

Short Intro to Large Language Models (LLM) - a brief educational talk for PEARLS Days

Pardis Niknejadi

1st PEARLS Days Workshop
Freiburg 24.09.2026

HELMHOLTZ RESEARCH FOR
GRAND CHALLENGES

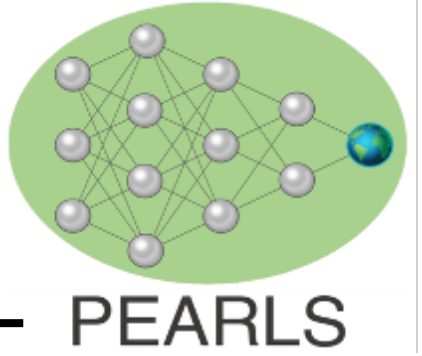


Bundesministerium
für Forschung, Technologie
und Raumfahrt

universität freiburg

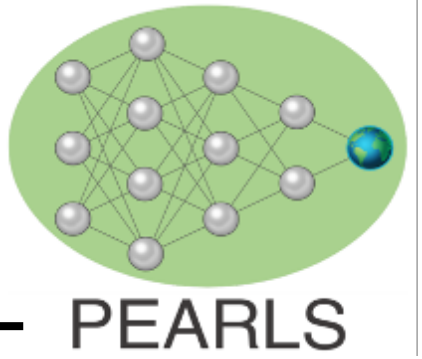


Outline



- What is the dream
- The paper that jumpstarted LLM (a brief journal club)
- Other relevant papers
- Open Question
- My Notes on how to use LLMs
- A brief Outlook

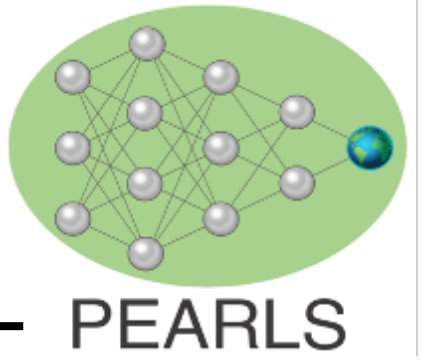
Reminder: "How about before AI-assisted machine setup?"



These train of 3-7 HEXTOF electrons came from macropulses 427-530 for which the GMD measured $41 \mu\text{J}$, with aperture set to 3 mm. Go to machine state during macropulses 427.

I will discuss the challenges of this tomorrow.

How Did LLMs Come About? The Math and Science Behind Them



An autoregressive LLM is a learned function that repeatedly estimates

$$P(\mathbf{x}_t \mid \mathbf{x}_1, \dots, \mathbf{x}_{t-1}).$$

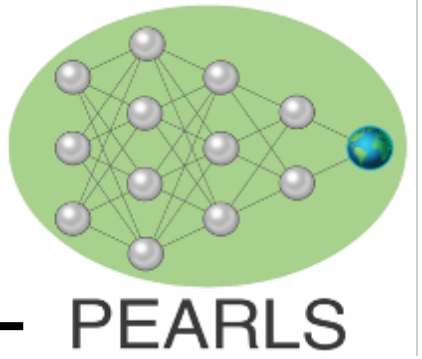
Its sophistication comes from the interaction of:

- ▶ a powerful **architecture**,
- ▶ enormous **training data**,
- ▶ large-scale **optimization**, and
- ▶ learned internal **representations**.

Important distinction

The Transformer equations do not themselves contain knowledge. A randomly initialized Transformer has the architecture, but not the useful behavior. Much of the capability is encoded in the trained parameters.

LLM Mathematical "toolkit"

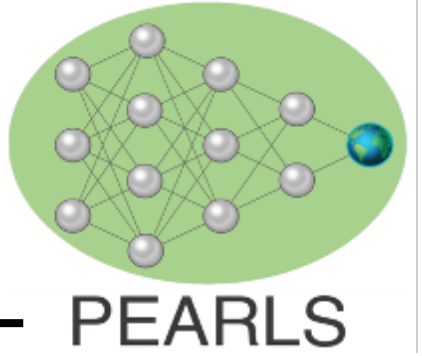


Linear algebra

Tokens and intermediate states are vectors. Learned transformations are dominated by matrix operations:

$$y = Wx + b.$$

LLM Mathematical "toolkit"



Linear algebra

Tokens and intermediate states are vectors. Learned transformations are dominated by matrix operations:

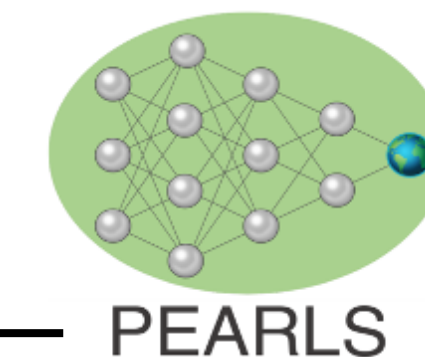
$$y = Wx + b.$$

Calculus + optimization

Derivatives tell us how changing parameters changes error:

$$\theta \leftarrow \theta - \eta \nabla_{\theta} L.$$

LLM Mathematical "toolkit"



Linear algebra

Tokens and intermediate states are vectors. Learned transformations are dominated by matrix operations:

$$y = Wx + b.$$

Calculus + optimization

Derivatives tell us how changing parameters changes error:

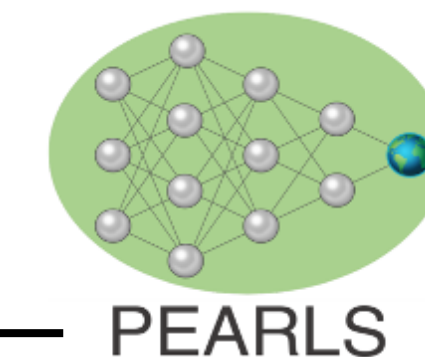
$$\theta \leftarrow \theta - \eta \nabla_{\theta} L.$$

Probability

The output is a probability distribution over possible next tokens:

$$P(x_t \mid x_{<t}).$$

LLM Mathematical "toolkit"



Linear algebra

Tokens and intermediate states are vectors. Learned transformations are dominated by matrix operations:

$$y = Wx + b.$$

Calculus + optimization

Derivatives tell us how changing parameters changes error:

$$\theta \leftarrow \theta - \eta \nabla_{\theta} L.$$

Probability

The output is a probability distribution over possible next tokens:

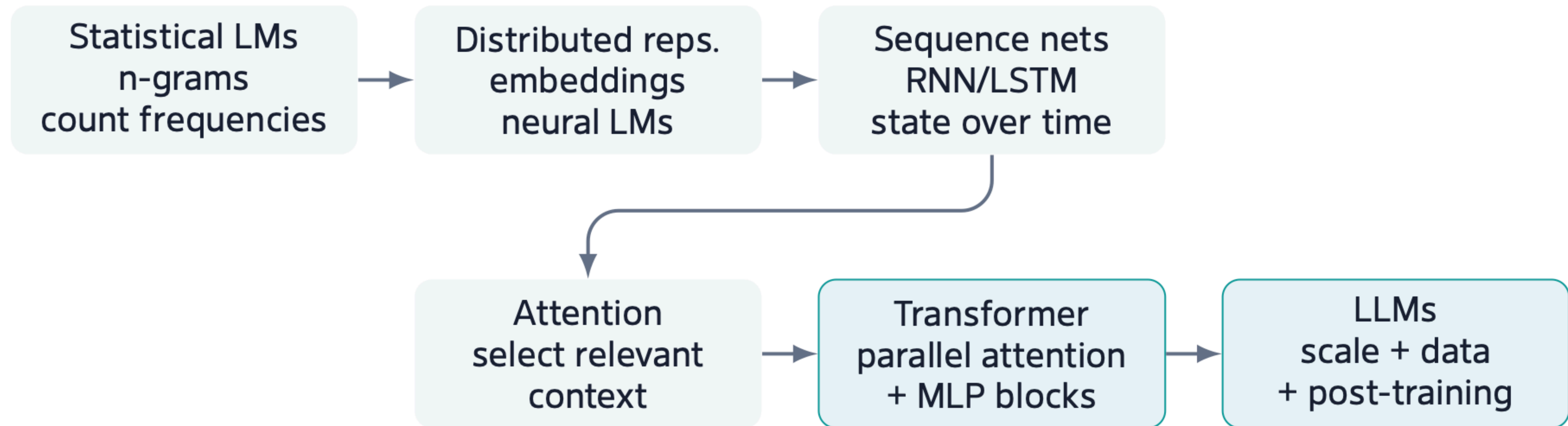
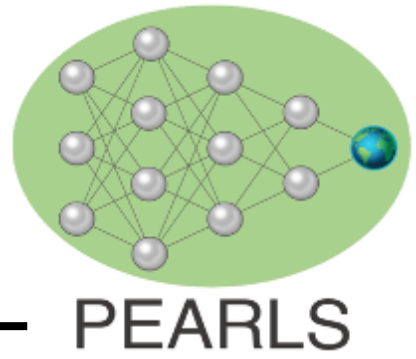
$$P(x_t \mid x_{<t}).$$

Information theory

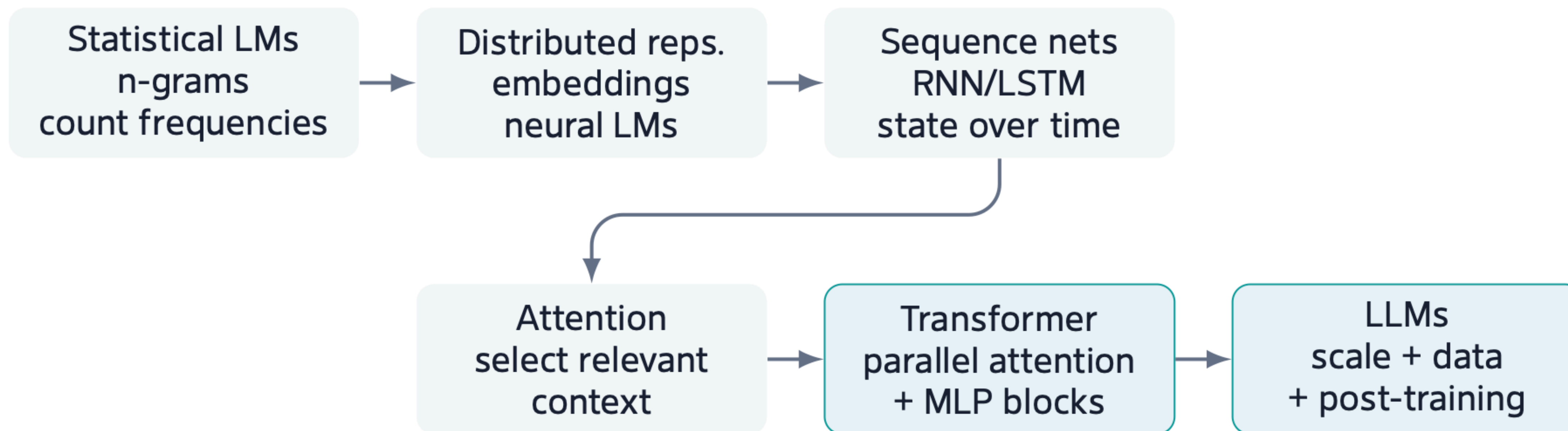
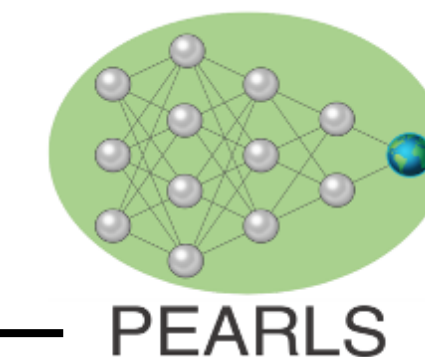
Surprisal and cross-entropy connect prediction, information, and compression:

$$I(x) = -\log p(x).$$

How did we get here? A very compressed history

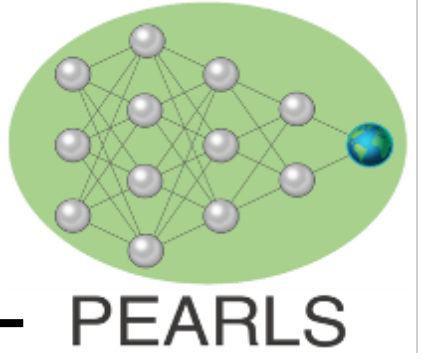


The 2017 inflection point



Vaswani et al.'s Attention Is All You Need introduced the Transformer architecture, making attention the central mechanism and enabling highly parallel training.

Embedding makes the connection from symbols to geometry



A tokenizer maps text to discrete token IDs. An embedding table maps each ID to a vector:

$$\text{token ID} \longrightarrow \mathbf{x} \in \mathbb{R}^d.$$

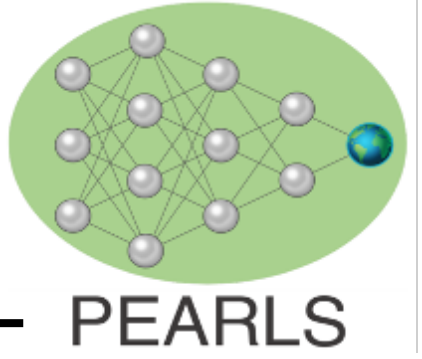
The representation then changes layer by layer as context is incorporated. "river bank" $\neq$

"investment bank"

Why this matters

Meaning is not stored as a dictionary entry. Useful linguistic and conceptual structure is represented in high-dimensional patterns distributed across activations and parameters.

Embedding makes the connection from symbols to geometry



A tokenizer maps text to discrete token IDs. An embedding table maps each ID to a vector:

$$\text{token ID} \longrightarrow \mathbf{x} \in \mathbb{R}^d.$$

The representation then changes

layer by layer as context is incorporated.

"river bank"

≠

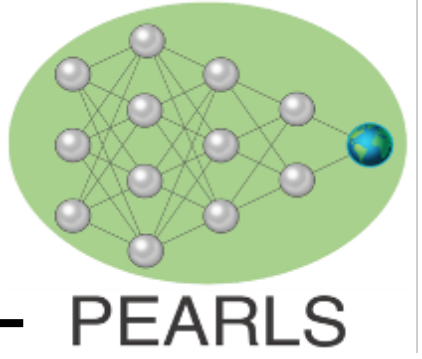
"investment bank"

Why this matters

Meaning is not stored as a dictionary entry. Useful linguistic and conceptual structure is represented in high-dimensional patterns distributed across activations and parameters.

An embedding is a starting representation; contextual representations inside later layers are richer and depend on the surrounding text.

Attention: which information should interact?



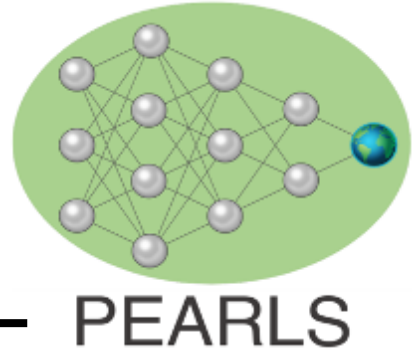
For each position, learned projections create queries, keys, and values:

$$Q = XW_Q, \quad K = XW_K, \quad V = XW_V.$$

Scaled dot-product attention is

$$\text{Attn}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V.$$

Attention: which information should interact?



Intuition

Query: What am I looking for?

Key: What kind of information do I contain?

Value: What information should I contribute if selected?

Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez* †
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukaszkaizer@google.com

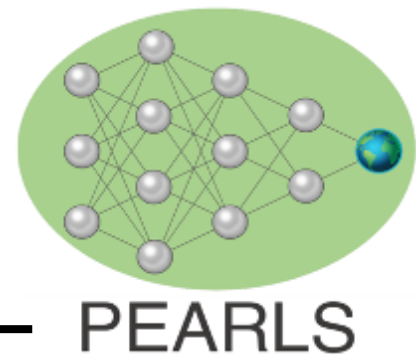
Illia Polosukhin* ‡
illia.polosukhin@gmail.com

Abstract

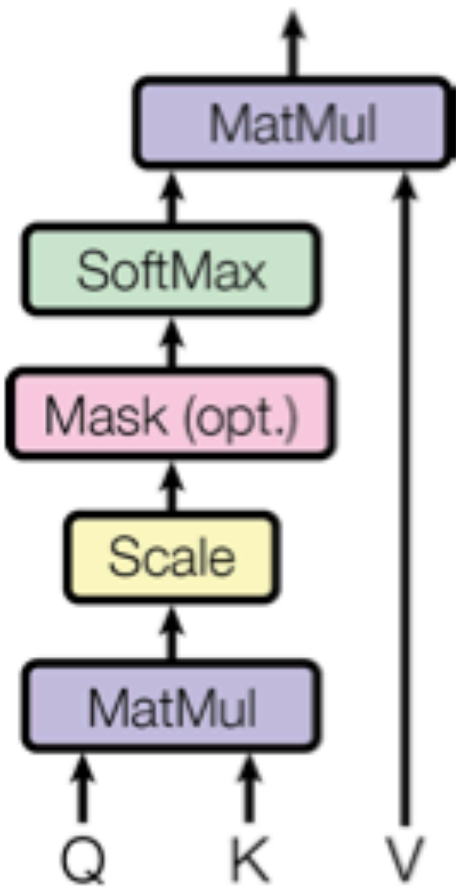
The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

A. Vaswani et al. (2017). Attention Is All You Need. NeurIPS. <https://arxiv.org/abs/1706.03762>

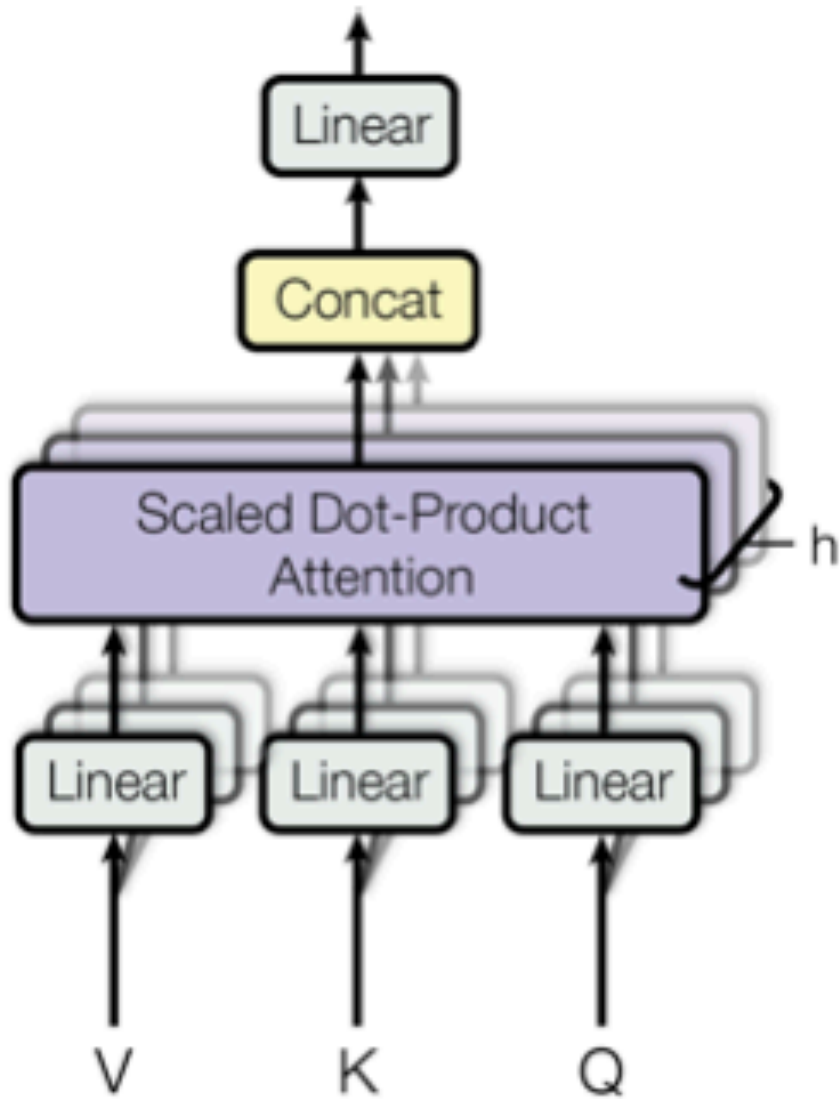
Scaled Dot-Product Attention and Multi-Head Attention



Scaled Dot-Product Attention



Multi-Head Attention



Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez*[†]
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukaszkaier@google.com

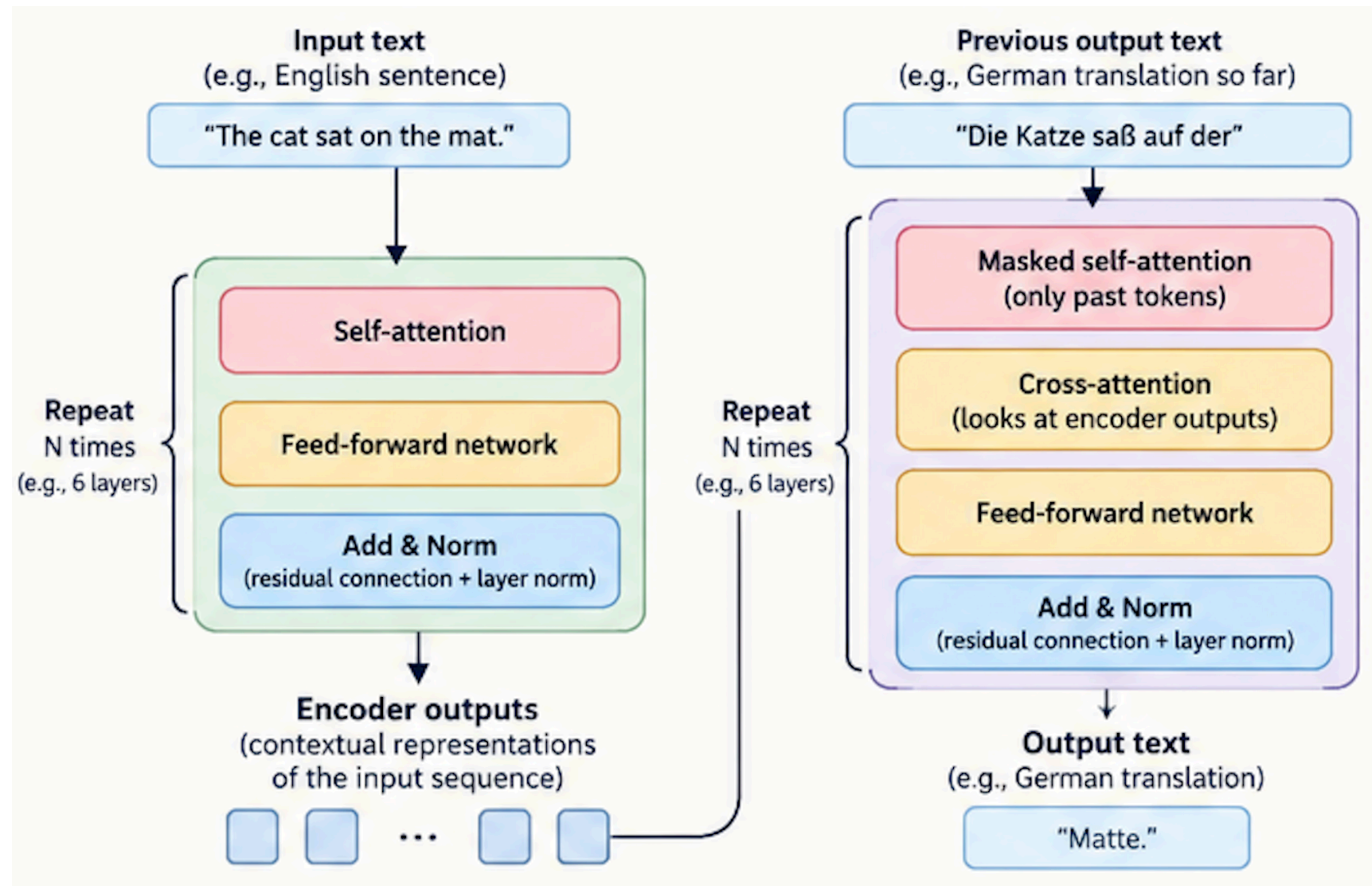
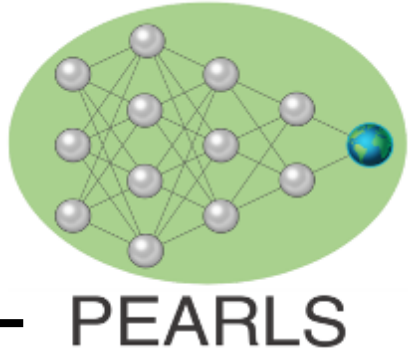
Illia Polosukhin*[‡]
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

Vaswani et al. (2017). Attention Is All You Need. NeurIPS. <https://arxiv.org/abs/1706.03762>

Multi-Head Attention (an example use case)



Attention Is All You Need

Ashish Vaswani*
Google Brain
avaswani@google.com

Noam Shazeer*
Google Brain
noam@google.com

Niki Parmar*
Google Research
nikip@google.com

Jakob Uszkoreit*
Google Research
usz@google.com

Llion Jones*
Google Research
llion@google.com

Aidan N. Gomez*[†]
University of Toronto
aidan@cs.toronto.edu

Lukasz Kaiser*
Google Brain
lukaszkaizer@google.com

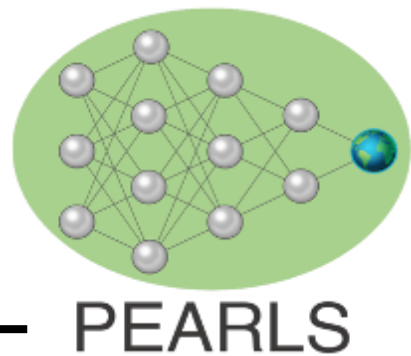
Illia Polosukhin*[‡]
illia.polosukhin@gmail.com

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. Our model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.

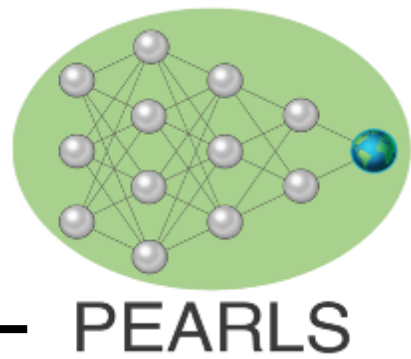
A. Vaswani et al. (2017). Attention Is All You Need. NeurIPS. <https://arxiv.org/abs/1706.03762>

From the Multi-Head Attention to today's Modern LLMs



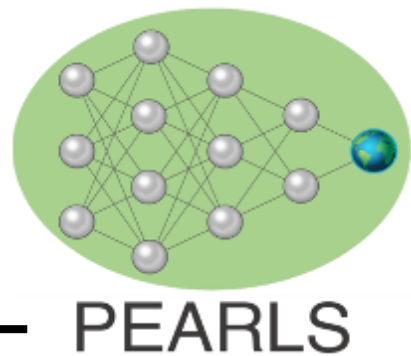
ASPECT	2017 TRANSFORMER	MODERN GPT-STYLE LLM
Architecture	Encoder + decoder	Decoder only
Main task	Seq. $\rightarrow$ seq.	Next-token prediction
Attention	Full / masked / cross	Causal self-attention
Input $\rightarrow$ out-put	Source $\rightarrow$ target	Context $\rightarrow$ next token
Generation	Conditioned on source	Token $\rightarrow$ token $\rightarrow$...
What changed?	General architecture	Scale + data + compute

From the Multi-Head Attention to today's Modern LLMs



ASPECT	2017 TRANSFORMER	MODERN GPT-STYLE LLM
Architecture	Encoder + decoder	Decoder only
Main task	Seq. $\rightarrow$ seq.	Next-token prediction
Attention	Full / masked / cross	Causal self-attention
Input $\rightarrow$ output	Source $\rightarrow$ target	Context $\rightarrow$ next token
Generation	Conditioned on source	Token $\rightarrow$ token $\rightarrow$...
What changed?	General architecture	Scale + data + compute

Possible next “journal club” papers on training and scale



Training Compute-Optimal Large Language Models

Jordan Hoffmann*, Sebastian Borgeaud*, Arthur Mensch*, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals and Laurent Sifre*

*Equal contributions

We investigate the optimal model size and number of tokens for training a transformer language model under a given compute budget. We find that current large language models are significantly under-trained, a consequence of the recent focus on scaling language models whilst keeping the amount of training data constant. By training over 400 language models ranging from 70 million to over 16 billion parameters on 5 to 500 billion tokens, we find that for compute-optimal training, the model size and the number of training tokens should be scaled equally: for every doubling of model size the number of training tokens should also be doubled. We test this hypothesis by training a predicted compute-optimal model, *Chinchilla*, that uses the same compute budget as *Gopher* but with 70B parameters and 4× more more data. *Chinchilla* uniformly and significantly outperforms *Gopher* (280B), GPT-3 (175B), Jurassic-1 (178B), and Megatron-Turing NLG (530B) on a large range of downstream evaluation tasks. This also means that *Chinchilla* uses substantially less compute for fine-tuning and inference, greatly facilitating downstream usage. As a highlight, *Chinchilla* reaches a state-of-the-art average accuracy of 67.5% on the MMLU benchmark, greater than a 7% improvement over *Gopher*.

J. Hoffmann et al. (2022). Training Compute-Optimal Large Language Models.
<https://arxiv.org/abs/2203.15556>

Scaling Laws for Neural Language Models

Jared Kaplan *
Johns Hopkins University, OpenAI
jaredk@jhu.edu

Sam McCandlish*
OpenAI
sam@openai.com

Tom Henighan
OpenAI
henighan@openai.com

Tom B. Brown
OpenAI
tom@openai.com

Benjamin Chess
OpenAI
bchess@openai.com

Rewon Child
OpenAI
rewon@openai.com

Scott Gray
OpenAI
scott@openai.com

Alec Radford
OpenAI
alec@openai.com

Jeffrey Wu
OpenAI
jeffwu@openai.com

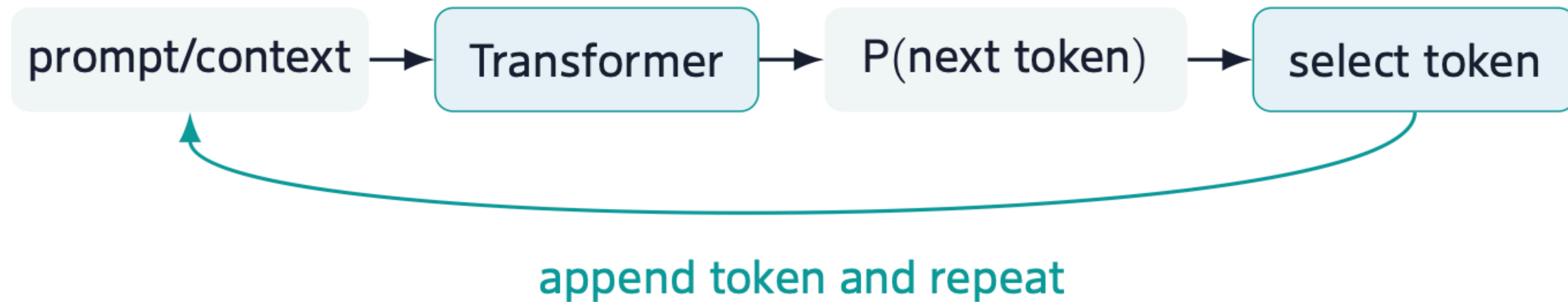
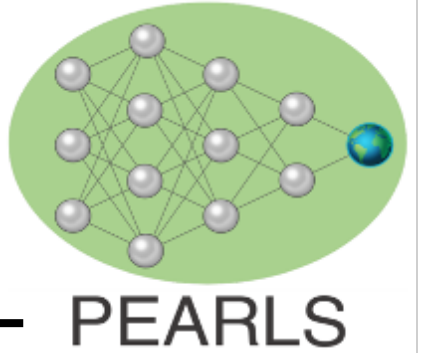
Dario Amodei
OpenAI
damodei@openai.com

Abstract

We study empirical scaling laws for language model performance on the cross-entropy loss. The loss scales as a power-law with model size, dataset size, and the amount of compute used for training, with some trends spanning more than seven orders of magnitude. Other architectural details such as network width or depth have minimal effects within a wide range. Simple equations govern the dependence of overfitting on model/dataset size and the dependence of training speed on model size. These relationships allow us to determine the optimal allocation of a fixed compute budget. Larger models are significantly more sample-efficient, such that optimally compute-efficient training involves training very large models on a relatively modest amount of data and stopping significantly before convergence.

J. Kaplan et al. (2020). Scaling Laws for Neural Language Models. <https://arxiv.org/abs/2001.08361>

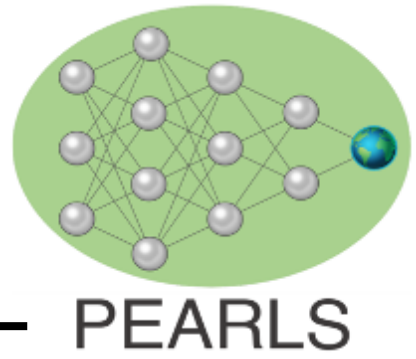
Inference: what happens when I ask a question?



Training vs. inference

During ordinary inference, the pretrained parameters are not being re-trained by your prompt. Your documents become context; they do not replace the model's learned parameters.

With out constraints (an example)



How many moon Jupiter has?

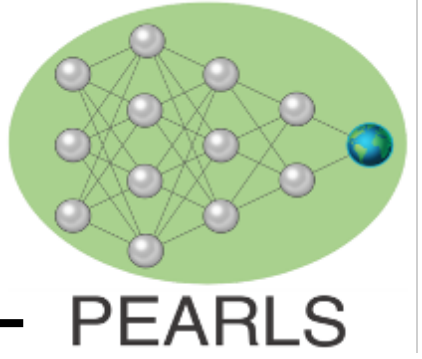
Jupiter has **95** officially recognized moons.



The four largest—Io, Europa, Ganymede, and **Callisto**—are called the **Galilean moons**, because Galileo Galilei discovered them in 1610.



With out constraints (an example)



How many moon Jupiter has?

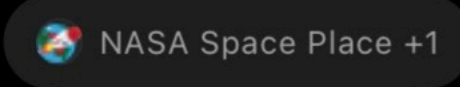
Jupiter has **95** officially recognized moons.



The four largest—Io, Europa, Ganymede, and **Callisto**—are called the **Galilean moons**, because Galileo Galilei discovered them in 1610.



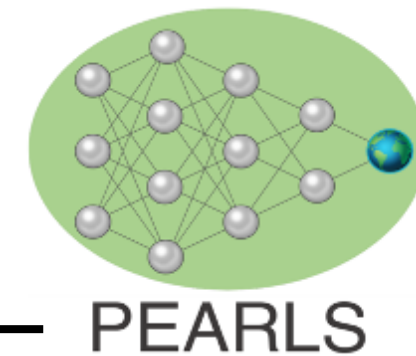
How many moon revolve around Jupiter?

Jupiter has **115** officially recognized moons as of 2026.  

The four largest and most famous are Io, Europa, Ganymede, and **Callisto**, called the **Galilean moons**. Ganymede is the **largest** moon in the entire **Solar System**.



With out constraints (an example)



How many moon Jupiter has?

Jupiter has **95 officially recognized moons**.



The four largest—Io, Europa, Ganymede, and **Callisto**—are called the **Galilean moons**, because Galileo Galilei discovered them in 1610.



How many moon revolve around Jupiter?

Jupiter has **115 officially recognized moons** as of 2026.

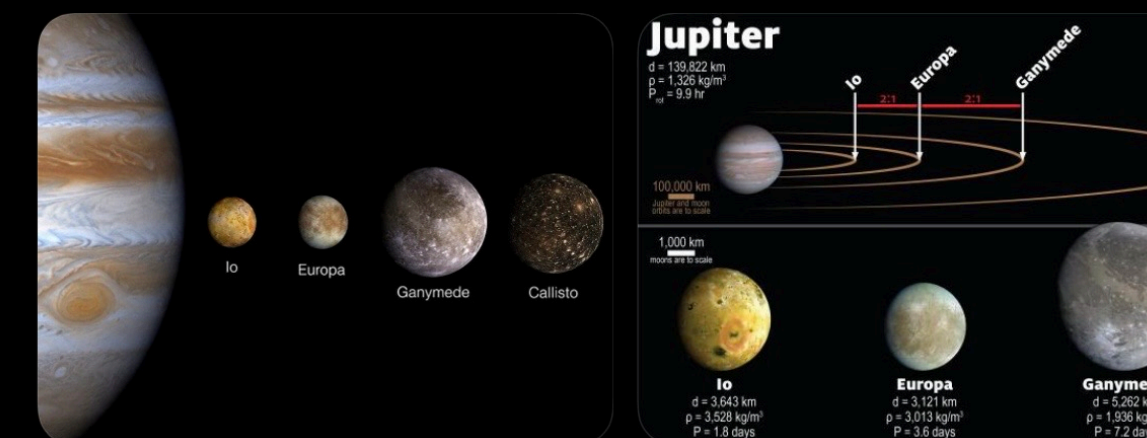
The four largest and most famous are Io, Europa, Ganymede, and Callisto, called the **Galilean moons**. Ganymede is the **largest moon in the entire Solar System**.



How many moon does Jupiter have?

Jupiter has **101 officially recognized moons** as of 2026.

The four biggest and most famous are the **Galilean moons**, discovered by Galileo in 1610:



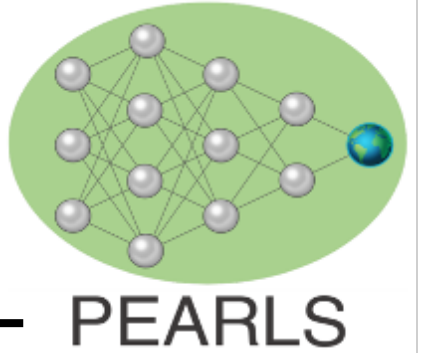
Io — covered in active volcanoes.

Europa — has an icy crust and probably a huge subsurface ocean.

Ganymede — the **largest moon in the Solar System**, even larger than Mercury.

Callisto — heavily cratered and one of the Solar System's oldest-looking surfaces.

A "good" requirement: constrain claims, not cognition



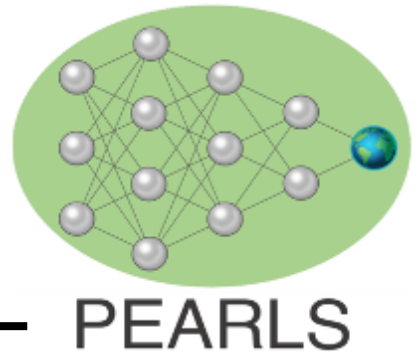
Instead of demanding that the model somehow forget everything else, require that every externally checkable claim be supported by authorized evidence.



Abstention rule

If authorized evidence does not support a claim, revise/remove it—or answer: "The provided sources do not contain enough information."

Open question: Where do new behaviors come from?



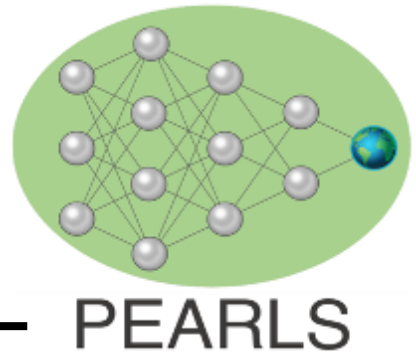
Add tools and controls when

- ▶ facts must be current or traceable,
- ▶ answers must stay within a corpus,
- ▶ calculations must be exact,
- ▶ decisions are high-stakes or auditable.

Scientific caution

"Emergence" can mean different things. Some capabilities appear sharply under particular benchmarks; later work showed that some apparent discontinuities can depend on the metric used.

Open question: Where do new behaviors come from?



Add tools and controls when

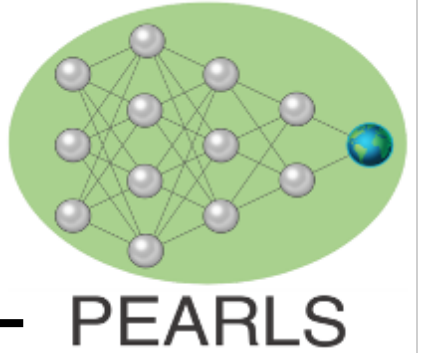
- ▶ facts must be current or traceable,
- ▶ answers must stay within a corpus,
- ▶ calculations must be exact,
- ▶ decisions are high-stakes or auditable.

Scientific caution

"Emergence" can mean different things. Some capabilities appear sharply under particular benchmarks; later work showed that some apparent discontinuities can depend on the metric used.

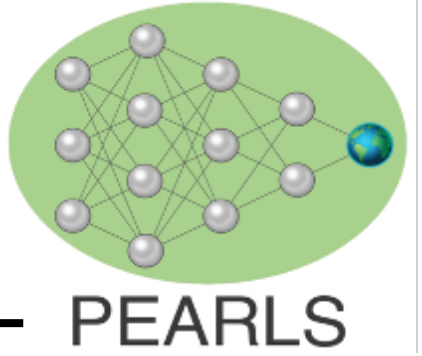
The forward computation is known exactly, but we do not yet have a complete scientific account of why all high-level capabilities arise from the learned parameters

Some practical strategies for grounded LLM systems



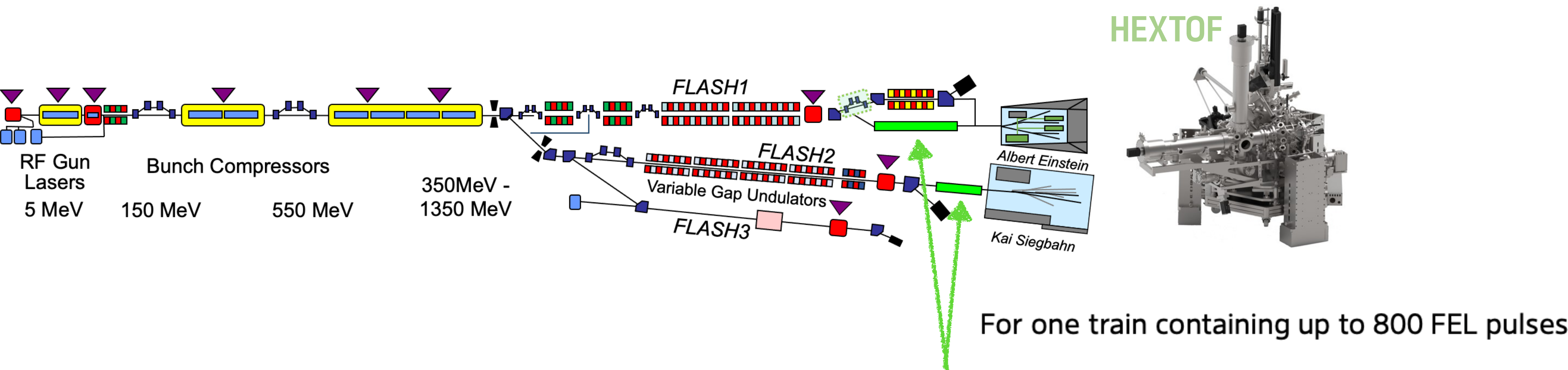
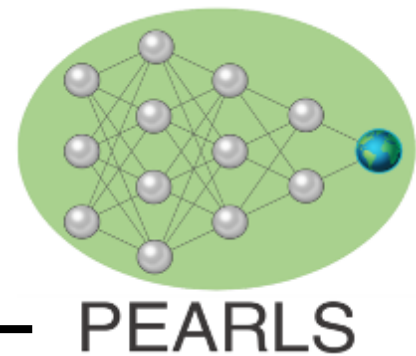
1. **Define the allowed evidence set.** Version it; control access.
2. **Retrieve narrowly.** Give the model the passages most relevant to the question.
3. **Ask for evidence-linked claims.** Citations should point to specific passages/pages.
4. **Permit derivation, not invention.** Calculations and logical consequences may be allowed when all premises come from the corpus.
5. **Verify.** Check citations, numbers, quotations, and whether evidence actually entails the claim.
6. **Abstain by design.** "Insufficient evidence" is a successful outcome, not a failure.

Questions and Outlook



- Final note: The capability comes from combining the architecture with enormous amounts of data and optimization that produces many learned internal representations.
- LLM journal club?
- How far are we from having an AI agent setting up the machine????
- How far are we from having an AI assistant in the control room? We may be surprised by how quickly the gap closes.
 - Emergence
 - Inference-time capabilities
 - Tool use and training

Reminder from Yesterday 's talk: Our Training Data



Magnet / RF	: $x(t)$	setpoint / readback			
BPM	: (x_b, y_b)	beam position	GMD	: $D[b, q]$	pulse-resolved parameters
Toroid / BCM	: Q_b	bunch charge	VLS	: $I[b, \lambda]$	pulse-resolved spectra
BAM	: t_b	arrival time	OPIS	: $V[b, c, t]$	multi-channel time traces
LLRF	: $V(t), \phi(t)$	waveforms	Ce:YAG	: $I(x, y)$	2D image, possibly integrated
BLM	: $I(t)$	waveform			
Screen	: $I(x, y)$	image			

$X_{\text{accelerator}}[n, b]$
 $E_{\text{GMD}}[n, b]$
 $T(t)$
 $\{\text{HEXTOF events}\}_{n,b}$

Reminder: DESY Computing and Controls setup

