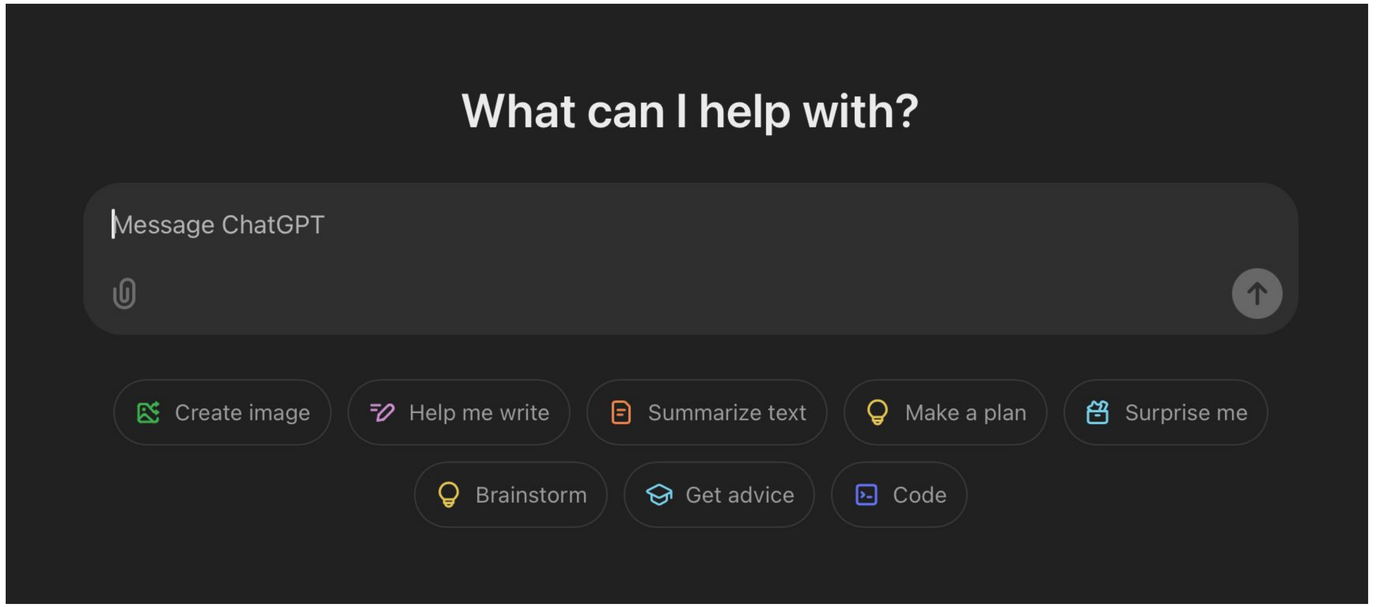
Optimization for Large Language Models (LLMs)

Guest Lecture: Optimization for Machine Learning Course

Alejandro Hernández Cano & Alexander Hägele

additional materials by Y. Dubois, L. v. Werra, Jason Wei.

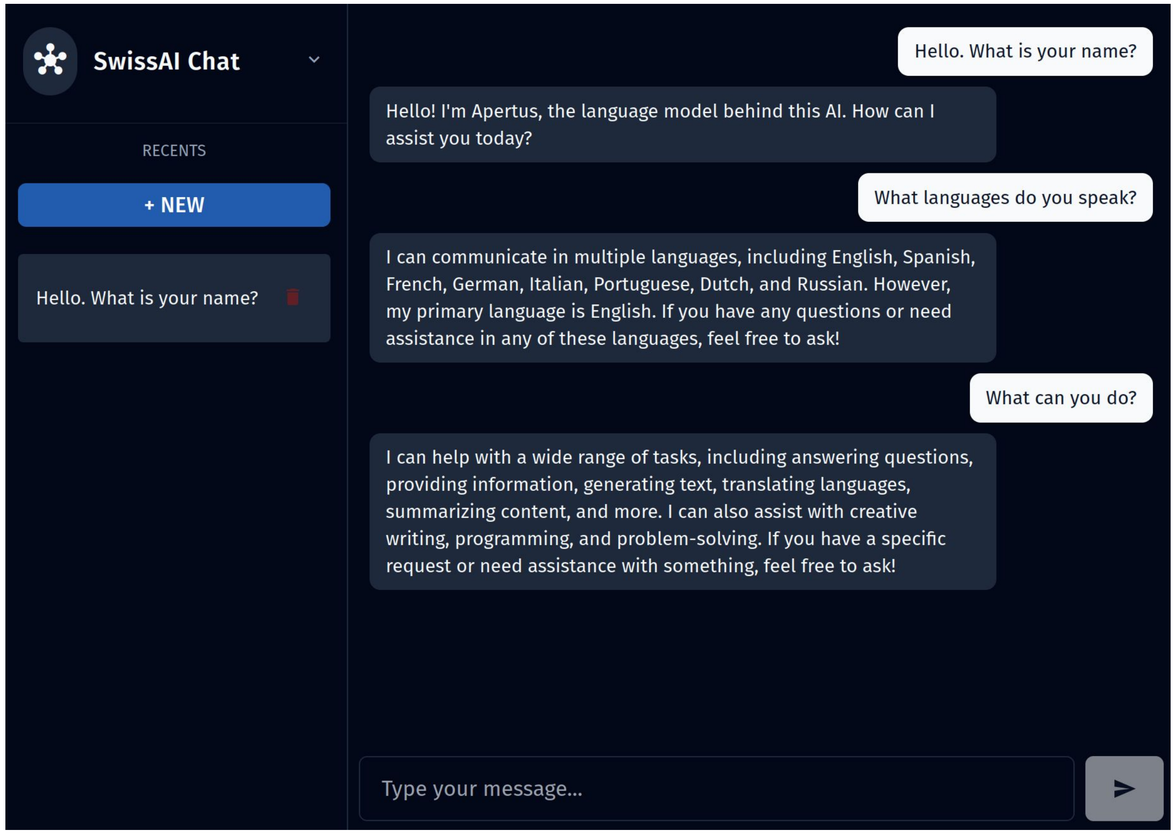
LLMs are Everywhere!



Goal: Understand what it takes to build a large language model

… and show how we are doing it with SwissAI!

The Swiss AI Initiative Model



Early access preview: https://chat.swissai.cscs.ch/

Outline

Part 1 **Building Blocks**

- Transformers
- Language Modeling
- Pretraining vs. Posttraining

Part 2 **Pretraining**

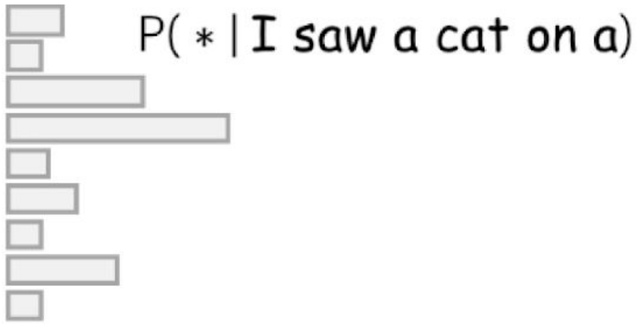
- Data
- Distributed Training
- *Optimization*

Part 3 **Real World Showcase: SwissAI LLM**

- Alps Cluster
- Data
- Optimization

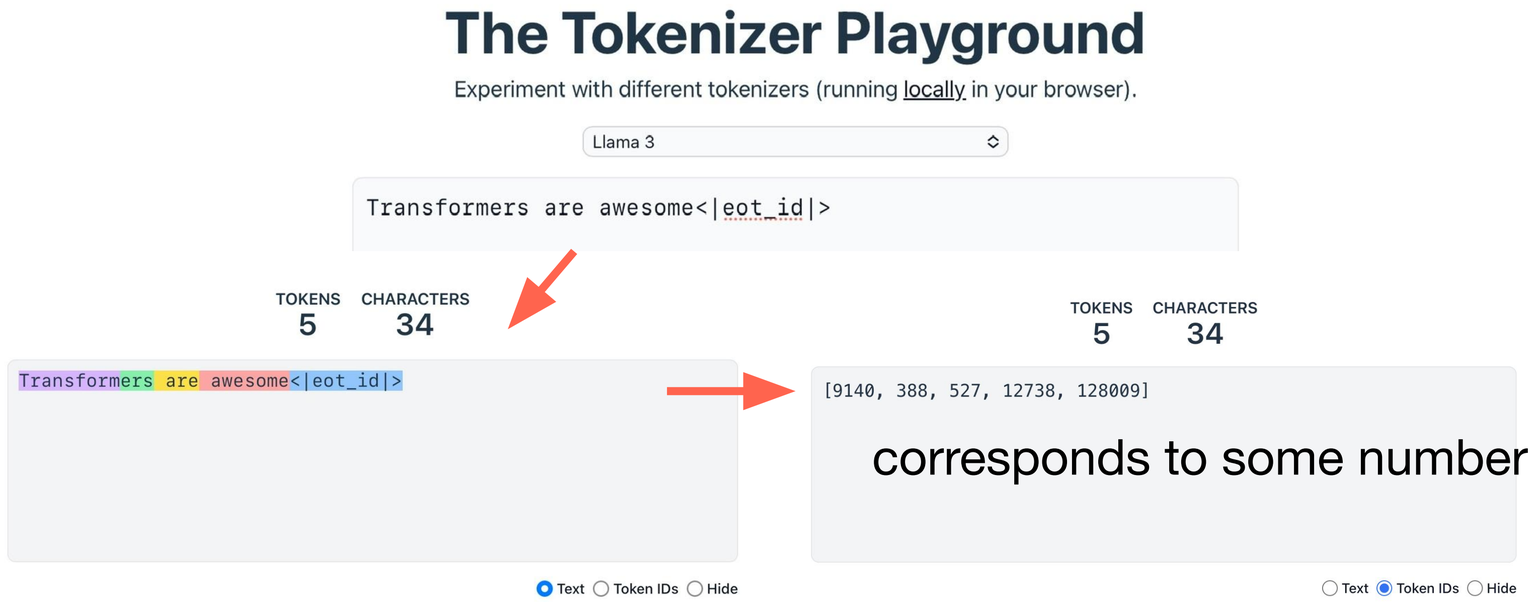
Language Modelling

- **Goal:** Describe distributions over sequences of text.
- **Input:** A sentence, “I saw a cat on a mat”.
- **Optimize to:** Learn next-word distribution.



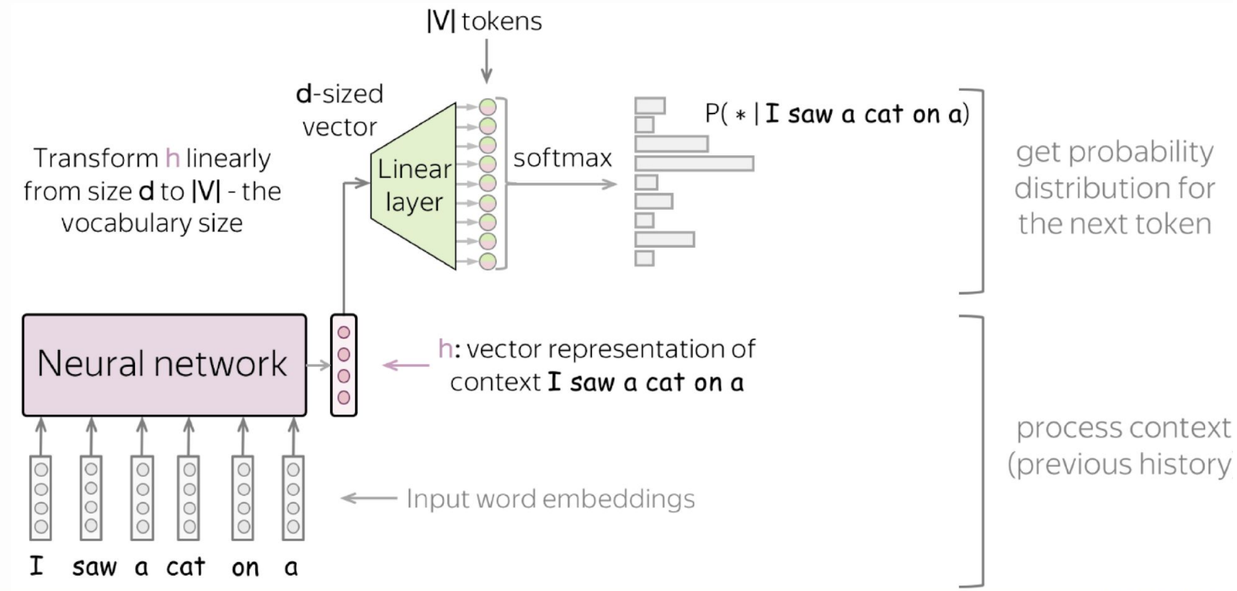
Tokenization

- Splits the input text into a sequence of input tokens (typically word fragments + some special symbols) according to some predefined tokenization algorithm:



Embeddings

- Each token *i* will be mapped to an embedding parameter vector w_i.
- Each sentence *"I saw a cat on a"* → *[50, 21, 43, 9]* → $[w_{50}, w_{21}, w_{43}, w_9]$ is transformed to a sequence of vectors.
- The language model will further refine these vector representations to predict next-token probabilities.

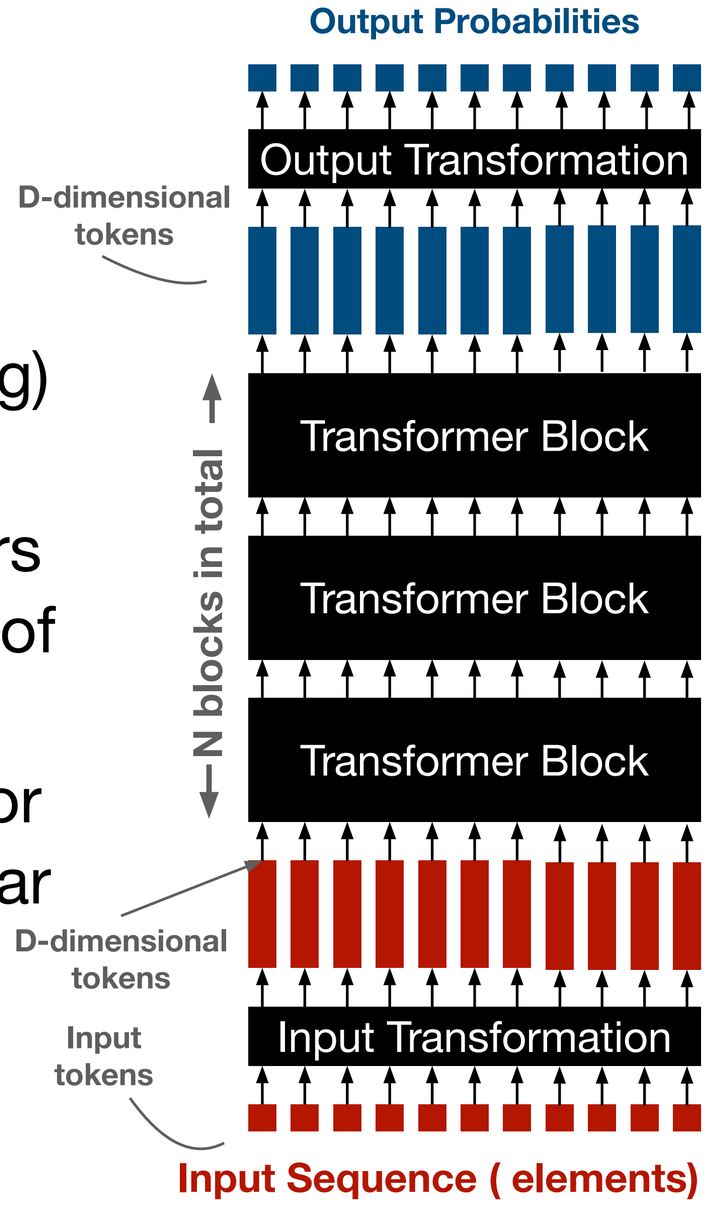


Transformers as Language Models

Input: Sequence of tokens (sentence).

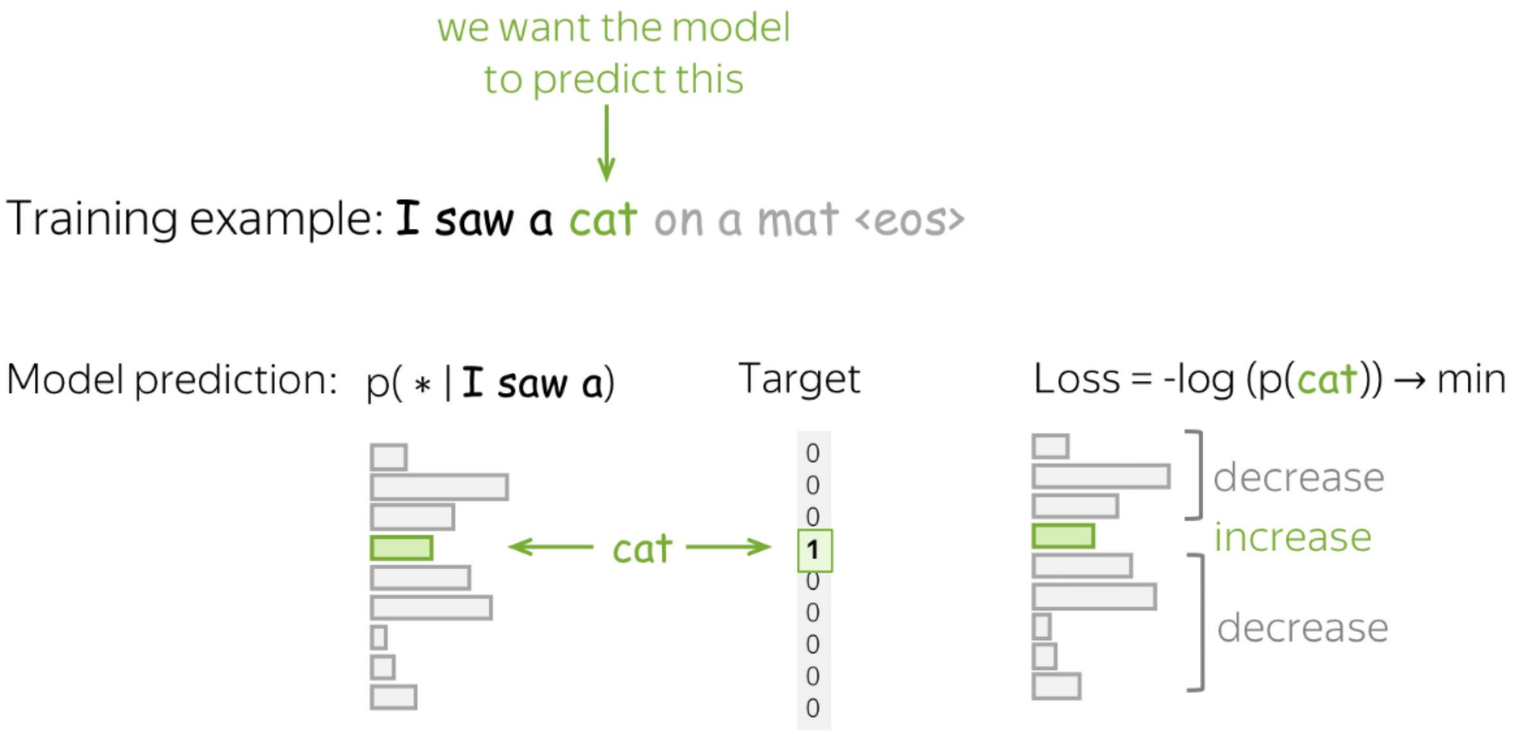
Output: Next-token probabilities.

- **Input Transformation:** Obtains real-valued (embedding) vectors.
- **Transformer block:** Transforms a sequence of vectors of dimension into a new sequence of vectors of dimension using **self-attention** and **MLP sub-blocks**
- **Output transformation**: Converts the learnt vector representation to next-token probabilities (with a linear projection).



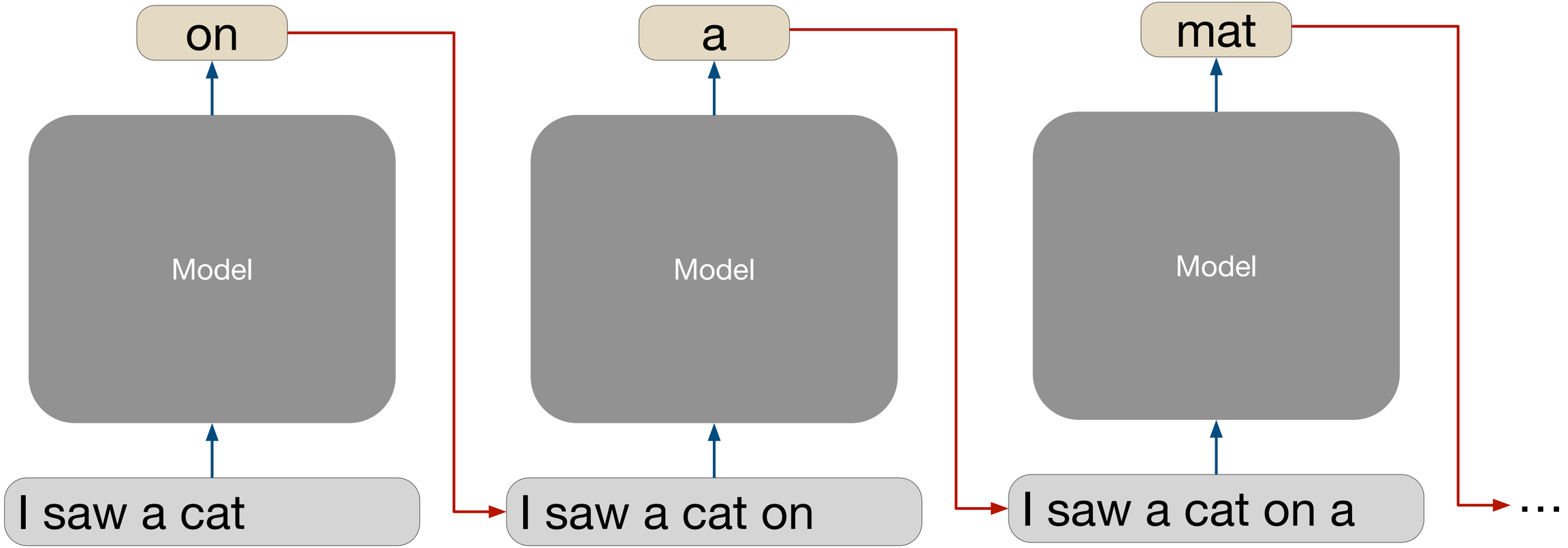
Optimizing Transformers

- Cross entropy loss and a gradient-based method.



Text generation

- Via iterative autoregressive inference: sample next-token probabilities and using the generated tokens as the input of next iterations.



Pretraining vs Posttraining

	Pretraining	Posttraining
Data Quantity	Massive, ~the internet (Bilions-trillions of words)	Small, millions of examples
Data Quality	Low(er)	High
Data Source	Webcrawls, Papers, Textbooks, Github, …	(Often) Human
Goal	General Learning, Knowledge	Make model usable, give specific skills + character, alignment, …

Outline

Part 1 **Building Blocks**

- Transformers
- Language Modeling
- Pretraining vs. Posttraining

Part 2 **Pretraining**

- Data
- Distributed Training
- *Optimization*

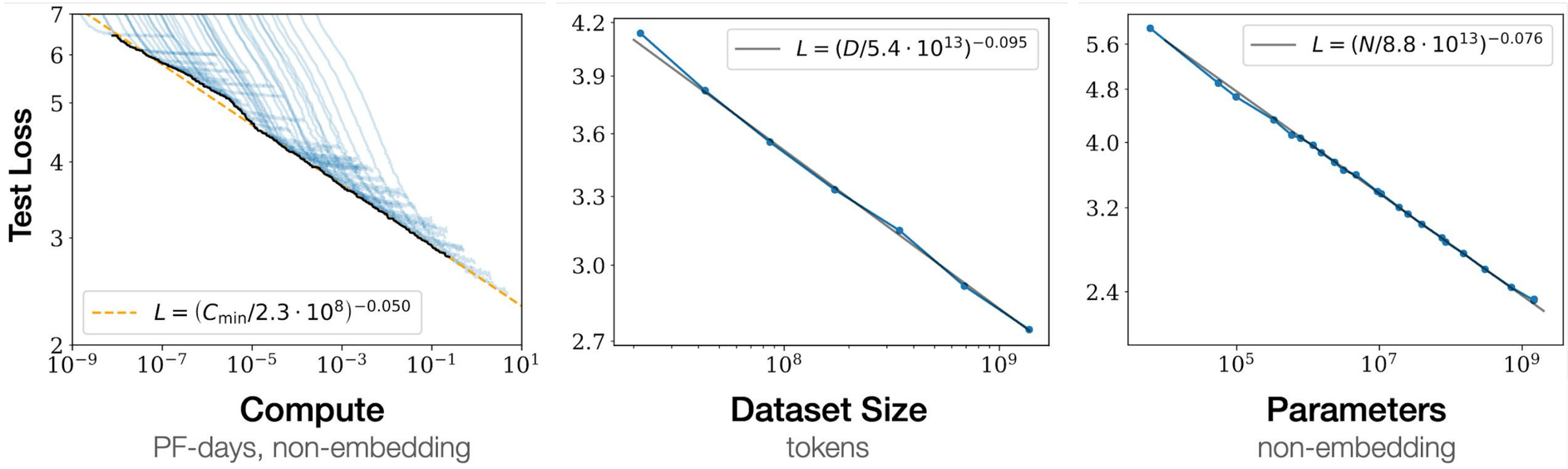
Part 3 **Real World Showcase: SwissAI LLM**

- Alps Cluster
- Data
- Optimization

Intuitions For LLMs: Scaling Laws

Intuition: The Power of **Scale**

Compute = Model Size * Data



Kaplan et al., 2020. "Scaling laws for neural language models."

"*Language modeling performance (loss) improves smoothly as we increase the model size, dataset size, and amount of compute for training.*" = **Scaling Laws**

Intuitions For LLMs: Scaling Laws

Intuition: The Power of **Scale**

Compute = Model Size * Data

Challenges:

1. **Scaling Data**
2. **Scaling Model**
3. **Efficient Optimization**
   a. **Both computationally and algorithmically!**

Challenge 1: Data

the secret sauce

- **Goal of Pretraining:** General purpose model
  - Train on massive quantities of text
    - Latest Qwen3: 36T (***trillion***) tokens

- **Challenges**
  - Maximize coverage, diversity
  - Maximize quality, correctness
  - Find right measures of data quality
  - Efficient processing of trillions of tokens (multiple TB of data)

Challenge 1: Data

the secret sauce

- **Goal of Pretraining:** General purpose model
  - Train on massive quantities of text
- **Where to find data**
  - **Common Crawl:** starting point
  - **Code:** Github etc.
  - **Curated:**
    - Websites
    - Books
- **(Synthetic Data: new trend, utilizing existing models)**

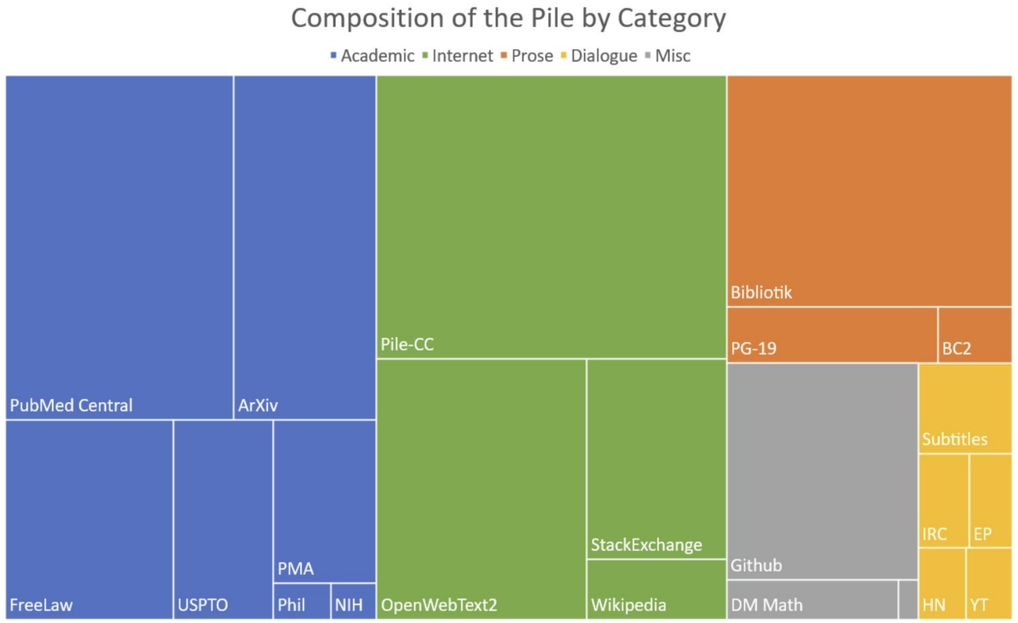


Figure 1: Treemap of Pile components by effective size.

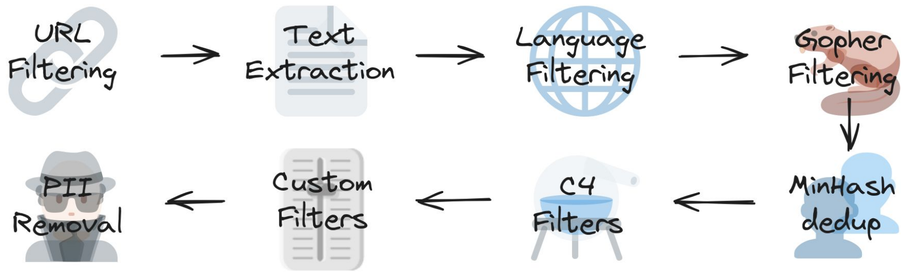
Gao et al., 2021. "The Pile: An 800GB Dataset of Diverse Text for Language Modeling."

Challenge 1: Data

the secret sauce

Example: FineWeb

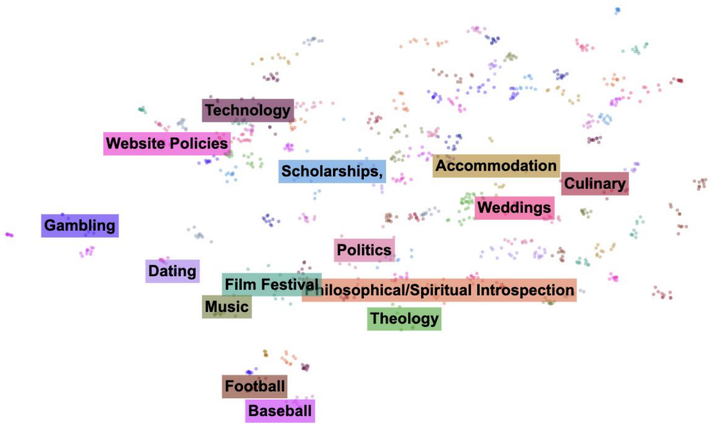
- Based on CommonCrawl
- 15T tokens
- 44TB disk space
- Heavy filtering
- Transparent processing pipeline



The FineWeb pipeline

HuggingFaceFW / blogpost-fineweb-v1 538

🍷 FineWeb: decanting the web for the finest text data at scale



The 🍷 FineWeb dataset, clustered and annotated with educational score labels

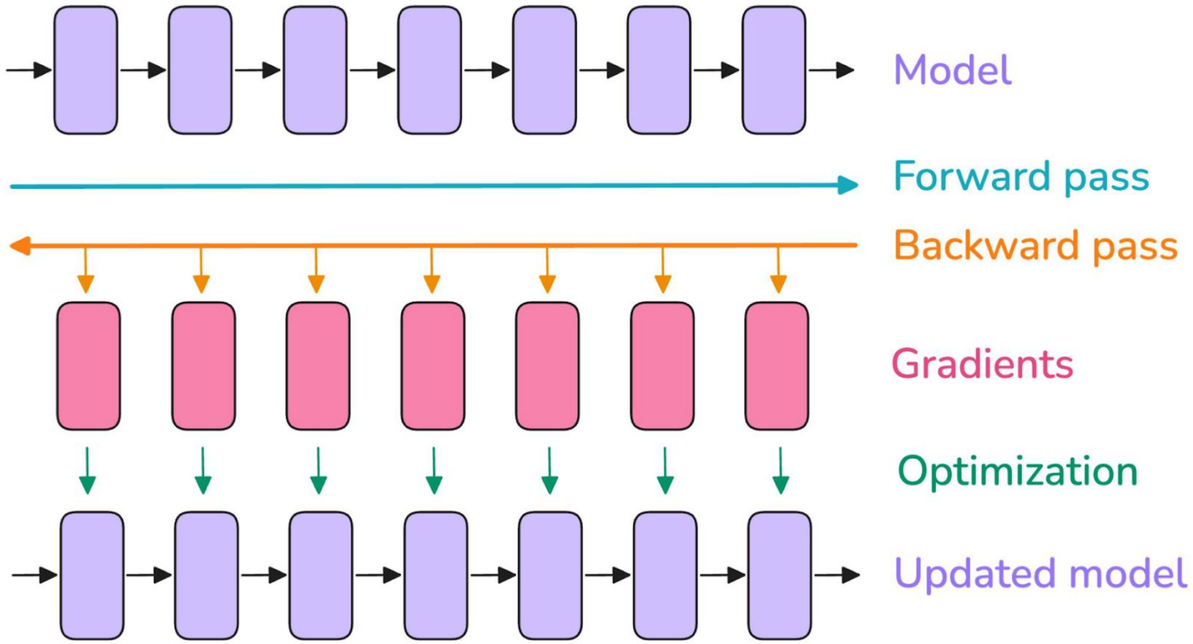
AUTHORS	AFFILIATION	PUBLISHED
Guilherme Penedo, Hynek Kydlíček, Loubna Ben Allal, Anton Lozhkov, Colin Raffel, Leandro Werra, Thomas Wolf	HuggingFace	May 31, 2024

https://huggingface.co/spaces/HuggingFaceFW/blogpost-fineweb-v1

Challenge 2: Distributed Training

How to scale

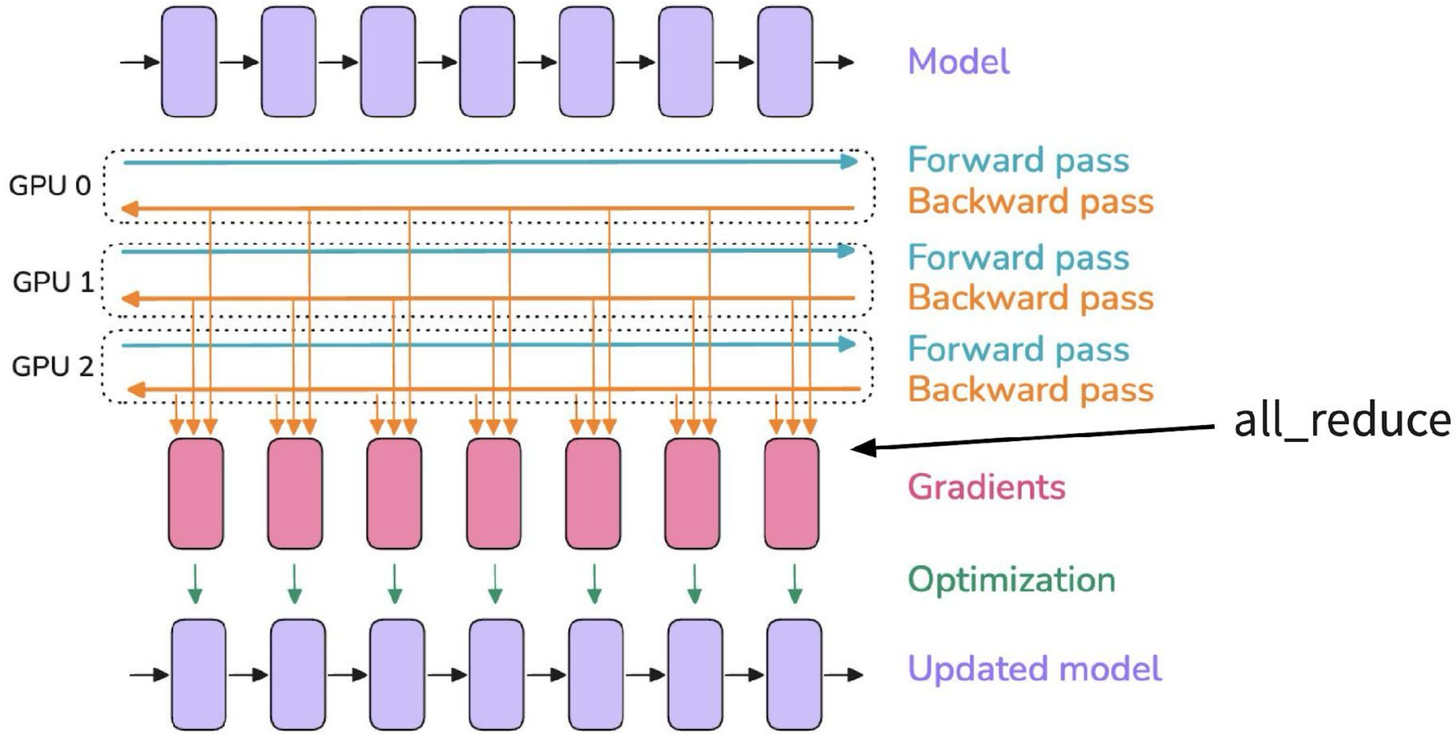
- **Argument 1:** Large models require multiple GPUs
- **Argument 2:** We want to process a huge amount of data (large batch size)
- Recall basic optimization of neural networks:



Challenge 2: Distributed Training

How to scale

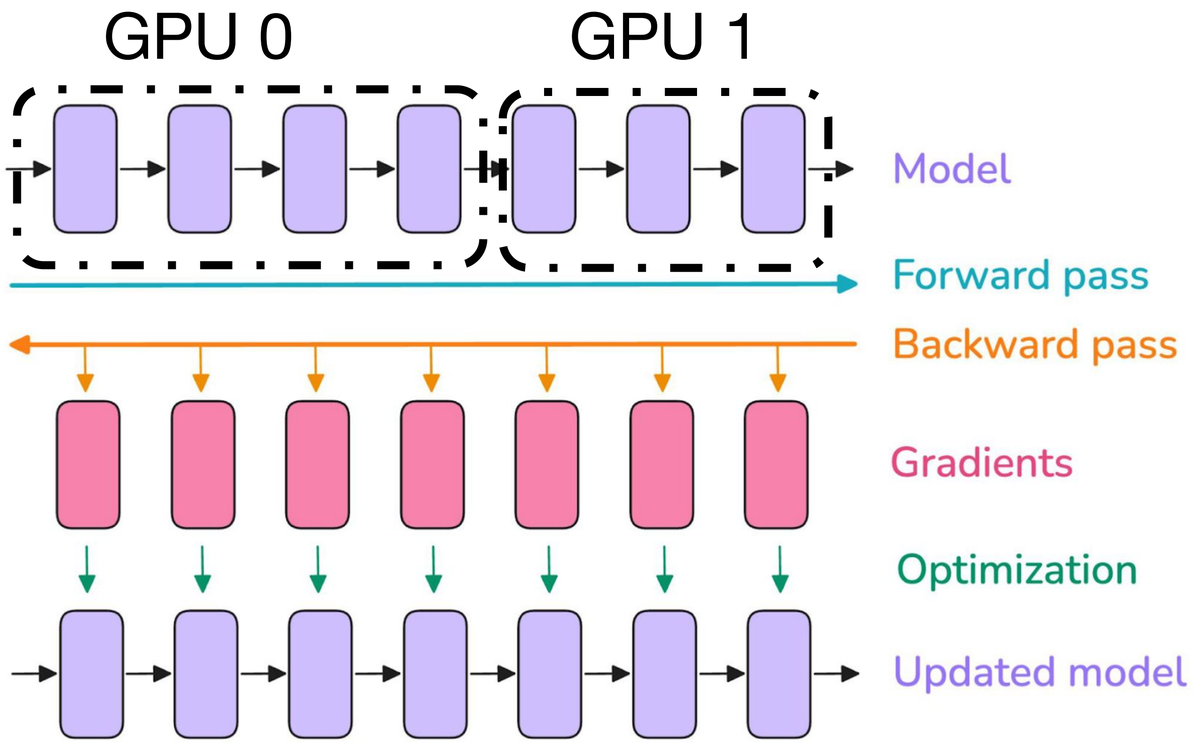
- **1: Data Parallelism:** Distribute batches (*micro batches)* across GPUs



Challenge 2: Distributed Training

How to scale

- **2: Pipeline Parallelism:** Distribute layers across GPUs



Challenge 2: Distributed Training

How to scale

- **3D Parallelism**
  - **1: Data Parallelism:** distribute batches (*micro batches)* across GPUs
  - **2: Pipeline Parallelism:** distribute layers across GPUs
  - **3: Tensor Parallelism:** split matrices across GPUs (not shown here)
- **4D (new)**:
  - **4:** Sequence Parallelism: split tokens across GPUs (not shown here)

Challenge 3: Efficient Optimization

Say you have a dataset D and your transformer model M. How do you train?

Standard Recipe

	Parameter
Optimizer	AdamW (w/ Weight Decay)
Momentum	$(\beta_1,\beta_2)=(0.9, 0.99)$
Weight Decay	0.1
Sequence Length	e.g. 4096 Tokens
Batch Size	Large! Mio. of Tokens, e.g. 4-16M
LR Schedule	Warmup + Annealing to Low Value
Peak LR	Depends on Model. Typical Value: 1e-3
Gradient Clipping	Norm 1.0

Challenge 3: Efficient Optimization

AdamW

Consider gradient $\boldsymbol{g} = \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\boldsymbol{\theta}}(\boldsymbol{x})$

First Momentum

$$\boldsymbol{m}^{(t)} = \beta_1 \boldsymbol{m}^{(t-1)} + (1-\beta_1)\boldsymbol{g}^{(t)}, \quad \hat{\boldsymbol{m}}^{(t)} = \frac{\boldsymbol{m}^{(t)}}{1-\beta_1^t}$$

Second Momentum

$$\boldsymbol{\nu}^{(t)} = \beta_2 \boldsymbol{\nu}^{(t-1)} + (1-\beta_2)\boldsymbol{g}^{(t)^2}, \quad \hat{\boldsymbol{\nu}}^{(t)} = \frac{\boldsymbol{\nu}^{(t)}}{1-\beta_2^t}$$

$$\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)} - \eta^{(t)} \left(\frac{\hat{\boldsymbol{m}}^{(t)}}{\sqrt{\hat{\boldsymbol{\nu}}^{(t)}} + \epsilon} + \lambda \boldsymbol{\theta}^{(t-1)} \right)$$

Learning Rate *Weight Decay*

Stores 3x model parameters: $\boldsymbol{\theta}^{(t)}, \boldsymbol{m}^{(t)}, \boldsymbol{\nu}^{(t)}$

Challenge 3: Efficient Optimization

How to scale

- **Ch** **e 3x the mod**
  - ical size), pa
- **Alternative:** ZeRO distributed optimizer.

Needs the full states **m**, **v**

Only needs a fraction $\mathbf{m}_i$, $\mathbf{v}_i$

Standard framework for *i*-th worker:

1. Compute $\nabla \mathbf{w}\, L(B_i)$.
2. Aggregate $\nabla \mathbf{w}\, L = \sum_j \nabla \mathbf{w}\, L(B_j)$.
3. Update $\mathbf{w}$ using $\nabla \mathbf{w}\, L$.

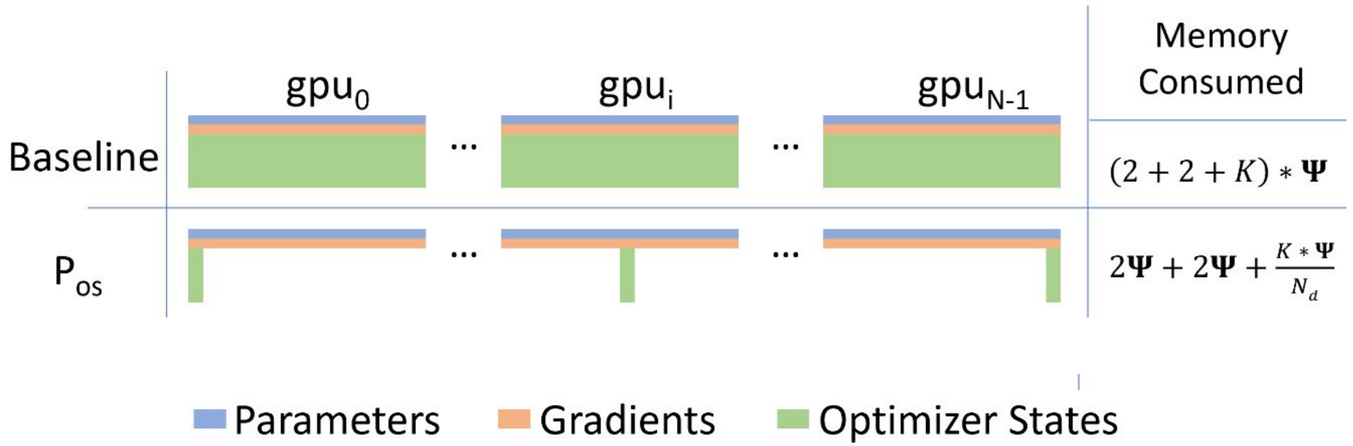
Alternative: *i*-th worker only updates a fraction of parameters $\mathbf{w}_i$:

1. Compute $\nabla \mathbf{w}\, L(B_i)$
2. Aggregate only $\nabla \mathbf{w}_i\, L = \sum_j \nabla \mathbf{w}_i\, L(B_j)$.
3. Update $\mathbf{w}_i$ using $\nabla \mathbf{w}_i\, L$.
4. Communicate updated $\mathbf{w}_i$ to the other workers.

Challenge 3: Efficient Optimization

How to scale

- **Challenge: Adam needs to store 3x the model parameters**
  - Consider: For a 70B model (typical size), parameters are ~1
- **Zero1: distributed optimizer shards optimizer states.**

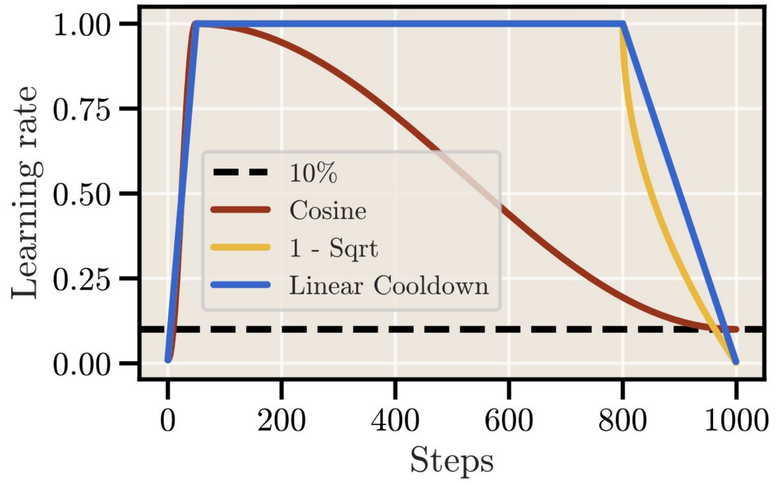


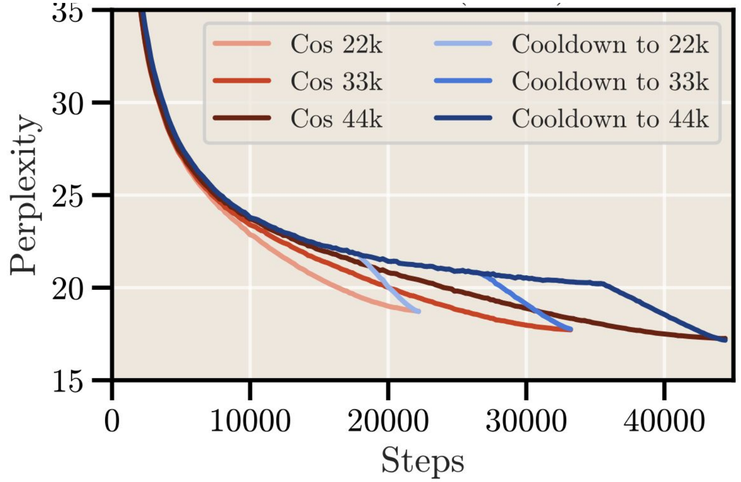
- Optimizer updates are only done with local m and v, then broadcast

Source: Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, Yuxiong He, ZeRO: Memory Optimizations Toward Training Trillion Parameter Models

Challenge 3: Efficient Optimization

LR Schedules





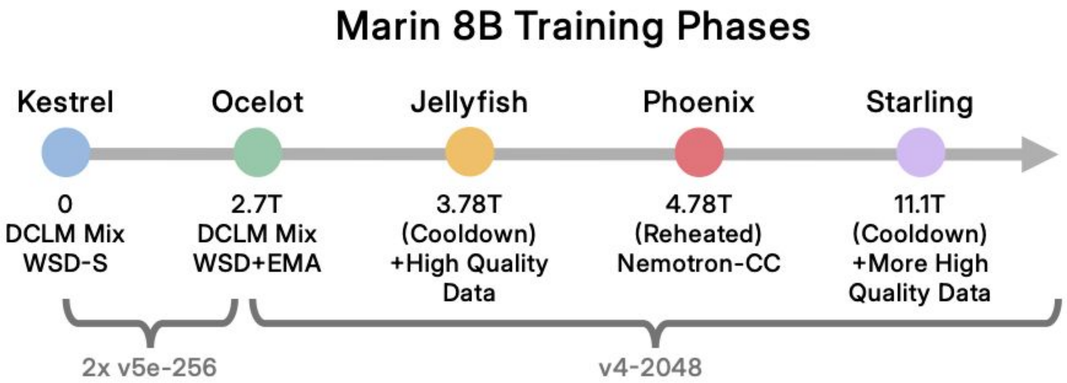
- Warmup for stability
- Annealing/cooldown to reach best loss
  - Different schedules: Cosine, Linear, WSD
- Rewarming leads to loss increase
- Important: big runs are only done once!

Hägele, A., Bakouch, E., Kosson, A., Von Werra, L., & Jaggi, M. (2024). Scaling laws and compute-optimal training beyond fixed training durations. *Advances in Neural Information Processing Systems*, *37*, 76232-76264.

Challenge 3: Efficient Optimization

LR Schedules

- Important: big runs are only done once!
- Schedule naturally separates training into phases (new "*Midtraining*")
- E.g., mix in higher quality data towards the end



https://marin.readthedocs.io/en/latest/reports/marin-8b-retro/

Challenge 3: Efficient Optimization

Optimization Algorithms

Pagliardini, M., Ablin, P., & Grangier, D. (2025). The AdEMAMix Optimizer: Better, Faster, Older. ICLR 2025

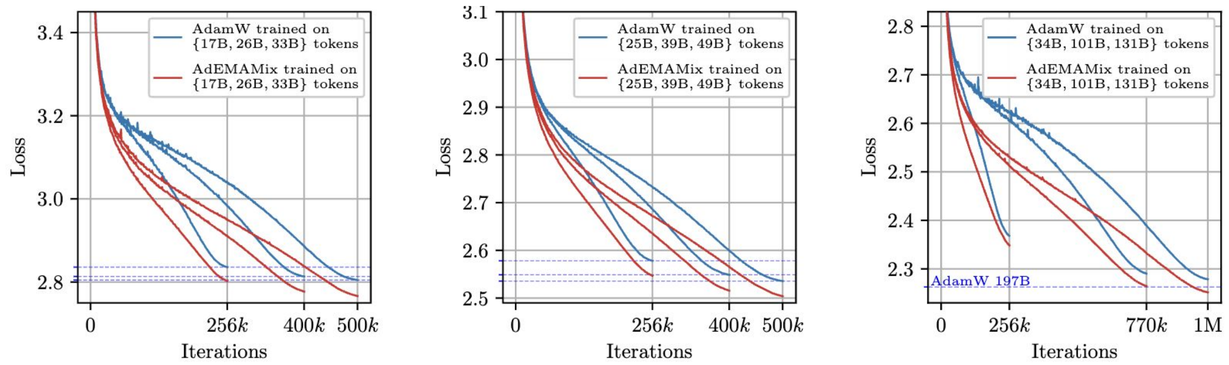
- We are far from done with optimization algorithms!

THE ADEMAMIX OPTIMIZER:
BETTER, FASTER, OLDER

Matteo Pagliardini †
EPFL

Pierre Ablin
Apple

David Grangier
Apple



(a) 110M parameters. (b) 330M parameters. (c) 1.3B parameters.

Figure 1: **Comparing AdamW and AdEMAMix on language modeling.** In **(a,b,c)**, we plot the loss obtained using AdamW and AdEMAMix (our optimizer) to train Transformer models of various sizes on the Redpajama dataset. In **(a)**, we train multiple baselines for $256k$, $400k$, and $500k$

Also see 2nd order optimizers: Shampoo, SOAP, Muon, …

Recap: Pretraining

- Importance of Data
- Distributed Training
  - 4D Parallelism
- Optimization
  - AdamW, Distributed Optimizers
  - LR Schedules and Training Phases
  - AdEMAMix

Outline

Part 1 **Building Blocks**

- Transformers
- Language Modeling
- Pretraining vs. Posttraining

Part 2 **Pretraining**

- Data
- Distributed Training
- *Optimization*

Part 3 **Real World Showcase: SwissAI LLM**

- Alps Cluster
- Data
- Optimization

Swiss AI Initiative: Multilingual LLMs

Data | Modeling | Evaluation

Leveraging the world's most AI-capable supercomputer to build **fully open multilingual LLMs** and **innovate across the full stack**

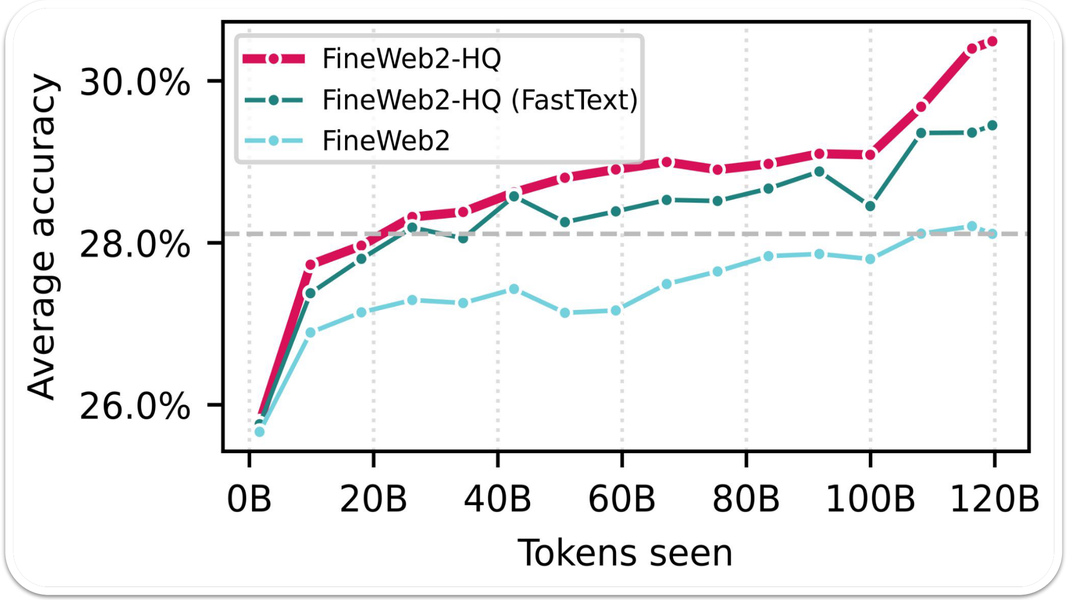
Partial Slide Courtesy: Vinko Sabolčec, Alexander Hägele, Angelika Romanou

Breaking Barriers with Multilingual Data

Multilingual | Open | Fair

We use multilingual datasets covering almost **2000 languages and scripts** and **comply with training opt-outs**.

Data Curation at Scale



We build upon recent trends in data curation and **push English and multilingual performance further** using model-based filtering methods.

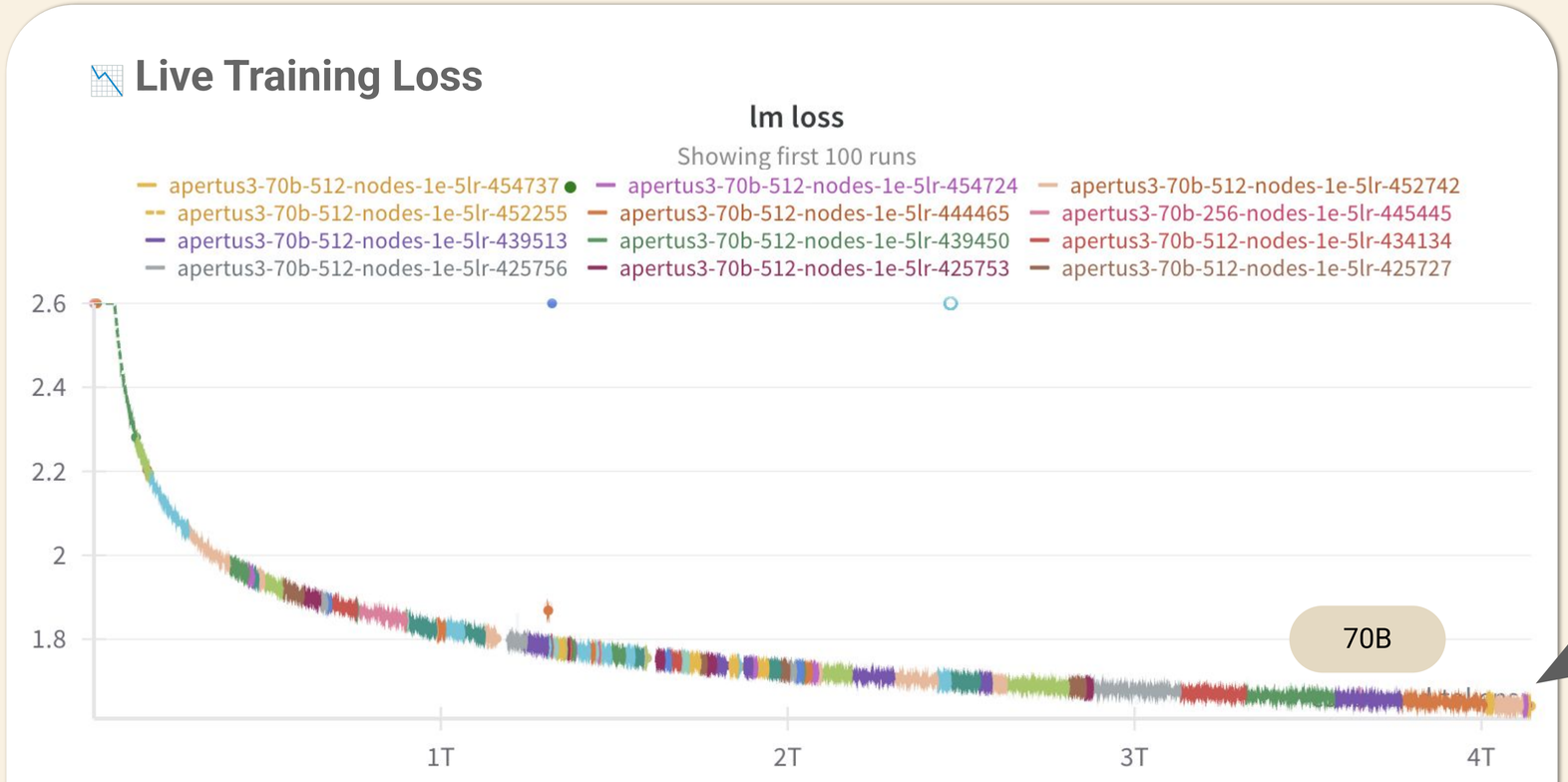
FILTERING	DATA EFFICIENT	TOTAL TOKENS
Transformer based	**3x less data**	**9 trillion**

Messmer, B., Sabolčec, V., & Jaggi, M. (2025). Enhancing Multilingual LLM Pretraining with Model-Based Data Selection. arXiv preprint arXiv:2502.10361.

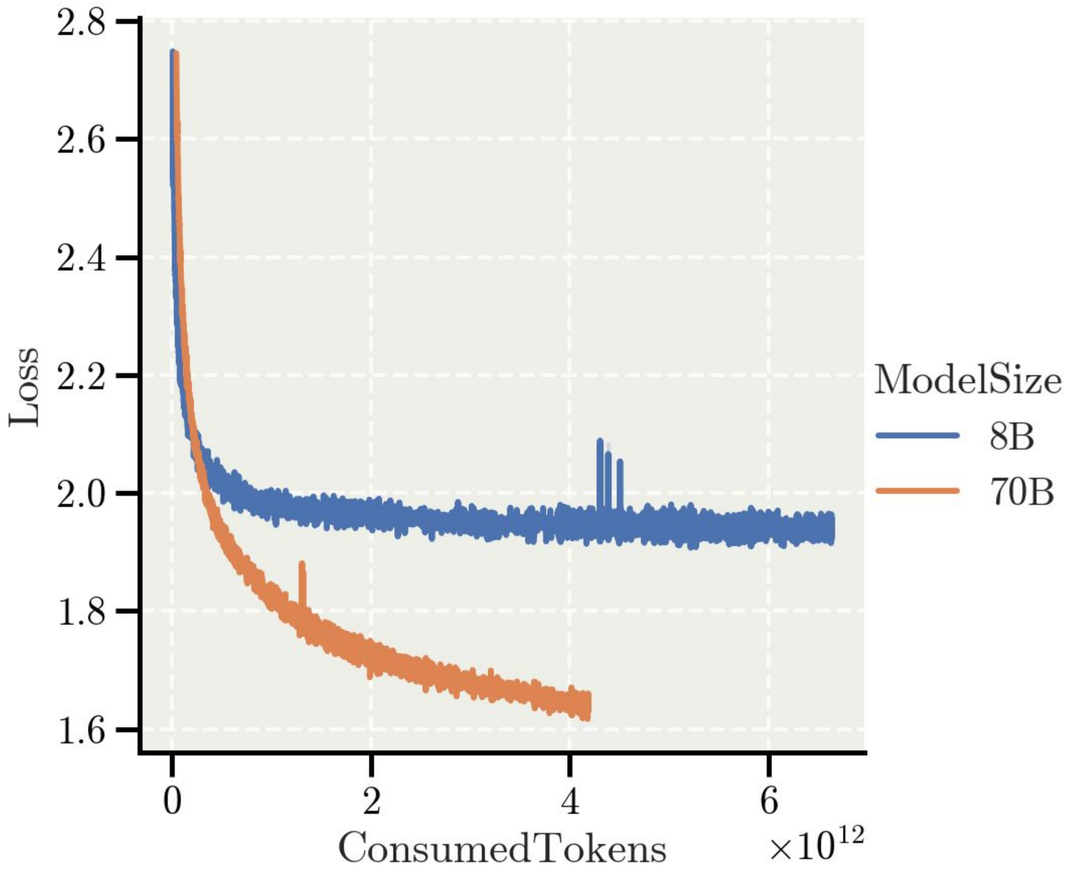
Training *New* 8B and 70B Parameter LLMs

Optimization | Efficiency | **Scale** | Engineering

Pretraining LLMs of varying sizes (in particular 8/70B!) concurrently across 2500+ GPUs with engineering and **scientific innovations** in both modelling and optimization.



and still training!



Not Just Llama: Model and Training Innovations!

We leverage our expertise and ongoing research from machine learning, optimization and deep learning architectures to **find stronger models and more efficient training recipes (*and test them at scale!*).**

OPTIMIZER

AdEMAMix

+ Goldfish Loss

ARCHITECTURE

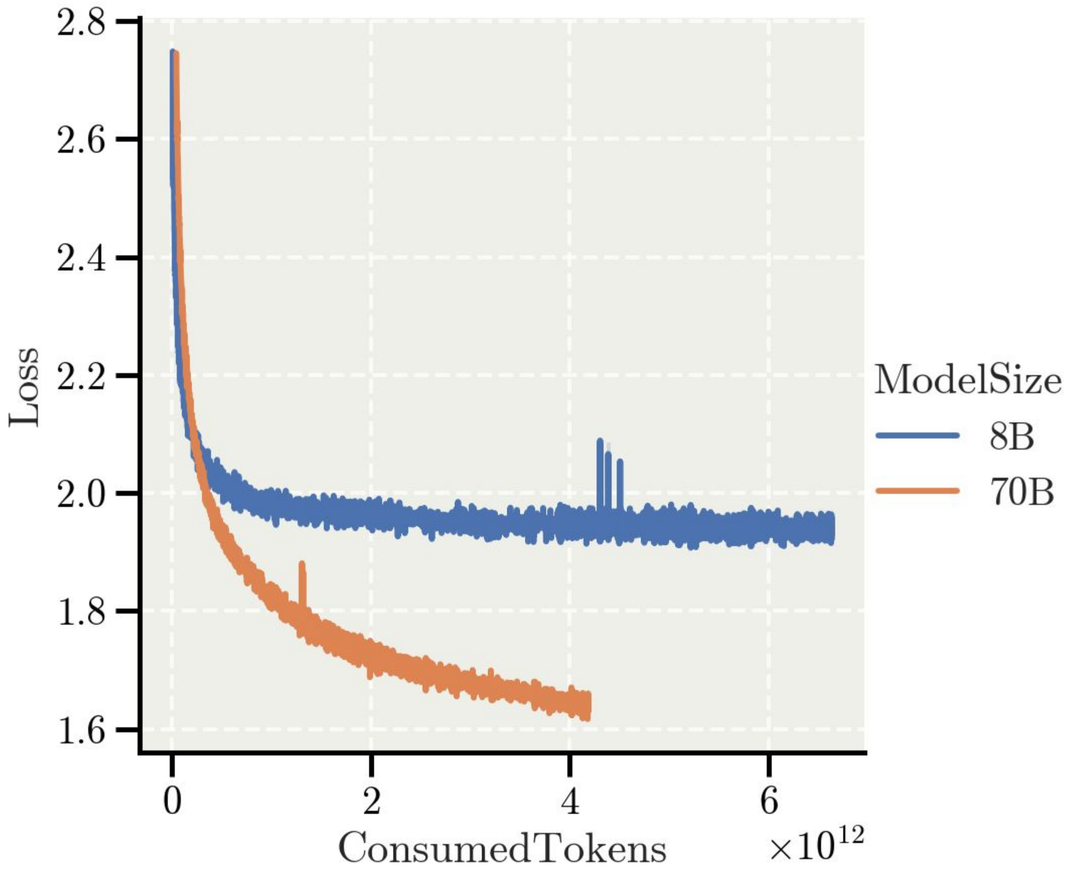
xIELU Activation + QK-Norm

LR SCHEDULE

WSD

For Continual Training

more soon (FP8, MoEs, Distillation, …)



Not Just ML: Low-level software optimizations!

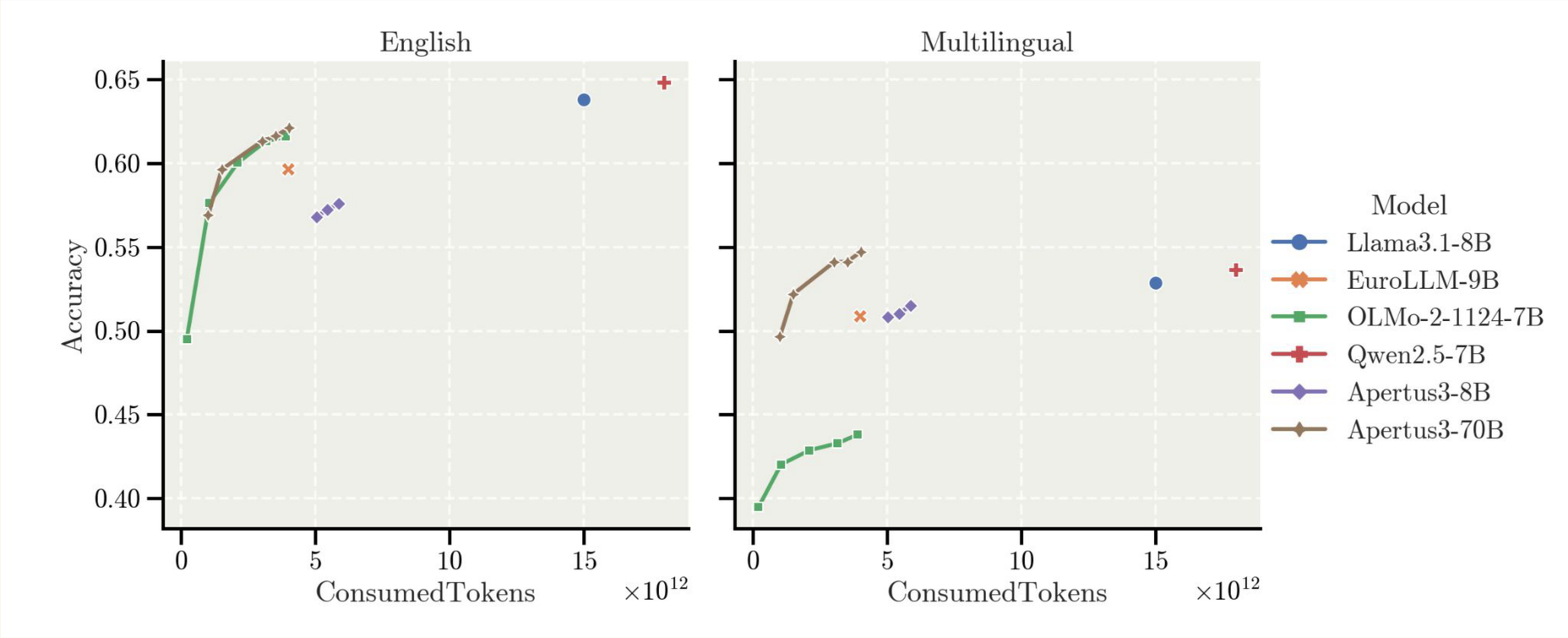
Continuous effort together with CSCS has allowed the large-scale deployment.

Efficient Backend	**Optimized Megatron Framework**	**High-Performance File System**
Node Interconnect	**Custom CUDA Kernels**	**Compute and Communication Overlap**
Optimal DP, TP, PP Size	**Optimal Batch Size**	**...**

Ongoing Evaluation

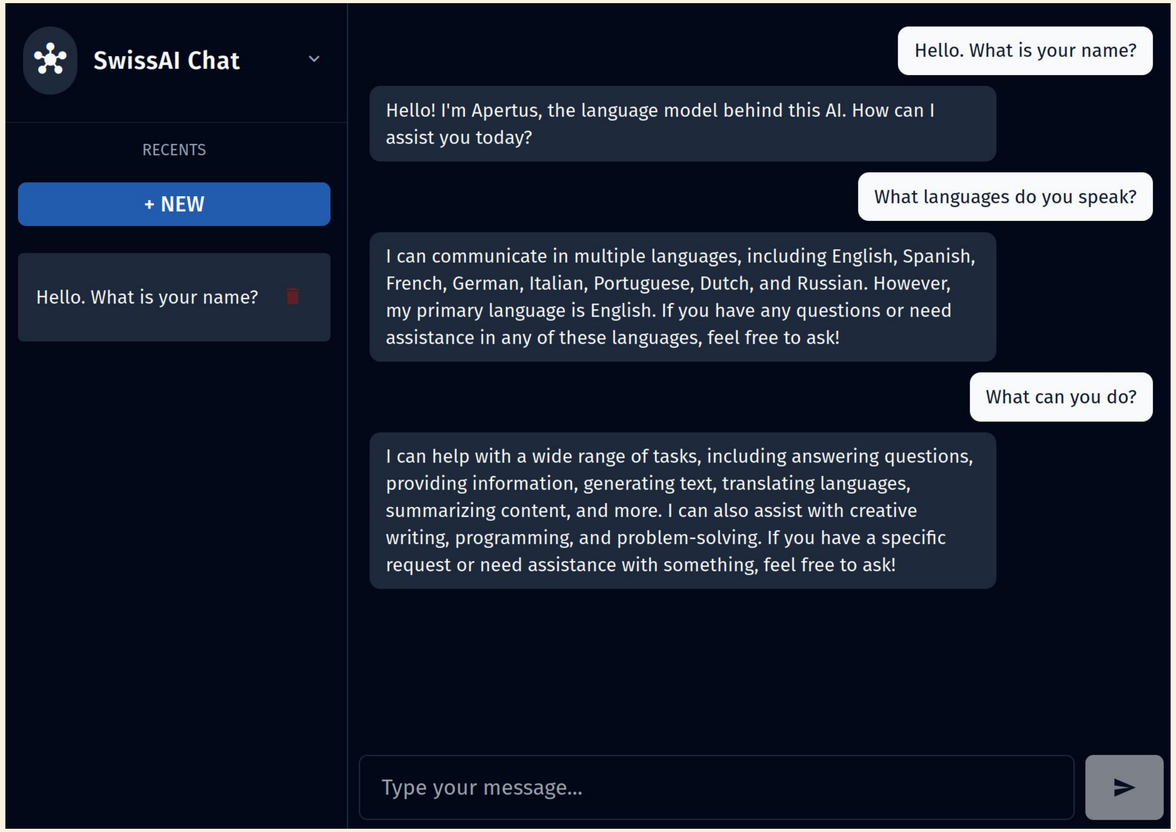
Massive Multitask **Multilingual** Language Understanding

Good English Performance; **Great** Multilingual Capabilities



Post-training pipeline

Using intermediate checkpoints



Early access preview: https://chat.swissai.cscs.ch/.
Full release coming soon!

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH

Thank you 🙏

EPFL

CLAIRE

ML&O

nlp lab


Antoine Bosselut


Martin Jaggi


Imanol Schlag


Antoni-Joan Solergibert


Bettina Messmer


Negar Foroutan


Angelika Romanou


Alexander Hägele


Alejandro Hernández Cano


Vinko Sabolčec


Skander Moalla

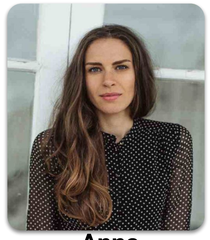

Allen Huang


Maximilian Böther


Simin Fan


Ihor Protsenko


Matteo Pagliardini


Anna Sotnikova


Zeming Chen


Badr AlKhamissi


Syrielle Montariol

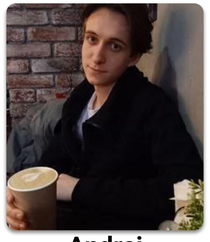

Paul Teiletche


Andreas Marfurt


Kyle Matoba


Roman Macháček


Jan Deriu


Andrei Semenov

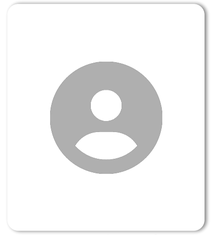

Auguste Poiroux


Dhia Garbaya


Dongyang Fan


Ayush Kumar Tarun


Matin Ansaripour


Juan Garcia


Nicholas Browning


Lukas Drescher


Henrique Mendonça


Theofilos Manitaras


Stefano Schuppli


Joost VandeVondele


Fawzi Roberto Mohamed


Eduard Durech

Ilia Badanin

Tiancheng Chen

... many more!