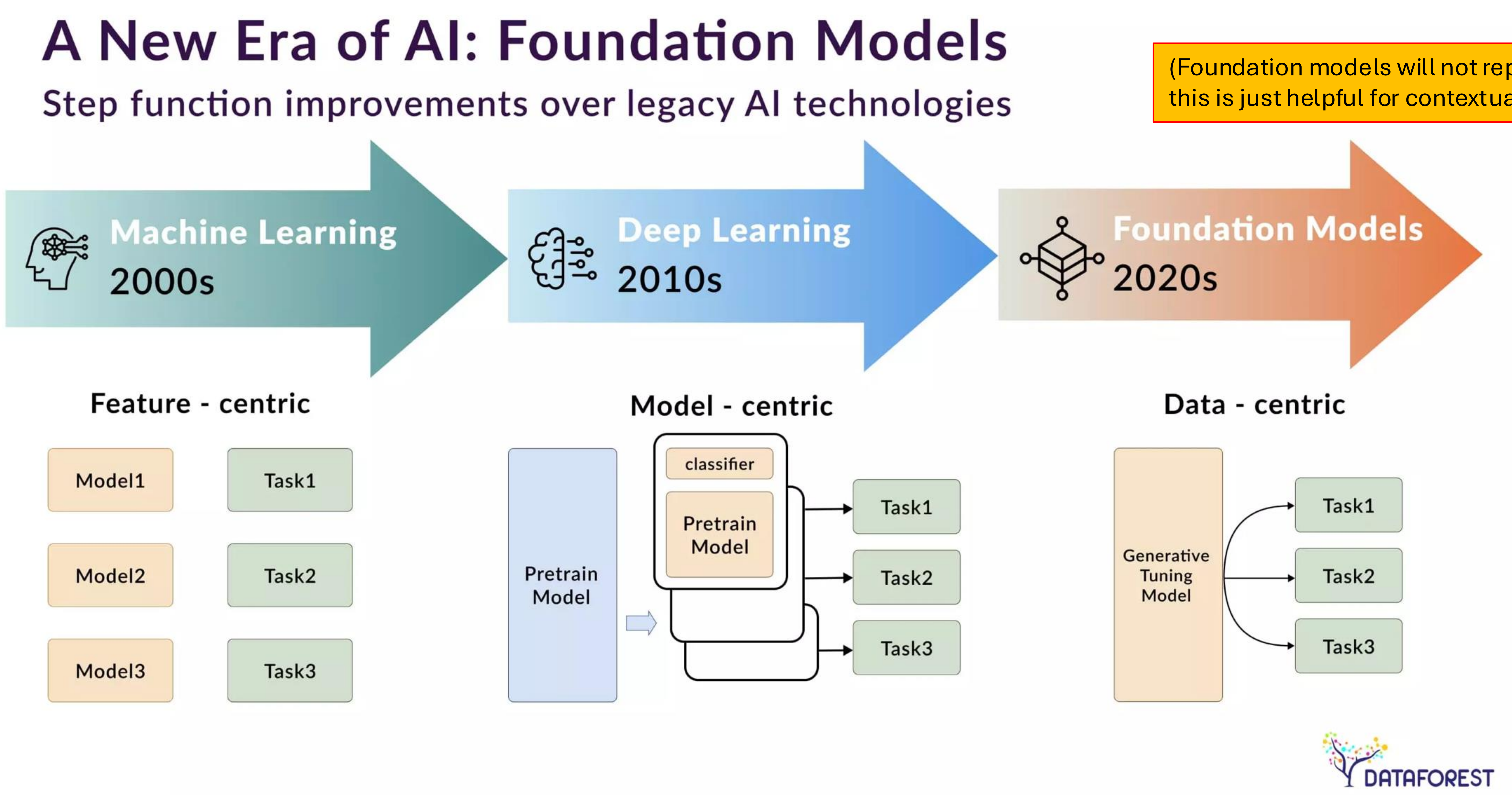
Figure 1: **(left)** Transformer architecture and training objectives used in this work. **(right)** Input transformations for fine-tuning on different tasks. We convert all structured inputs into token sequences to be processed by our pre-trained model, followed by a linear+softmax layer.

Alec Radford et al., Improving Language Understanding by Generative Pre-Training, 2018

The Era of Foundation Models

Foundation Models

(Foundation models will not replace deep learning, this is just helpful for contextualizing the process)



https://dataforest.ai/blog/ai-foundation-models-for-big-business-innovation

Large Language Model Scaling "Laws"

The bigger the better

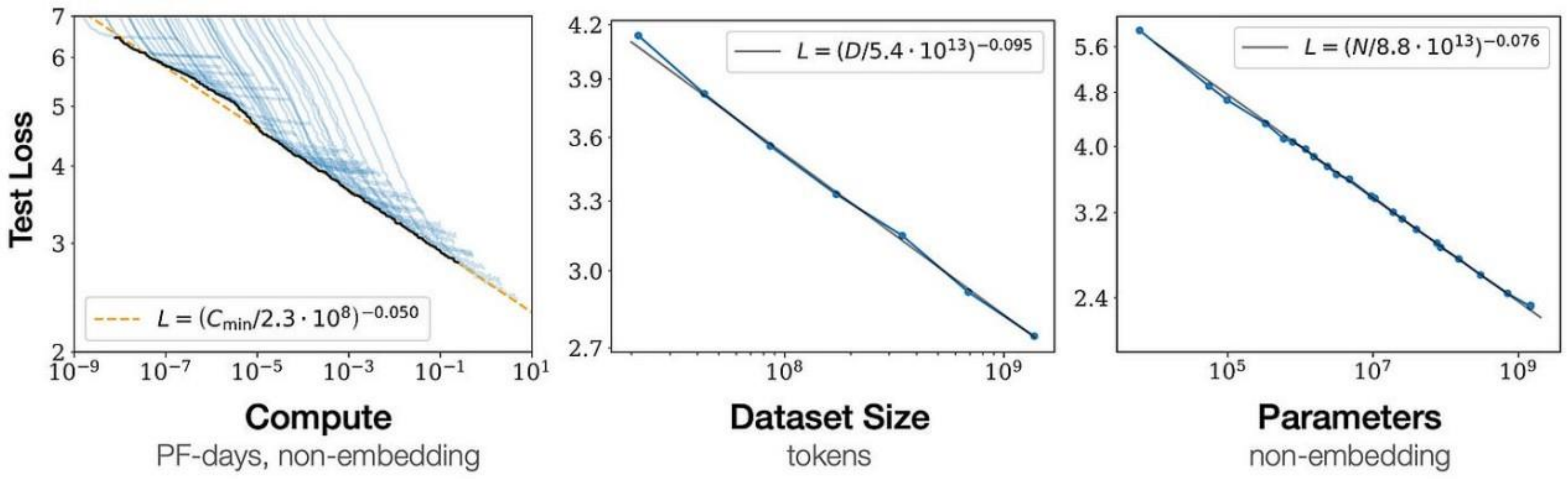
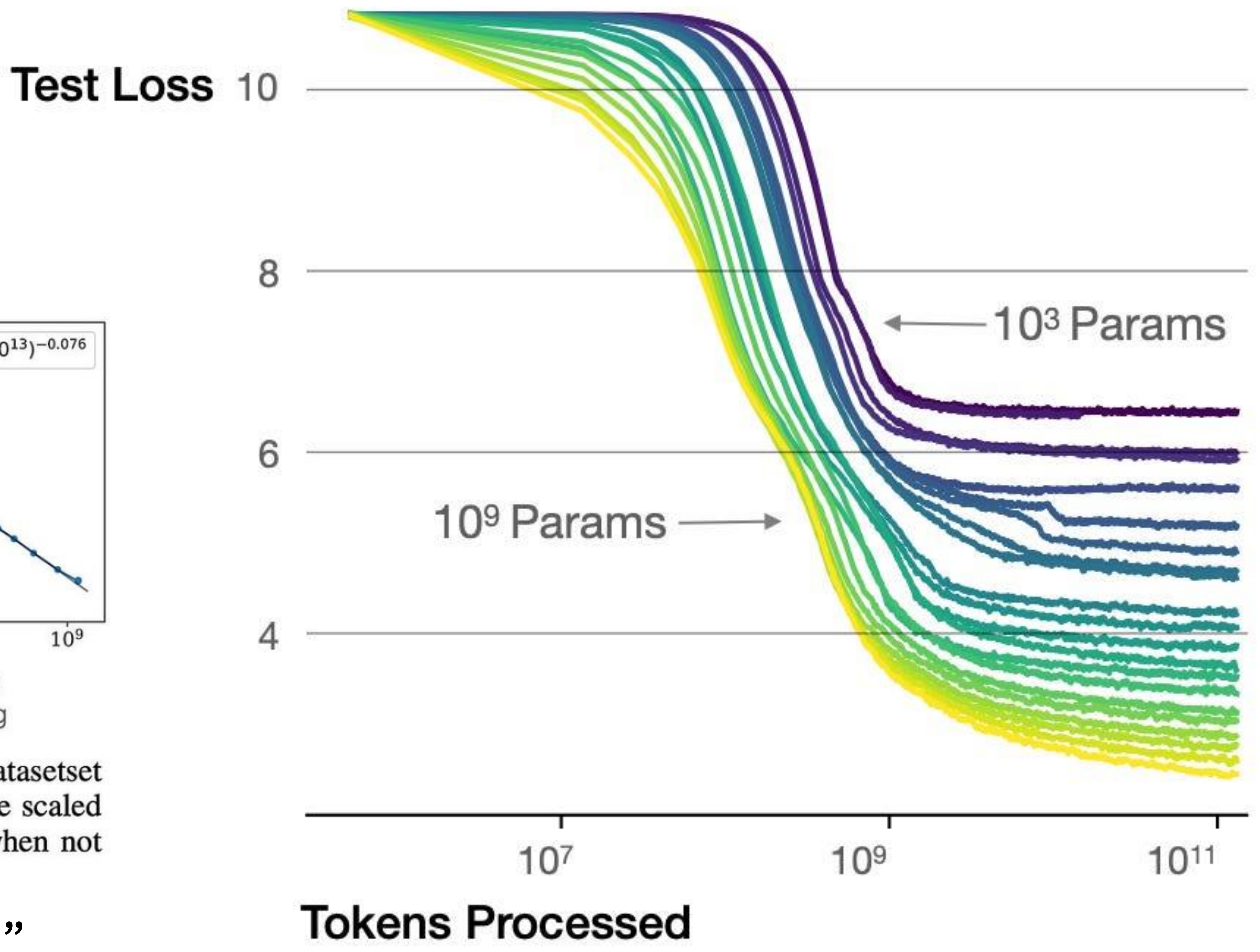


Figure 1 Language modeling performance improves smoothly as we increase the model size, datasetset size, and amount of compute[2] used for training. For optimal performance all three factors must be scaled up in tandem. Empirical performance has a power-law relationship with each individual factor when not bottlenecked by the other two.

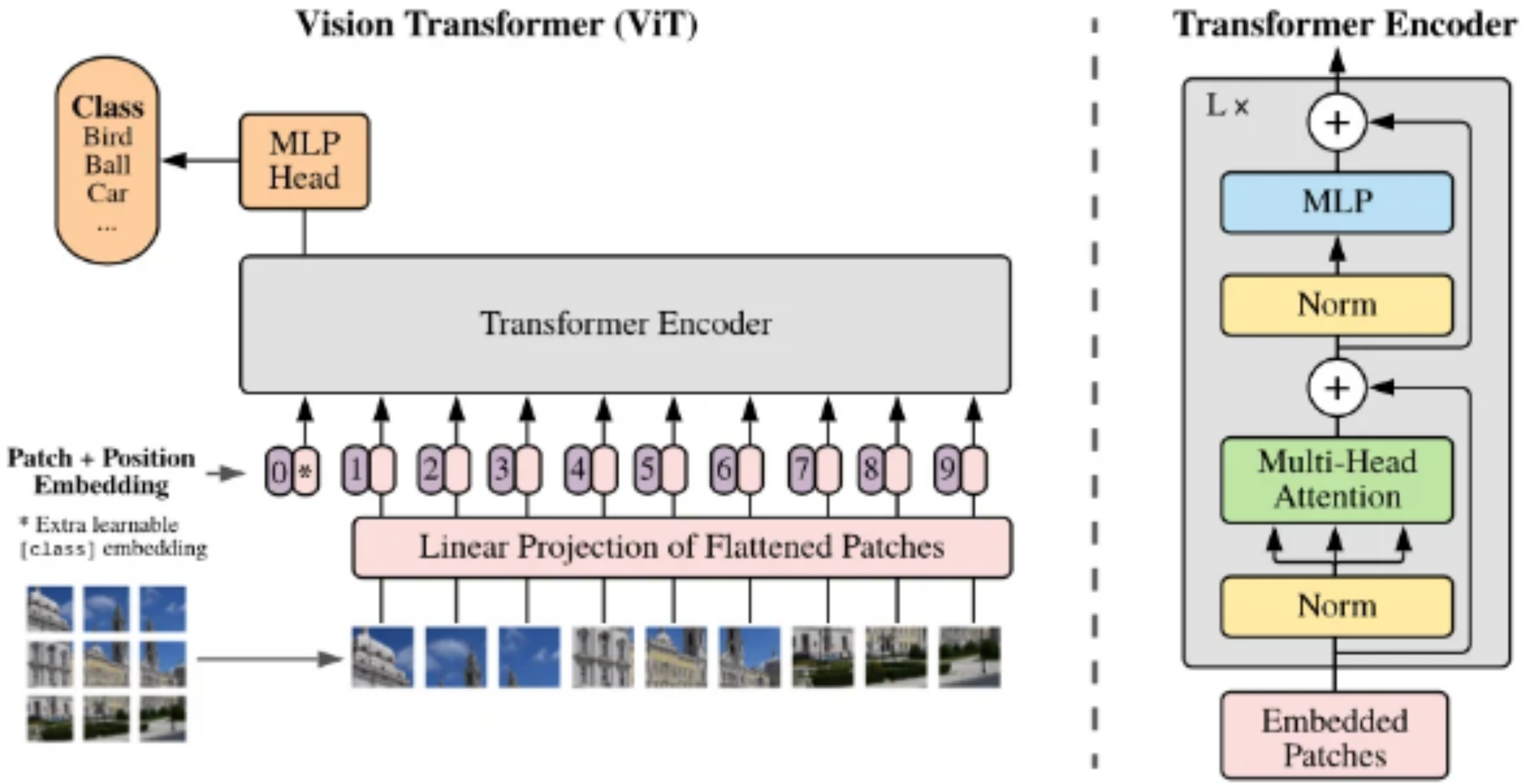
Kaplan et al. "Scaling Laws for Neural Language Models"

Larger models require **fewer samples** to reach the same performance



How to move Beyond Language?

Vision Transformers!



See you on Wednesday!