T. HOEFLER

From Large Language Models to Reasoning Language Models

Three Eras in The Age of Computation.

with contributions by the whole SPCL deep learning team (M. Besta, J. Barth, E. Schreiber, T. Ben-Nun, S. Li, and many others), Microsoft Azure (M. Heddes, J. Belk, S. Scott, D. Goel, M. Castro) and collaborators (D. Alistarh and others)

In Honor of Bill Gropp, ICERM, Providence, September 2025

SPCL

Institute of Science and Technology Austria

Bill and Artificial Intelligence (my jaded perspective ☺)

A high-performance, portable implementation of the MPI message passing interface standard W Gropp, E Lusk, N Doss, A Skjellum Parallel computing 22 (6), 789-828	3569	1996

Bill and Artificial Intelligence ☺

A high-performance, portable implementation of the MPI message passing interface standard 3569 1996
W Gropp, E Lusk, N Doss, A Skjellum
Parallel computing 22 (6), 789-828

- **Much of it started with MPI and parallelism!**
  - Parallel training formed the basis of the AI revolution – from a single GPU to Mega Clusters (100k+ GPUs)

Optimization of collective communication operations in MPICH 1287 2005
R Thakur, R Rabenseifner, W Gropp
The International Journal of High Performance Computing Applications 19 (1 …

Bill and Artificial Intelligence ☺

- **Much of it started with MPI and parallelism!**
  - Parallel training formed the basis of the AI revolution – fro
  - Bill's role in the MPI Forum, especially on collectives in col
- **Pretty much all of AI training runs over NCCL today**
  - Sylvain was heavily inspired by MPI
  - Communicators, Messages, Collectives

    extended with GPU streams

Demystifying NCCL: An In-depth Analysis of GPU Communication Protocols and Algorithms

Zhiyi Hu[1*], Siyuan Shen[1*], Tommaso Bonato[1], Sylvain Jeaugey[2], Cedell Alexander[3], Eric Spada[3], James Dinan[2], Jeff Hammond[2], Torsten Hoefler[1]
[1]ETH Zürich, Switzerland
zhiyhu@student.ethz.ch, {siyuan.shen, tommaso.bonato, torsten.hoefler}@inf.ethz.ch
[2]NVIDIA Corporation, {sjeaugey, jdinan, jehammond}@nvidia.com
[3]Broadcom Inc., {cedell.alexander, eric.spada}@broadcom.com
* These authors contributed equally to this work

***Abstract*—The NVIDIA Collective Communication Library (NCCL) is a critical software layer enabling high-performance collectives on large-scale GPU clusters. Despite being open source with a documented API, its internal design remains largely opaque. The orchestration of communication channels, selection of protocols, and handling of memory movement across devices and nodes are not well understood, making it difficult to analyze performance or identify bottlenecks. This paper presents a comprehensive analysis of NCCL, focusing on its communication protocol variants (Simple, LL, and LL128), mechanisms governing intra-node and inter-node data movement, and ring- and tree-based collective communication algorithms. The insights obtained from this study serve as the foundation for ATLAHS, an application-trace-driven network simulation toolchain capable of accurately reproducing NCCL communication patterns in large-scale AI training workloads. By demystifying NCCL's internal architecture, this work provides guidance for system researchers and performance engineers working to optimize or simulate collective communication at scale.**

***Index Terms*—NVIDIA NCCL, Collective communication, Communication libraries, Multi-GPU cluster training**

I. Introduction

Efficient GPU-to-GPU communication is essential for achieving high performance in distributed artificial intelligence (AI) and high-performance computing (HPC) workloads. The NVIDIA Collective Communication Library (NCCL) is a prominent library widely adopted for scalable, optimized GPU communication [1], [2]. Unlike general-purpose message-

In this paper, we present a thorough and systematic exploration of NCCL's internal architecture. Our analysis specifically targets four primary aspects of NCCL's implementation: (1) a general overview, including API structure and communication channel management; (2) a detailed examination of communication protocols (Simple, LL, LL128); (3) an analysis of its data-transfer models; and (4) comprehensive analysis of its collective communication algorithms.

The insights gained from this study provide important context for performance modeling and architectural optimization. These insights have been adopted in simulation frameworks such as ATLAHS [6], an application-trace-driven network simulator developed to accurately replicate the communication patterns of NCCL-based machine learning workloads. By clarifying NCCL's internal design principles, this analysis supports system researchers, interconnect designers, and network architects in making more informed optimization decisions for GPU-centric high-performance computing environments.

The analysis in this paper is based on NCCL version 2.19.1. While specific implementation details may evolve in future releases, the core architectural mechanisms and communication strategies discussed here are expected to remain consistent, ensuring that the insights presented remain broadly applicable.

II. NCCL Overview

A. NCCL API

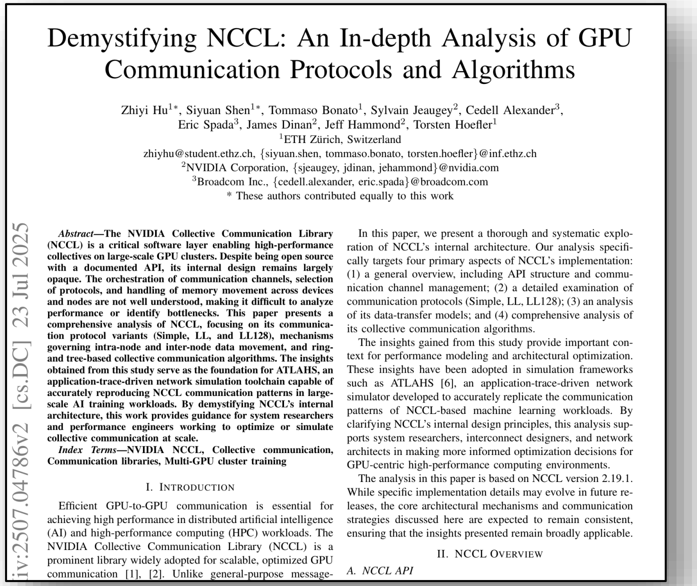
Bill and Artificial Intelligence ☺

[A high-performance, portable implementation of the MPI message passing interface standard](#) 3569 1996
W Gropp, E Lusk, N Doss, A Skjellum
Parallel computing 22 (6), 789-828

[Optimization of collective communication operations in MPICH](#) 1287 2005
R Thakur, R Rabenseifner, W Gropp
The International Journal of High Performance Computing Applications 19 (1 …

- **Much of it started with MPI and parallelism!**
  - Parallel training formed the basis of the AI revolution – from a single GPU to Mega Clusters (100k+ GPUs)
  - Bill's role in the MPI Forum, especially on collectives in collaboration with Rajeev were central

- **Pretty much all of AI training runs over NCCL today**
  - Sylvain was heavily inspired by MPI
  - Communicators, Messages, Collectives
    extended with GPU streams



Demystifying NCCL: An In-depth Analysis of GPU Communication Protocols and Algorithms

Zhiyi Hu[1*], Siyuan Shen[1*], Tommaso Bonato[1], Sylvain Jeaugey[2], Cedell Alexander[3], Eric Spada[3], James Dinan[2], Jeff Hammond[2], Torsten Hoefler[1]
[1]ETH Zürich, Switzerland
zhiyhu@student.ethz.ch, {siyuan.shen, tommaso.bonato, torsten.hoefler}@inf.ethz.ch
[2]NVIDIA Corporation, {sjeaugey, jdinan, jehammond}@nvidia.com
[3]Broadcom Inc., {cedell.alexander, eric.spada}@broadcom.com
* These authors contributed equally to this work

iv:2507.04786v2 [cs.DC] 23 Jul 2025

***Abstract*—The NVIDIA Collective Communication Library (NCCL) is a critical software layer enabling high-performance collectives on large-scale GPU clusters. Despite being open source with a documented API, its internal design remains largely opaque. The orchestration of communication channels, selection of protocols, and handling of memory movement across devices and nodes are not well understood, making it difficult to analyze performance or identify bottlenecks. This paper presents a comprehensive analysis of NCCL, focusing on its communication protocol variants (Simple, LL, and LL128), mechanisms governing intra-node and inter-node data movement, and ring- and tree-based collective communication algorithms. The insights obtained from this study serve as the foundation for ATLAHS, an application-trace-driven network simulation toolchain capable of accurately reproducing NCCL communication patterns in large-scale AI training workloads. By demystifying NCCL's internal architecture, this work provides guidance for system researchers and performance engineers working to optimize or simulate collective communication at scale.**

***Index Terms*—NVIDIA NCCL, Collective communication, Communication libraries, Multi-GPU cluster training**

I. Introduction

Efficient GPU-to-GPU communication is essential for achieving high performance in distributed artificial intelligence (AI) and high-performance computing (HPC) workloads. The NVIDIA Collective Communication Library (NCCL) is a prominent library widely adopted for scalable, optimized GPU communication [1], [2]. Unlike general-purpose message-

In this paper, we present a thorough and systematic exploration of NCCL's internal architecture. Our analysis specifically targets four primary aspects of NCCL's implementation: (1) a general overview, including API structure and communication channel management; (2) a detailed examination of communication protocols (Simple, LL, LL128); (3) an analysis of its data-transfer models; and (4) comprehensive analysis of its collective communication algorithms.

The insights gained from this study provide important context for performance modeling and architectural optimization. These insights have been adopted in simulation frameworks such as ATLAHS [6], an application-trace-driven network simulator developed to accurately replicate the communication patterns of NCCL-based machine learning workloads. By clarifying NCCL's internal design principles, this analysis supports system researchers, interconnect designers, and network architects in making more informed optimization decisions for GPU-centric high-performance computing environments.

The analysis in this paper is based on NCCL version 2.19.1. While specific implementation details may evolve in future releases, the core architectural mechanisms and communication strategies discussed here are expected to remain consistent, ensuring that the insights presented remain broadly applicable.

II. NCCL Overview

A. NCCL API

- **I am lucky to know Bill a… w!**
  - In base 2 ;-)

MPI

Bill and Artificial Intelligence ☺

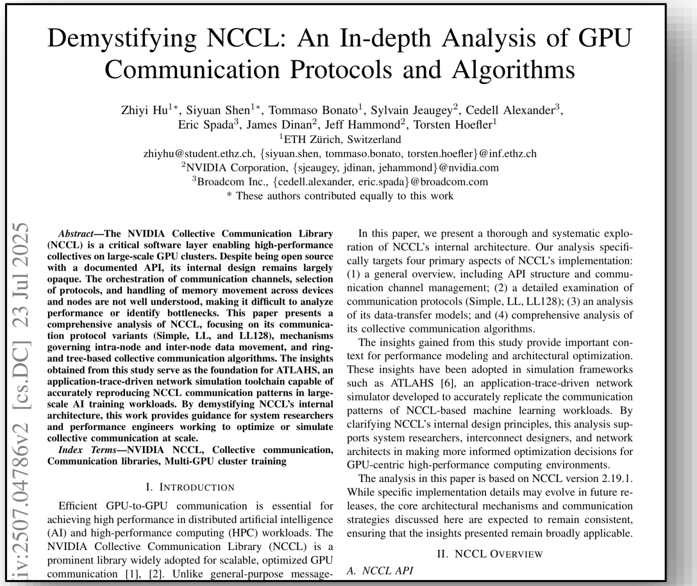
A high-performance, portable implementation of the MPI message passing interface standard — 3569 — 1996
W Gropp, E Lusk, N Doss, A Skjellum
Parallel computing 22 (6), 789-828

Optimization of collective communication operations in MPICH — 1287 — 2005
R Thakur, R Rabenseifner, W Gropp
The International Journal of High Performance Computing Applications 19 (1 …

- **Much of it started with MPI and parallelism!**
  - Parallel training formed the basis of the AI revolution – from a single GPU to Mega Clusters (100k+ GPUs)
  - Bill's role in the MPI Forum, especially on collectives in collaboration with Rajeev were central

- **Pretty much all of AI training runs over NCCL today**
  - Sylvain was heavily inspired by MPI
  - Communicators, Messages, Collectives

    extended with GPU streams



Demystifying NCCL: An In-depth Analysis of GPU Communication Protocols and Algorithms

Zhiyi Hu[1*], Siyuan Shen[1*], Tommaso Bonato[1], Sylvain Jeaugey[2], Cedell Alexander[3], Eric Spada[3], James Dinan[2], Jeff Hammond[2], Torsten Hoefler[1]
[1]ETH Zürich, Switzerland
zhiyhu@student.ethz.ch, {siyuan.shen, tommaso.bonato, torsten.hoefler}@inf.ethz.ch
[2]NVIDIA Corporation, {sjeaugey, jdinan, jehammond}@nvidia.com
[3]Broadcom Inc., {cedell.alexander, eric.spada}@broadcom.com
* These authors contributed equally to this work

iv:2507.04786v2 [cs.DC] 23 Jul 2025

***Abstract*—The NVIDIA Collective Communication Library (NCCL) is a critical software layer enabling high-performance collectives on large-scale GPU clusters. Despite being open source with a documented API, its internal design remains largely opaque. The orchestration of communication channels, selection of protocols, and handling of memory movement across devices and nodes are not well understood, making it difficult to analyze performance or identify bottlenecks. This paper presents a comprehensive analysis of NCCL, focusing on its communication protocol variants (Simple, LL, and LL128), mechanisms governing intra-node and inter-node data movement, and ring- and tree-based collective communication algorithms. The insights obtained from this study serve as the foundation for ATLAHS, an application-trace-driven network simulation toolchain capable of accurately reproducing NCCL communication patterns in large-scale AI training workloads. By demystifying NCCL's internal architecture, this work provides guidance for system researchers and performance engineers working to optimize or simulate collective communication at scale.**

***Index Terms*—NVIDIA NCCL, Collective communication, Communication libraries, Multi-GPU cluster training**

I. INTRODUCTION

Efficient GPU-to-GPU communication is essential for achieving high performance in distributed artificial intelligence (AI) and high-performance computing (HPC) workloads. The NVIDIA Collective Communication Library (NCCL) is a prominent library widely adopted for scalable, optimized GPU communication [1], [2]. Unlike general-purpose message-

In this paper, we present a thorough and systematic exploration of NCCL's internal architecture. Our analysis specifically targets four primary aspects of NCCL's implementation: (1) a general overview, including API structure and communication channel management; (2) a detailed examination of communication protocols (Simple, LL, LL128); (3) an analysis of its data-transfer models; and (4) comprehensive analysis of its collective communication algorithms.

The insights gained from this study provide important context for performance modeling and architectural optimization. These insights have been adopted in simulation frameworks such as ATLAHS [6], an application-trace-driven network simulator developed to accurately replicate the communication patterns of NCCL-based machine learning workloads. By clarifying NCCL's internal design principles, this analysis supports system researchers, interconnect designers, and network architects in making more informed optimization decisions for GPU-centric high-performance computing environments.

The analysis in this paper is based on NCCL version 2.19.1. While specific implementation details may evolve in future releases, the core architectural mechanisms and communication strategies discussed here are expected to remain consistent, ensuring that the insights presented remain broadly applicable.

II. NCCL OVERVIEW

A. NCCL API

- **I am lucky to know Bill and be his di … rs now!**
  - In base 2 ;-)

- **Not only technical interactions**

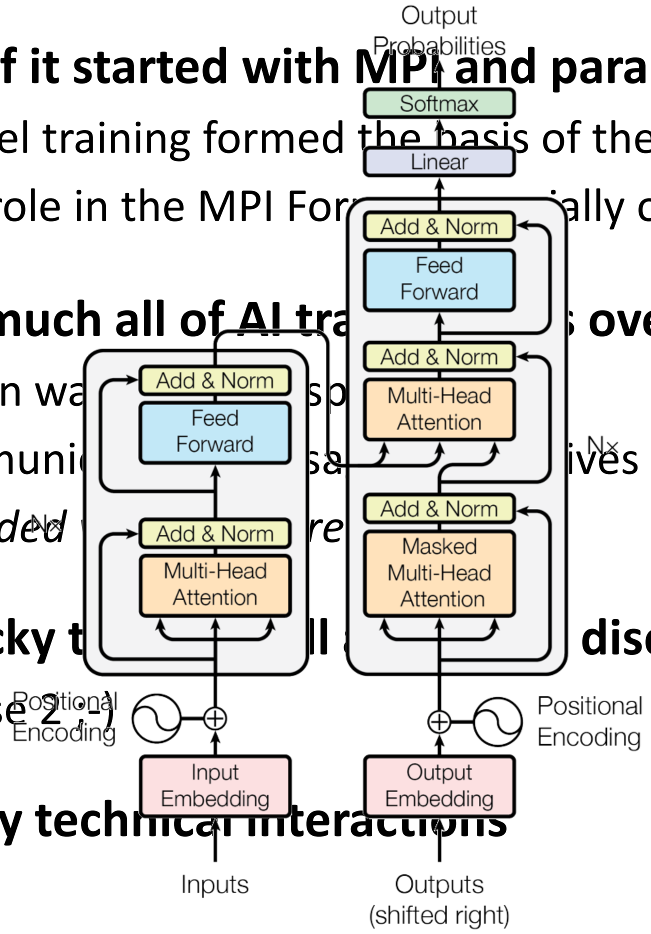
MPI

Bill and Artificial Intelligence ☺

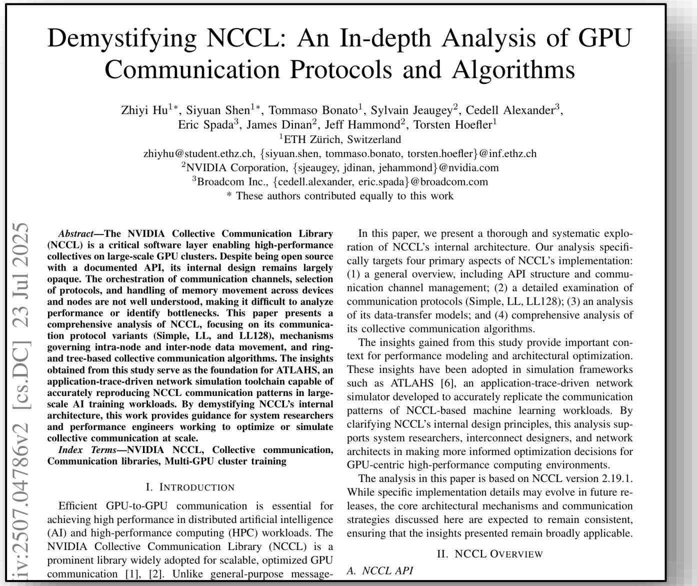
A high-performance, portable implementation of the MPI message passing interface standard | 3569 | 1996
W Gropp, E Lusk, N Doss, A Skjellum
Parallel computing 22 (6), 789-828

Optimization of collective communication operations in MPICH | 1287 | 2005
R Thakur, R Rabenseifner, W Gropp
The International Journal of High Performance Computing Applications 19 (1 …

- **Much of it started with MPI and parallelism!**
  - Parallel training formed the basis of the AI revolution – from a single GPU to Mega Clusters (100k+ GPUs)
  - Bill's role in the MPI For… ially on collectives in collaboration with Rajeev were central



- **Pretty much all of AI tra… over NCCL today**
  - Sylvain wa… sp…
  - Communi… sa… ives *extended* … re…



Demystifying NCCL: An In-depth Analysis of GPU Communication Protocols and Algorithms

Zhiyi Hu[1*], Siyuan Shen[1*], Tommaso Bonato[1], Sylvain Jeaugey[2], Cedell Alexander[3], Eric Spada[3], James Dinan[2], Jeff Hammond[2], Torsten Hoefler[1]
[1]ETH Zürich, Switzerland
zhiyhu@student.ethz.ch, {siyuan.shen, tommaso.bonato, torsten.hoefler}@inf.ethz.ch
[2]NVIDIA Corporation, {sjeaugey, jdinan, jehammond}@nvidia.com
[3]Broadcom Inc., {cedell.alexander, eric.spada}@broadcom.com
* These authors contributed equally to this work

iv:2507.04786v2 [cs.DC] 23 Jul 2025

***Abstract*—The NVIDIA Collective Communication Library (NCCL) is a critical software layer enabling high-performance collectives on large-scale GPU clusters. Despite being open source with a documented API, its internal design remains largely opaque. The orchestration of communication channels, selection of protocols, and handling of memory movement across devices and nodes are not well understood, making it difficult to analyze performance or identify bottlenecks. This paper presents a comprehensive analysis of NCCL, focusing on its communication protocol variants (Simple, LL, and LL128), mechanisms governing intra-node and inter-node data movement, and ring- and tree-based collective communication algorithms. The insights obtained from this study serve as the foundation for ATLAHS, an application-trace-driven network simulation toolchain capable of accurately reproducing NCCL communication patterns in large-scale AI training workloads. By demystifying NCCL's internal architecture, this work provides guidance for system researchers and performance engineers working to optimize or simulate collective communication at scale.**

***Index Terms*—NVIDIA NCCL, Collective communication, Communication libraries, Multi-GPU cluster training**

I. Introduction

Efficient GPU-to-GPU communication is essential for achieving high performance in distributed artificial intelligence (AI) and high-performance computing (HPC) workloads. The NVIDIA Collective Communication Library (NCCL) is a prominent library widely adopted for scalable, optimized GPU communication [1], [2]. Unlike general-purpose message-

In this paper, we present a thorough and systematic exploration of NCCL's internal architecture. Our analysis specifically targets four primary aspects of NCCL's implementation: (1) a general overview, including API structure and communication channel management; (2) a detailed examination of communication protocols (Simple, LL, LL128); (3) an analysis of its data-transfer models; and (4) comprehensive analysis of its collective communication algorithms.

The insights gained from this study provide important context for performance modeling and architectural optimization. These insights have been adopted in simulation frameworks such as ATLAHS [6], an application-trace-driven network simulator developed to accurately replicate the communication patterns of NCCL-based machine learning workloads. By clarifying NCCL's internal design principles, this analysis supports system researchers, interconnect designers, and network architects in making more informed optimization decisions for GPU-centric high-performance computing environments.

The analysis in this paper is based on NCCL version 2.19.1. While specific implementation details may evolve in future releases, the core architectural mechanisms and communication strategies discussed here are expected to remain consistent, ensuring that the insights presented remain broadly applicable.

II. NCCL Overview

A. NCCL API

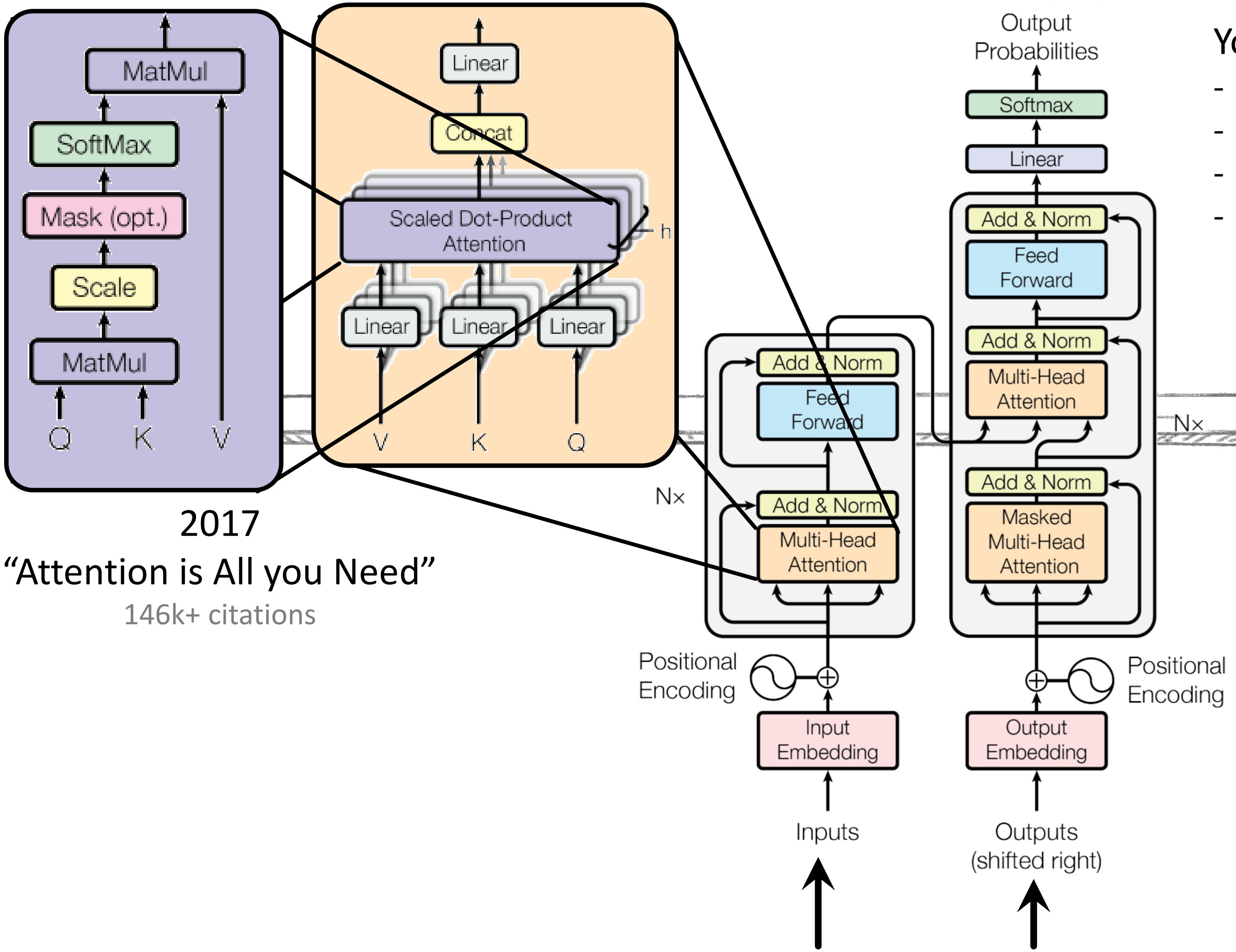
- **I am lucky t… ll … disciple for more than 10000 years now!**
  - In base 2 ;-)

- **Not only techn… interactions**

- **And here we are! Now what happened in AI and how?**
  - We got to gigantic models trained on gigantic systems – need to make those cheap
  - And what will the future bring? Artificial Human Intelligence?

From LLMs to AHI

Grossartig (90%)
Gross (7%)
Krass (2%)



2017
"Attention is All you Need"
146k+ citations

“Transformers are great" Transformer sind

You can explain the computation to your grandmother!

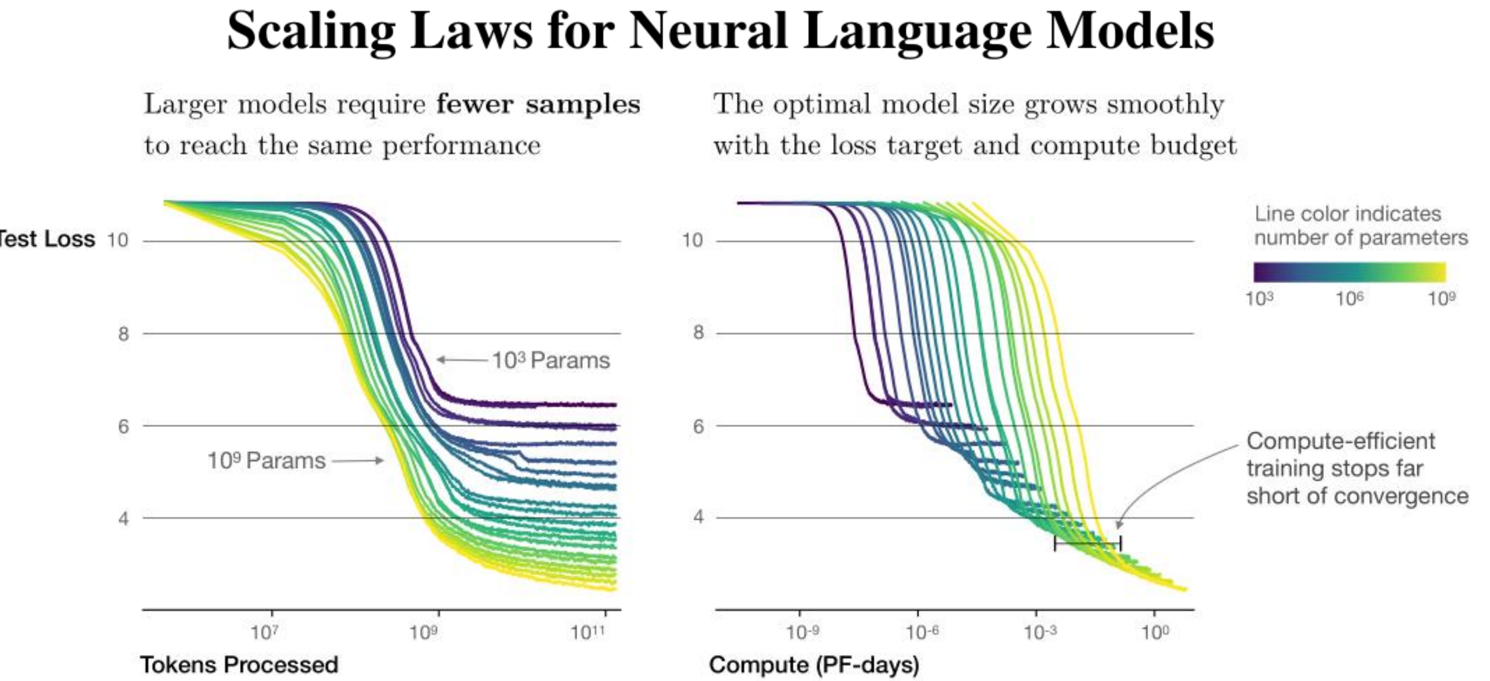
- Three simple kernels: MMM, Softmax, Layernorm
- >95%+ matrix multiplication
- Great fit for HPC GPUs
- Easy to parallelize

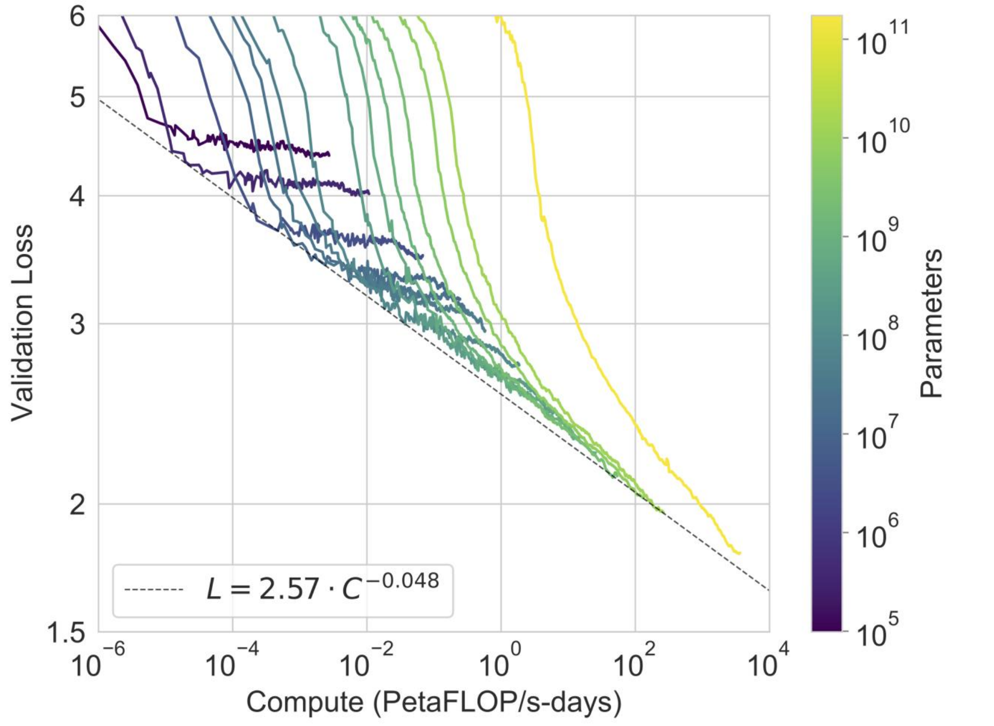
Text is encoded as tokens (very important!)

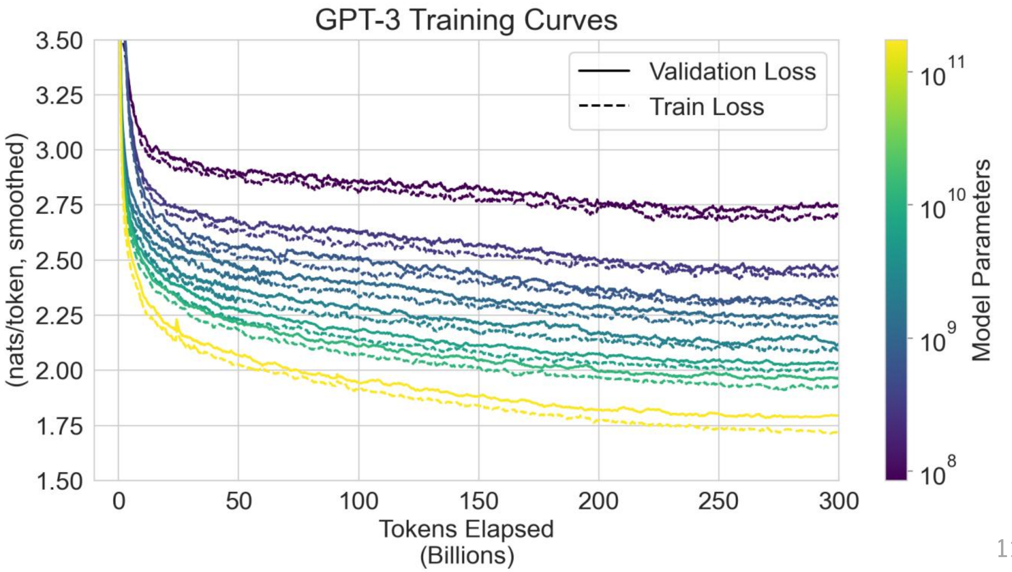
- Tokens are offsets into learned vector tables
- Often learned based on statistics
- Most common sub-strings (e.g., Byte Pair Encoding)
- Think of them as vectors
- Word2vec: “Efficient Estimation of Word Representations in Vector Space“ (45k+ citations)

From LLMs to AHI

```
Poor English input:  I eated the purple berries.
Good English output:  I ate the purple berries.
Poor English input:  Thank you for picking me as your designer.  I'd appreciate it.
Good English output:  Thank you for choosing me as your designer.  I appreciate it.
Poor English input:  The mentioned changes have done.  or I did the alteration that you
requested.  or I changed things you wanted and did the modifications.
Good English output:  The requested changes have been made.  or I made the alteration that you
requested.  or I changed things you wanted and made the modifications.
Poor English input:  I'd be more than happy to work with you in another project.
```
Good English output: I'd be more than happy to work with you on another project.







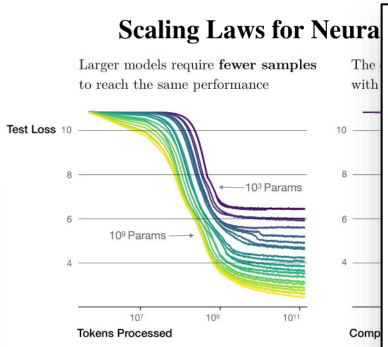
From LLMs to AHI

Microsoft invests $1 billion in OpenAI to pursue holy grail of artificial intelligence

Building artificial general intelligence is OpenAI's ambitious goal

By James Vincent | Jul 22, 2019, 10:08am EDT

"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"



2020 - **GPT-3** (202

"Language Mo

Few-Shot Lea

37k+ citati

Tesla unveils Dojo supercomputer: world's new most powerful AI training machine

Fred Lambert - Aug. 20th 2021 3:08 am PT @FredericLambert

Trump's AI Push: Understanding The $500 Billion Stargate Initiative

Garth Friesen Contributor

Specialist in global markets, economics and alternative investments.

Follow

Jan 23, 2025, 08:00am EST

Updated Jan 24, 2025, 07:25am EST

CNN Business Markets More Watch Listen Live TV

Microsoft plans to invest $80 billion on AI-enabled data centers in fiscal 2025

By Reuters

1 minute read · Published 4:50 PM EST, Fri January 3, 2025

Facebook parent Meta creates powerful AI supercomputer

Facebook's parent company Meta says it has created what it believes is among the fastest artificial intelligence supercomputers running today

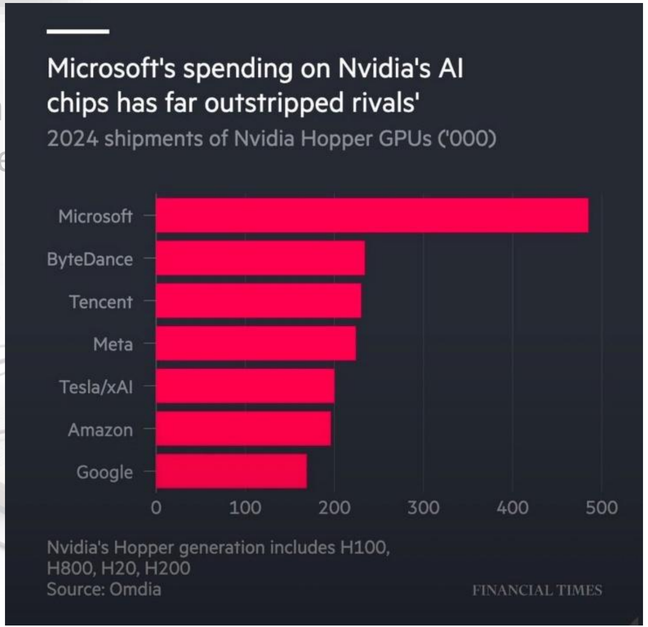
By The Associated Press
January 24, 2022, 10:33 PM

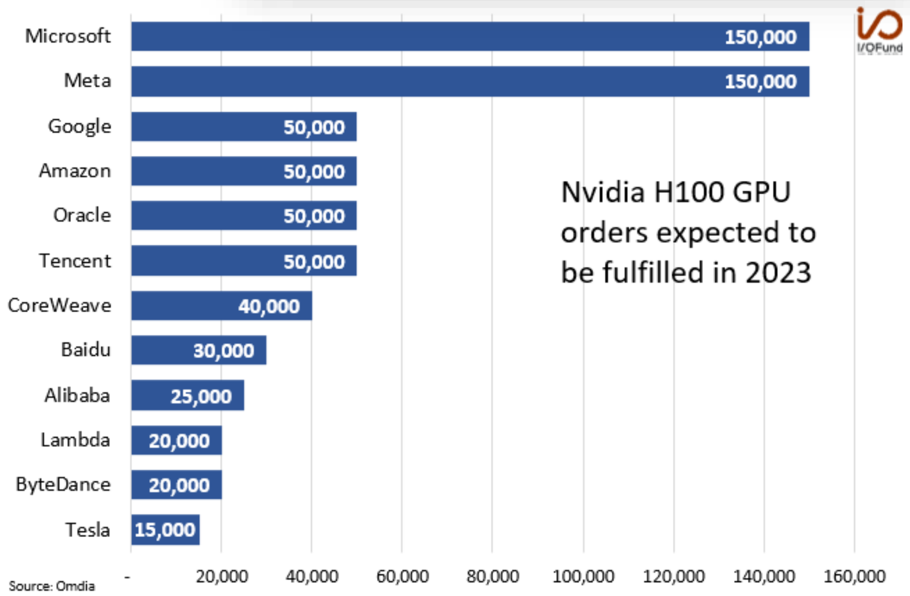
Share

BABY STEPS Google artificial intelligence supercomputer creates its own 'AI child' that can outperform its human-made rivals

The NASNet system was created by a neural network called AutoML earlier this year

Mark Hodge
15:22, 5 Dec 2017 | Updated: 11:27, 6 Dec 2017





Supercomputers fuel Modern AI

Facebook parent Meta creates powerful AI supercomputer

Facebook's parent company Meta says it has created what it beli... intelligence supercomputers running today

By The Associated Press
January 24, 2022, 10:33 PM

Meta is spending $30 billion on NVIDIA GPUs just to train their AI that will surpass human intelligence.

AjayKrish · Follow
3 min read · May 4, 2024

Tesla unveils Dojo supercomputer: world's new most powerful AI training machine

Fred Lambert - Aug. 20th 2021 3:08 am PT @FredericLambert

BABY STEPS Google artificial intelligence supercomputer creates its own 'AI child' that can outperform its human-made rivals

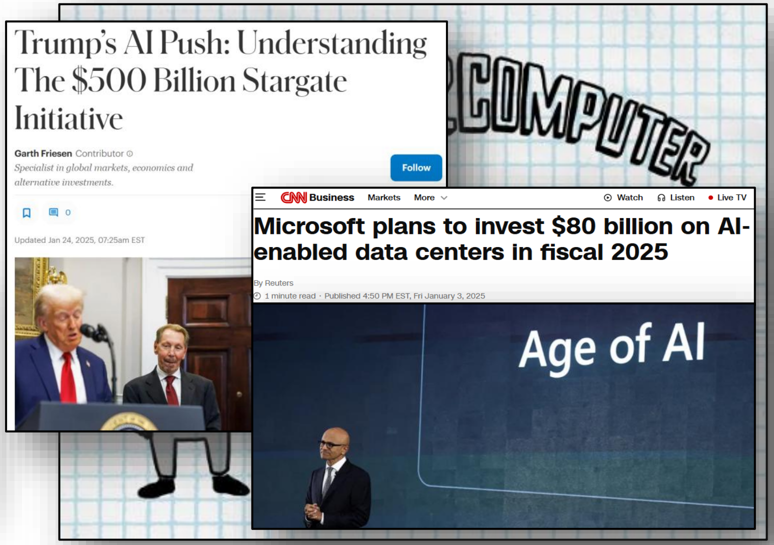
The NASNet system was created by a neural network called AutoML earlier this year

Mark Hodge
15:22, 5 Dec 2017 | Updated: 11:27, 6 Dec 2017

Microsoft invests $1 billion in OpenAI to pursue holy grail of artificial intelligence

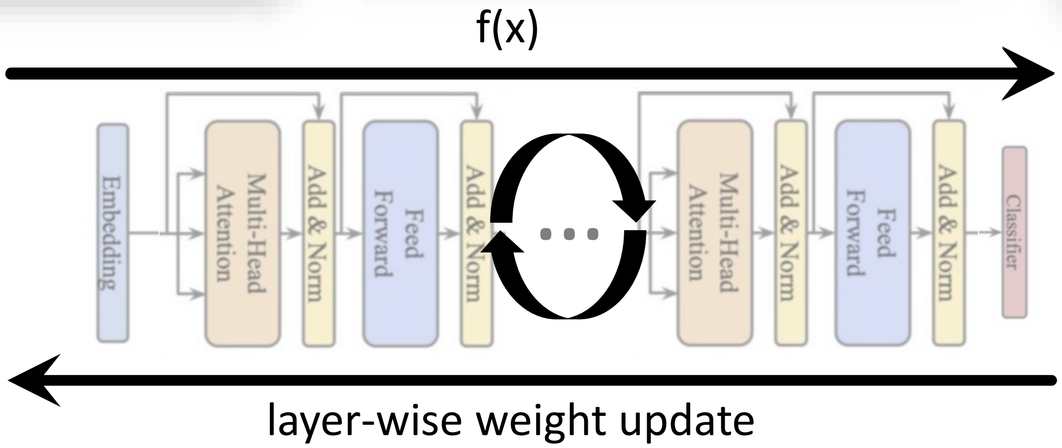
Building artificial general intelligence is OpenAI's ambitious goal

By James Vincent | Jul 22, 2019, 10:08am EDT



reddit

A robot may not injure a human being or, through inaction, allow a human being to come to ___



harm	0.74
injury	0.28
now	0.07
never	0.04
pain	0.33
boat	0.02
house	0.02

harm	1.00
injury	0.00
now	0.00
never	0.00
pain	0.00
boat	0.00
house	0.00

- PaLM-540B: 1.4 trillion tokens
- ImageNet (22k): A few TB
- Actually: **the whole internet**!

- PaLM-540B: 118 (complex) layers

  *540 bn parameters (**1 TiB** in fp16)*

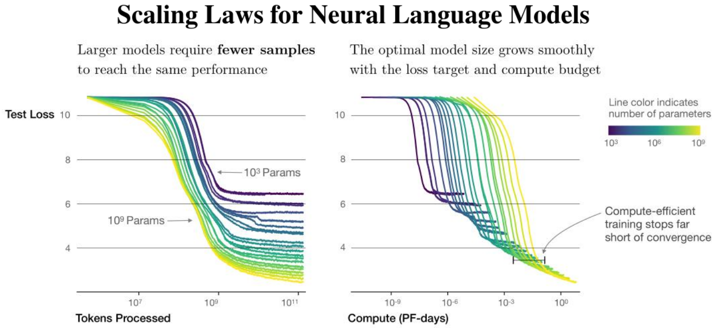
  2048-token "sentences"

- PaLM-540B: 256k token dict
- **takes weeks to train**

T. Ben-Nun, TH: Demystifying parallel and distributed deep learning: An in-depth concurrency analysis, ACM Computing Surveys (CSUR), 2019

From LLMs to AHI

2018 - **BERT**
"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"
122k+ citations

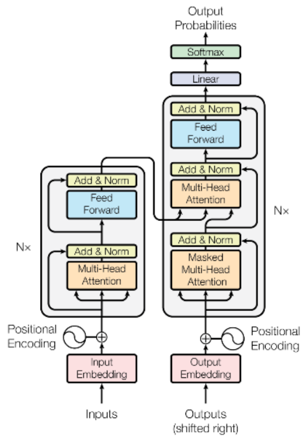


2020 - **GPT-3** (2020, scaling laws)
"Language Models are Few-Shot Learners"
37k+ citations

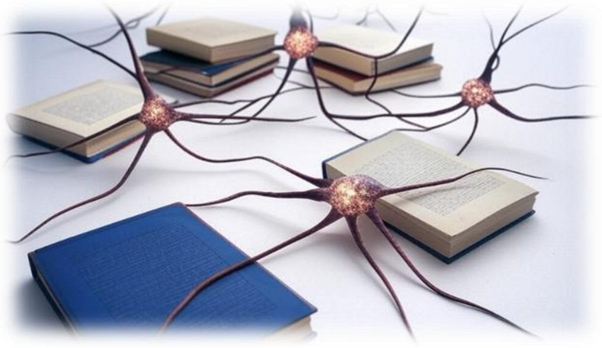
How to turn this into a business serving millions of customers?

era of model size scaling

2017 - **Transformers**
"Attention is All you Need"
146k+ citations



2019 - **GPT-2**
"Language Models are Unsupervised Multitask Learners"
14k+ citations



2022 – **ChatGPT** (RLHF, 2023, DPO)
"Training language models to follow instructions with human feedback"
14k+ citations

ChatGPT

From LLMs to AHI

How to turn this into a business serving millions of customers?

LLaMA by Meta

2023 – **Llama** (Qwen, Grok, etc.)
"LLaMA: Open and Efficient Foundation Language Models"
11k+ citations

Compete through Openness

Needs even more (pre)training compute!

era of data scaling

Reduce cost

smaller models, more better data, more training compute

optimize models computationally

reduce hardware cost and increase efficiency

2022 – **ChatGPT** (RLHF, 2023, DPO)
"Training language models to follow instructions with human feedback"
14k+ citations

ChatGPT

Optimization Determines the Future of AI

Optimization Determines the Future of AI

We need a Scientific Approach to it

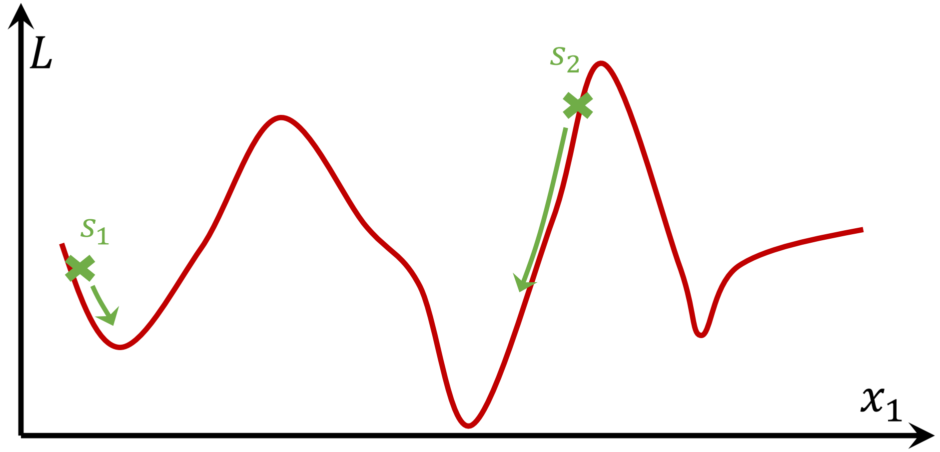
Next, let's see how to improve cost by 1,000x

Moving Data is Most Expensive!

Techniques to Shrink ML Data

Quantization – Running Gigantic LLMs on Reasonable Systems (arXiv:2210.17323)

- **Brains have limited precision! Why are we computing with FP32?**
  - Neurons in Hippocampus can "reliably distinguish 24 strengths" [1]
    4.6 bits of information!
  - For technical reasons (SGD, optimization, how we quantize)
- **PaLM-540B has up to 540 billion parameters**
  - 1.08 TiB in FP16/BF16, 540 GiB in FP8 ☹
  - Rounding to <5 bits is not so simple
  - Requires some foundation and many tricks
- **Consider "error landscape" of a trained model with weights w [2]**

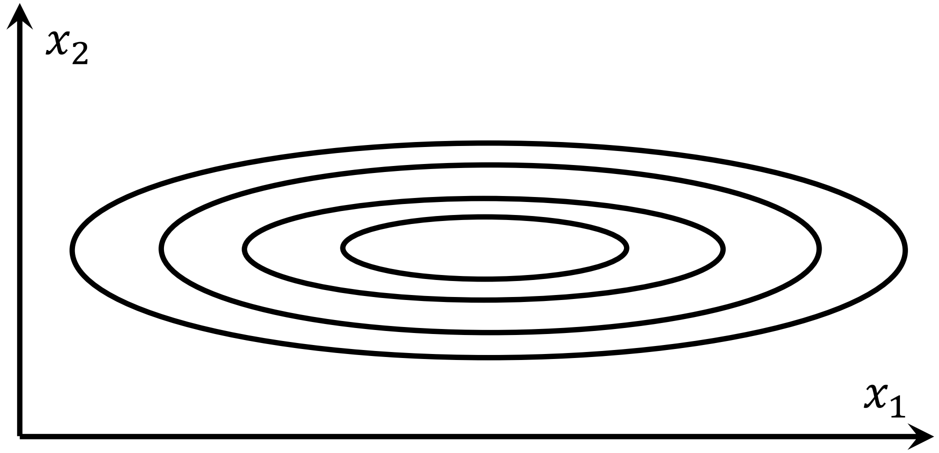


$$\partial E = \left(\frac{\partial E}{\partial w}\right)^T \partial w + \frac{1}{2}\partial w^T \left(\frac{\partial^2 E}{\partial^2 w}\right)\partial w + O(|\partial w|^3)$$

Gradient
(≈0)

"Curvature" of error
(aka. "sensitivity")

Higher-order terms
(=0 for quadratic loss)



[1] Bartol et al., "Hippocampal Spine Head Sizes Are Highly Precise", eLife 2015
[2] LeCun, Denker, Solla: "Optimal Brain Damage", NIPS'90

Quantization – Running Gigantic LLMs on Reasonable Systems (arXiv:2210.17323)

- **Quantization objective for low precision rounded weights $\hat{w}$**

  $$\mathrm{argmin}_{\hat{w}} \, \|wx - \hat{w}x\|^2$$

- **Solve PTQ optimization problem row by row of w**
  - Round row and push the error forward using the inverse Hessian
  - Update Hessian for each column
- **Tricks**
  - Block updates for better locality (10x speedup)
  - Use Cholesky to invert Hessian (higher stability)
  - Work one transformer block at a time (6 operators fit in memory)
  - Use quantized input from previous blocks for block i
- **Results**
  - Generative inference 2-4x faster
  - 3 bits → 66 GiB, fits in a single (high-end) A100 GPU!

Model	FP16	1024	512	256	128	64	32	3-bit
OPT-175B	8.34	11.84	10.85	10.00	9.58	9.18	8.94	8.68
BLOOM	8.11	11.80	10.84	10.13	9.55	9.17	8.83	8.64

Table 6: 2-bit GPTQ quantization results with varying group-sizes; perplexity on WikiText2.

GPTQ: ACCURATE POST-TRAINING QUANTIZATION FOR GENERATIVE PRE-TRAINED TRANSFORMERS

A PREPRINT

Elias Frantar*
IST Austria
Klosterneuburg, Austria
elias.frantar@ist.ac.at

Saleh Ashkboos
ETH Zurich
Switzerland
saleh.ashkboos@inf.ethz.ch

Torsten Hoefler
ETH Zurich
Switzerland
htor@inf.ethz.ch

Dan Alistarh
IST Austria & Neural Magic, Inc.
Klosterneuburg, Austria
dan.alistarh@ist.ac.at

ABSTRACT

Generative Pre-trained Transformer (GPT) models set themselves apart through breakthrough performance across complex language modelling tasks, but also by their extremely high computational and storage costs. Specifically, due to their massive size, even inference for large, highly-accurate GPT models may require multiple performant GPUs to execute, which limits the usability of such models. While there is emerging work on relieving this pressure via model compression, the applicability and performance of existing compression techniques is limited by the scale and complexity of GPT models. In this paper, we address this challenge, and propose GPTQ, a new one-shot weight quantization method based on approximate second-order information, that is both highly-accurate and highly-efficient. Specifically, GPTQ can quantize GPT models with 175 billion parameters in approximately four GPU hours, reducing the bitwidth down to 3 or 4 bits per weight,

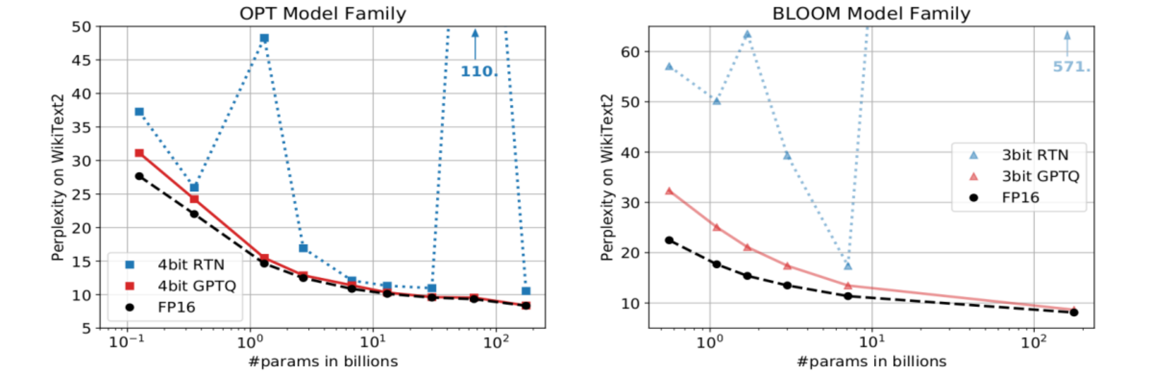


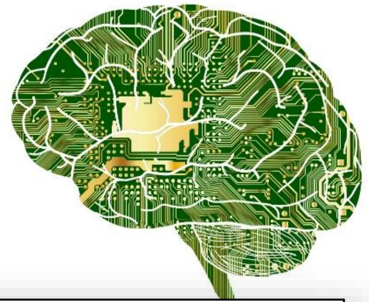
Figure 1: Quantizing OPT models to 4 and BLOOM models to 3 bit precision, comparing GPTQ with the FP16 baseline and round-to-nearest (RTN) [34, 5].

Quantization Reduces Data by an Order of Magnitude

10x

How to Go Further?

Model Sparsification … (arXiv:2102.00554)



- **Brains are not densely connected! Why are DNN computations dense?**
  - For technical reasons (training, implementation etc.)
  - We may want to shift towards sparse!

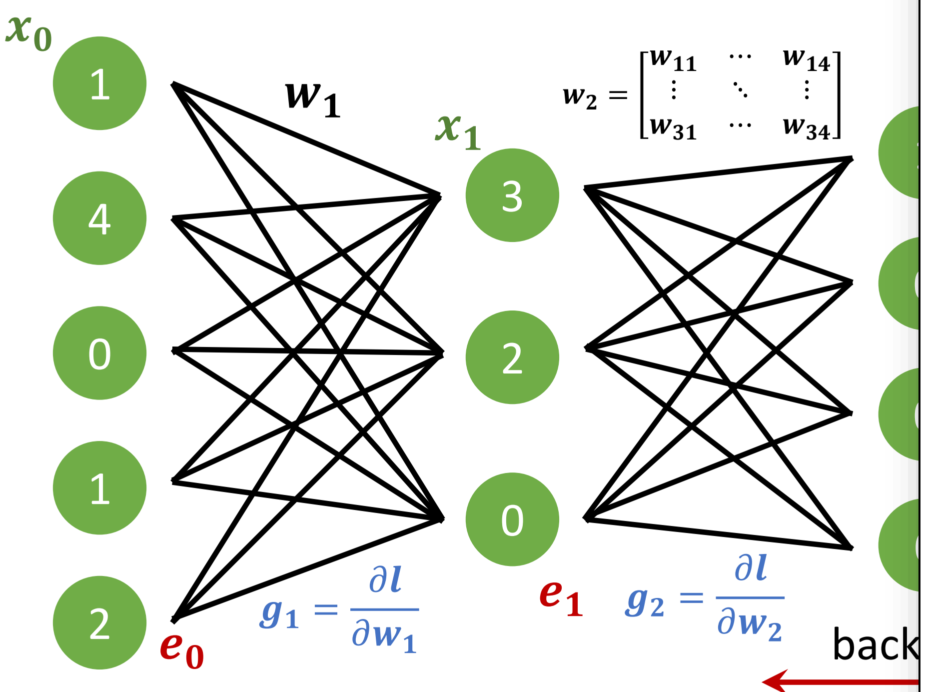
Intuition: not all features are always relevant!

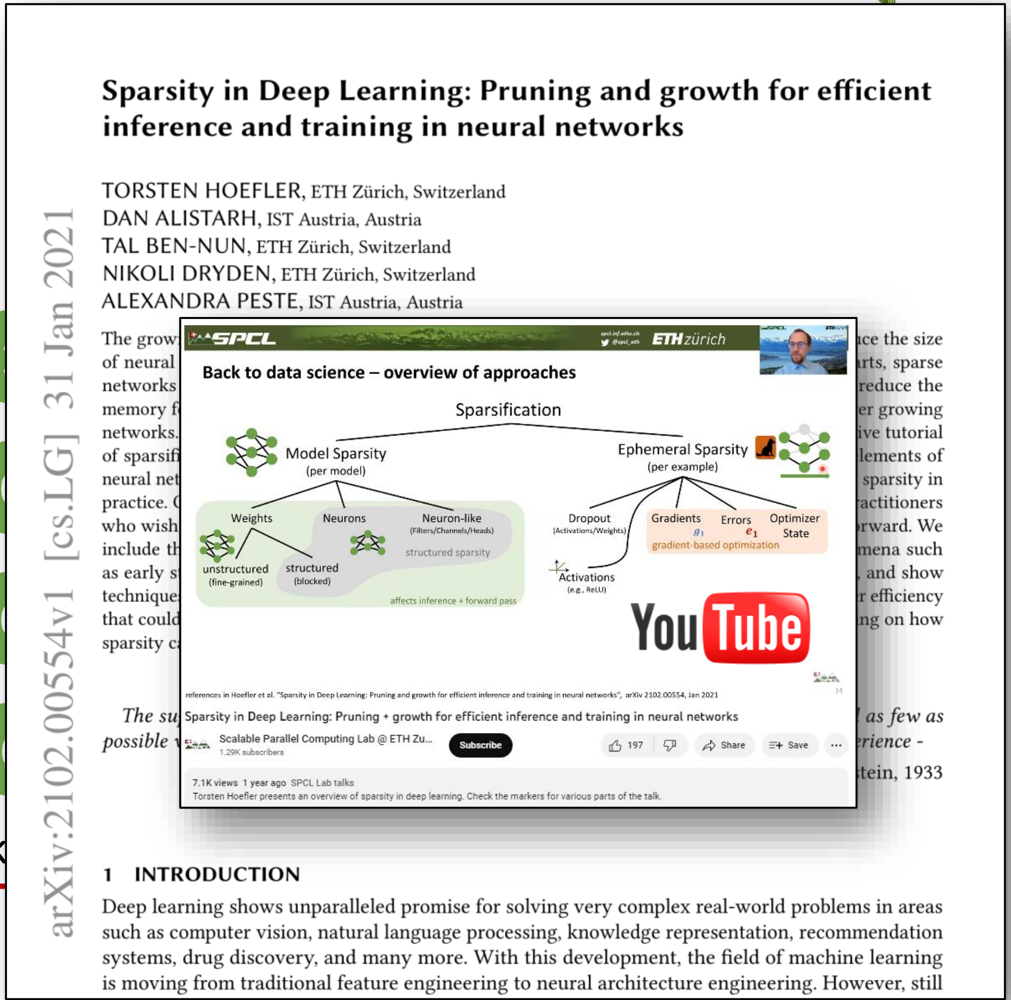
- ○ Represent as (sparse) vector space
- ✓ Less overfitting
- ✓ Interpretability
- ✓ Parsimony

the f_t_re wi_l b_ sp_rs_

Key results:

- 95% sparse ResNet-52, BERT, or GPT models
- Essentially same quality
- Up to 20x cheaper!





Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks

TORSTEN HOEFLER, ETH Zürich, Switzerland
DAN ALISTARH, IST Austria, Austria
TAL BEN-NUN, ETH Zürich, Switzerland
NIKOLI DRYDEN, ETH Zürich, Switzerland
ALEXANDRA PESTE, IST Austria, Austria

arXiv:2102.00554v1 [cs.LG] 31 Jan 2021

1 INTRODUCTION

Deep learning shows unparalleled promise for solving very complex real-world problems in areas such as computer vision, natural language processing, knowledge representation, recommendation systems, drug discovery, and many more. With this development, the field of machine learning is moving from traditional feature engineering to neural architecture engineering. However, still

The next step: Sparse-Quantized Representations - SpQR

SpQR: A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression

Tim Dettmers[*†] University of Washington
Ruslan Svirschevski[*] HSE University & Yandex
Vage Egiazarian[*] HSE University & Yandex
Denis Kuznedelev[*] Yandex & Skoltech
Elias Frantar IST Austria
Saleh Ashkboos ETH Zurich
Alexander Borzunov HSE University & Yandex
Torsten Hoefler ETH Zurich
Dan Alistarh IST Austria & NeuralMagic

published at ICLR’24

Abstract

Recent advances in large language model (LLM) pretraining have led to high-quality LLMs with impressive abilities. By compressing such LLMs via quantization to 3-4 bits per parameter, they can fit into memory-limited devices such as laptops and mobile phones, enabling personalized use. However, quantization down to 3-4 bits per parameter usually leads to moderate-to-high accuracy losses, especially for smaller models in the 1-10B parameter range, which are well-suited for edge deployments. To address this accuracy issue, we introduce the Sparse-Quantized Representation (SpQR), a new compressed format and quantization technique which enables for the first time *near-lossless* compression of LLMs across model scales, while reaching similar compression levels to previous methods. SpQR works by identifying and isolating *outlier weights*, which cause particularly-large quantization errors, and storing them in higher precision, while compressing all other weights to 3-4 bits, and achieves relative accuracy losses of less than 1% in perplexity for highly-accurate LLaMA and Falcon LLMs. This makes it possible to run 33B parameter LLM on a single 24 GB consumer GPU without any performance degradation at 15% speedup thus making powerful LLMs available to consumer without any downsides. SpQR comes with efficient algorithms for both encoding weights into its format, as well as decoding them efficiently at runtime[3]. Specifically, we provide an efficient GPU inference algorithm for SpQR which yields faster inference than 16-bit baselines at similar accuracy, while enabling memory compression gains of more than 4x.

Model Compression Enables

100x

More Efficient Processing

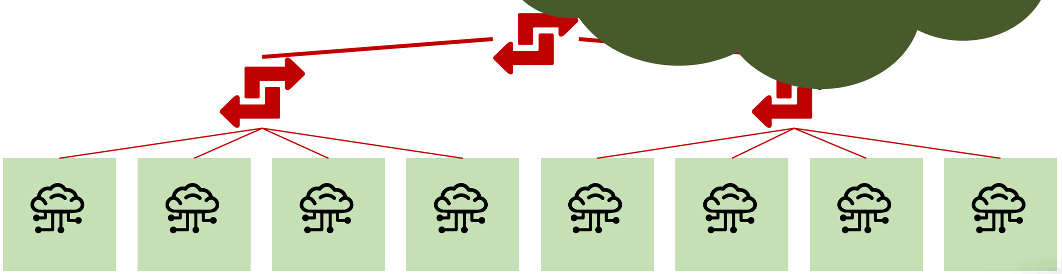
Which Makes Data Movement Even More Important!

Especially in the Network!

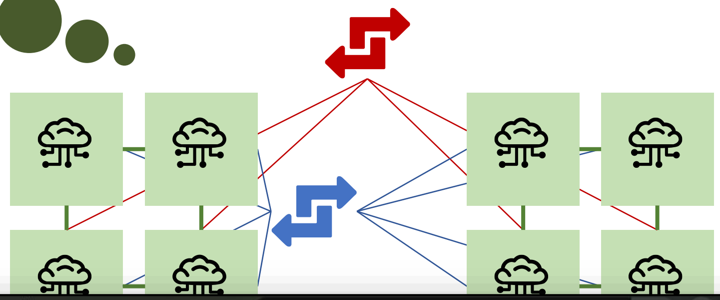
Bandwidth-cost-flexibility Tradeoffs (arXiv:2209.01346)

Global T
(e.g.

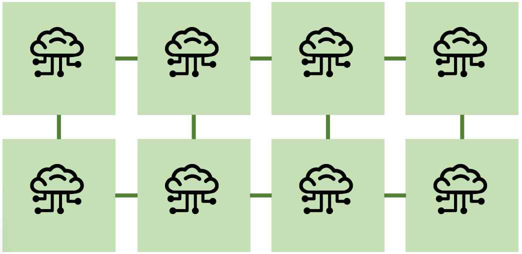
10-15x cheaper bandwidth than today's topologies!



HammingMesh
(many configurations)



Local Topology
(e.g., 2D Torus)



(large) reduce bandwidth

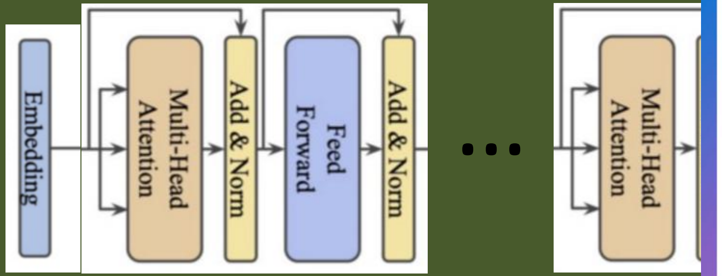
global bandwidth

placement flexibility

injection bandwidth

TH et. al.: HammingMesh: A Network Topology for Large-Scale Deep Learning, SC22 (best reproducibility award recipient) and arXiv (2209.01346)

Three Systems Dimensions in Large-scale Super-learning …



Altogether, we discussed a cost / performance improvement of

>1,000x

What now?

From LLMs to AHI

Ilya Sutskever

Pre-training as we know it will unquestionably end…because we have but one internet

2018 - BERT

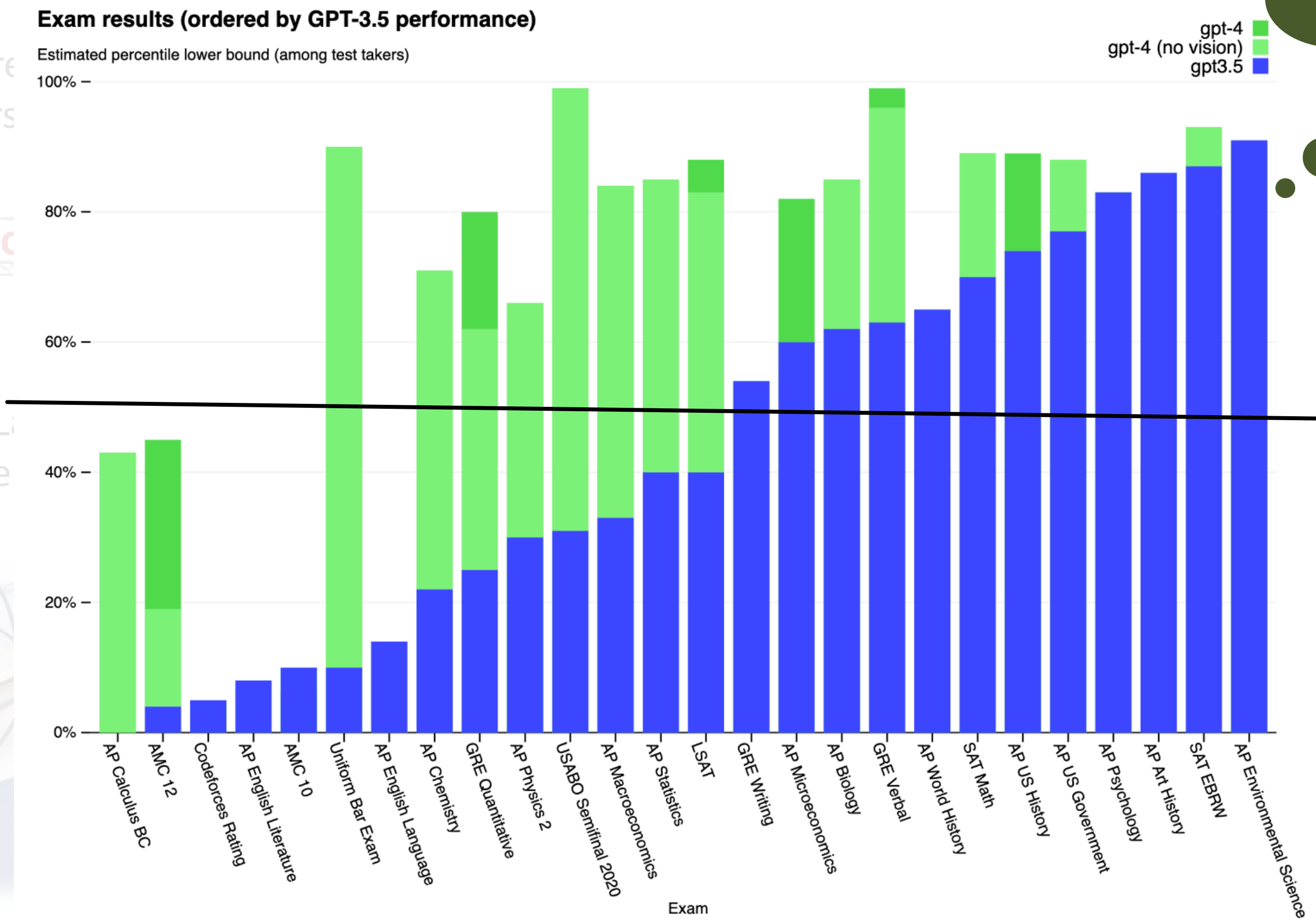
122k+ citations

2017 - Transformers

“Attention is All you Need”

146k+ citations

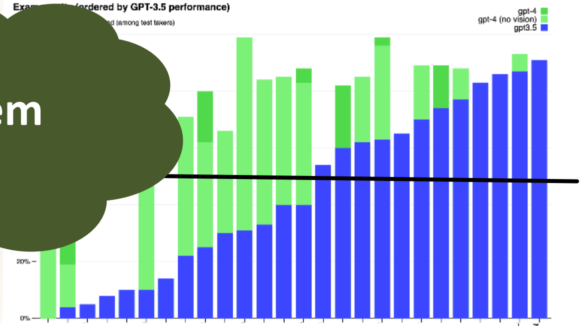
LLMs are a **great knowledge** base but **bad at reasoning**



From LLMs to AHI

Ilya Sutskever

Pre-training as we know it will unquestionably end…because we have but one internet



Let's teach them to reason!

"Let's proceed step by step" ☺

Sort the numbers "3, 2, 4, 5, 7, 12, 5, 6"

To sort "4, 6, 1, 8", I first split them into sets "4, 6" and "1, 8". Then I sort the sets and then I merge them sorted.

Sort the numbers "3, 2, 4, 5, 7, 12, 5, 6"

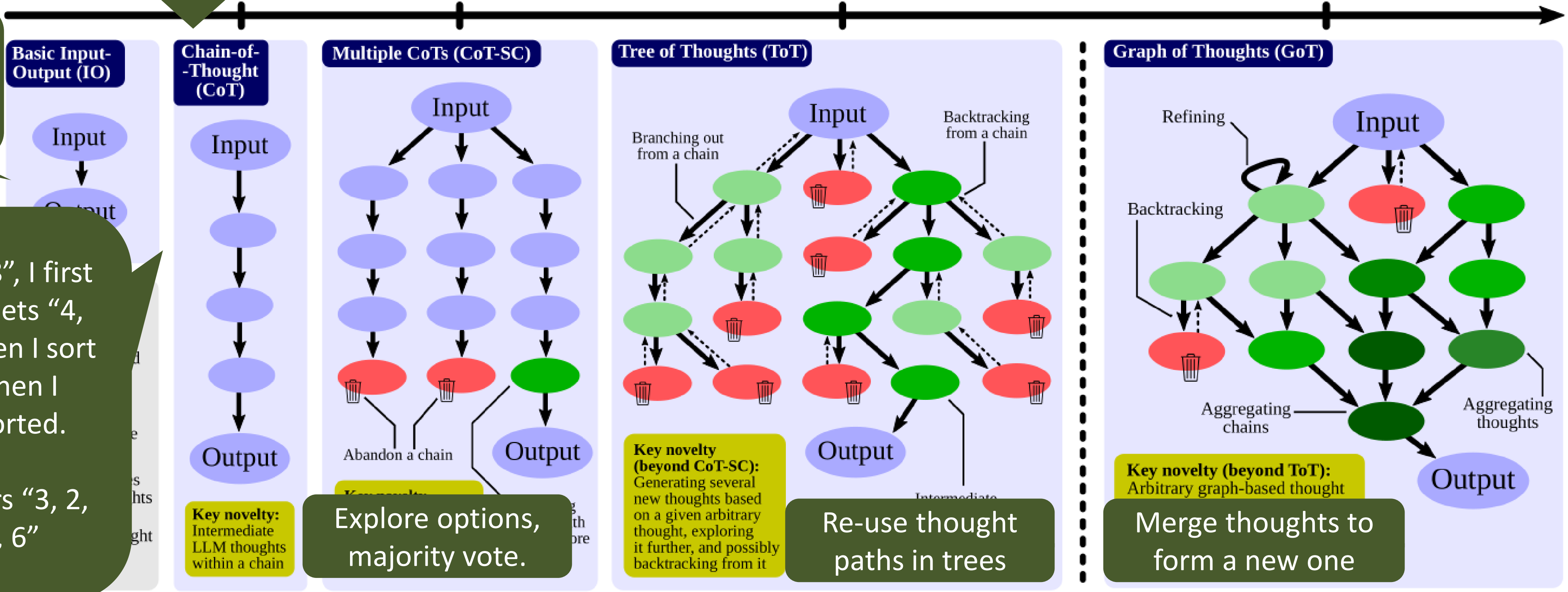
[Wang et al., March'22]

[Yao et al., May'23]
https://github.com/princeton-nlp/tree-of-thought-llm

[Long, May'23]
https://github.com/jieyilong/tree-of-thought-puzzle-solver

[Besta et al., August'23]
https://github.com/spcl/graph-of-thoughts

[Lei et al., August'23]



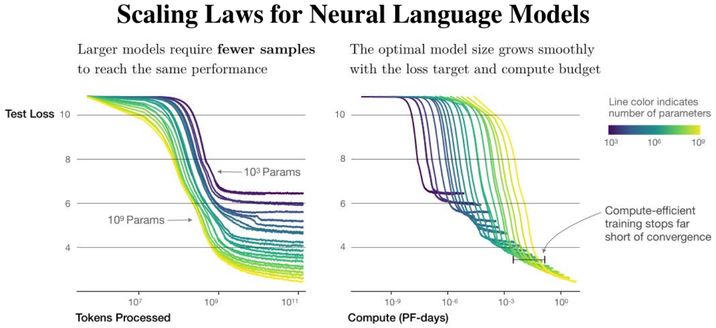
Explore options, majority vote.

Re-use thought paths in trees

Merge thoughts to form a new one

From LLMs to AHI

2018 - **BERT**
"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"
122k+ citations



2020 - **GPT-3** (2020, scaling laws)
"Language Models are Few-Shot Learners"
37k+ citations

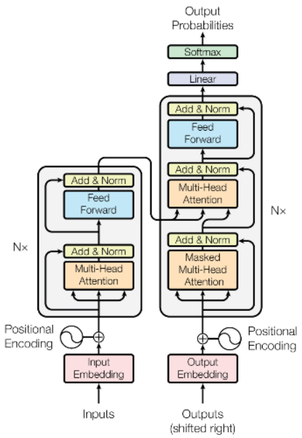
LLaMA by Meta

2023 – **Llama** (Qwen, Grok, etc.)
"LLaMA: Open and Efficient Foundation Language Models"
11k+ citations

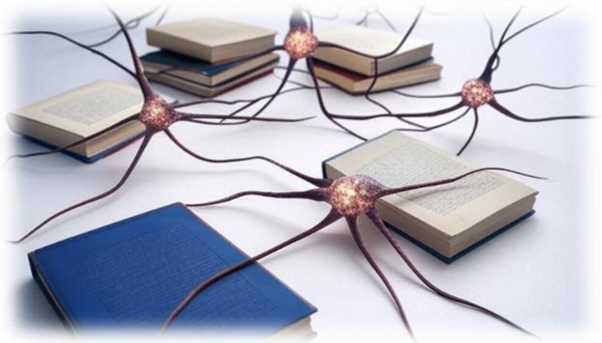
How to get to those "thoughts" (reasoning steps)

era of model size scaling — era of data scaling

2017 - **Transformers**
"Attention is All you Need"
146k+ citations



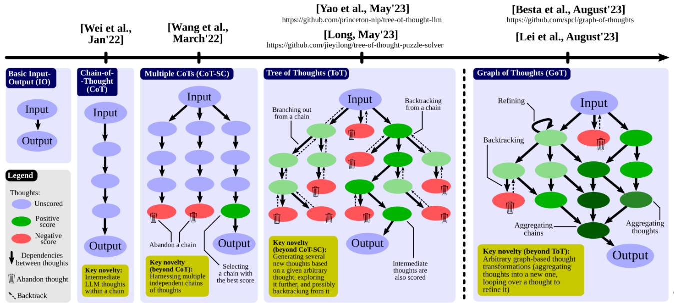
2019 - **GPT-2**
"Language Models are Unsupervised Multitask Learners"
14k+ citations



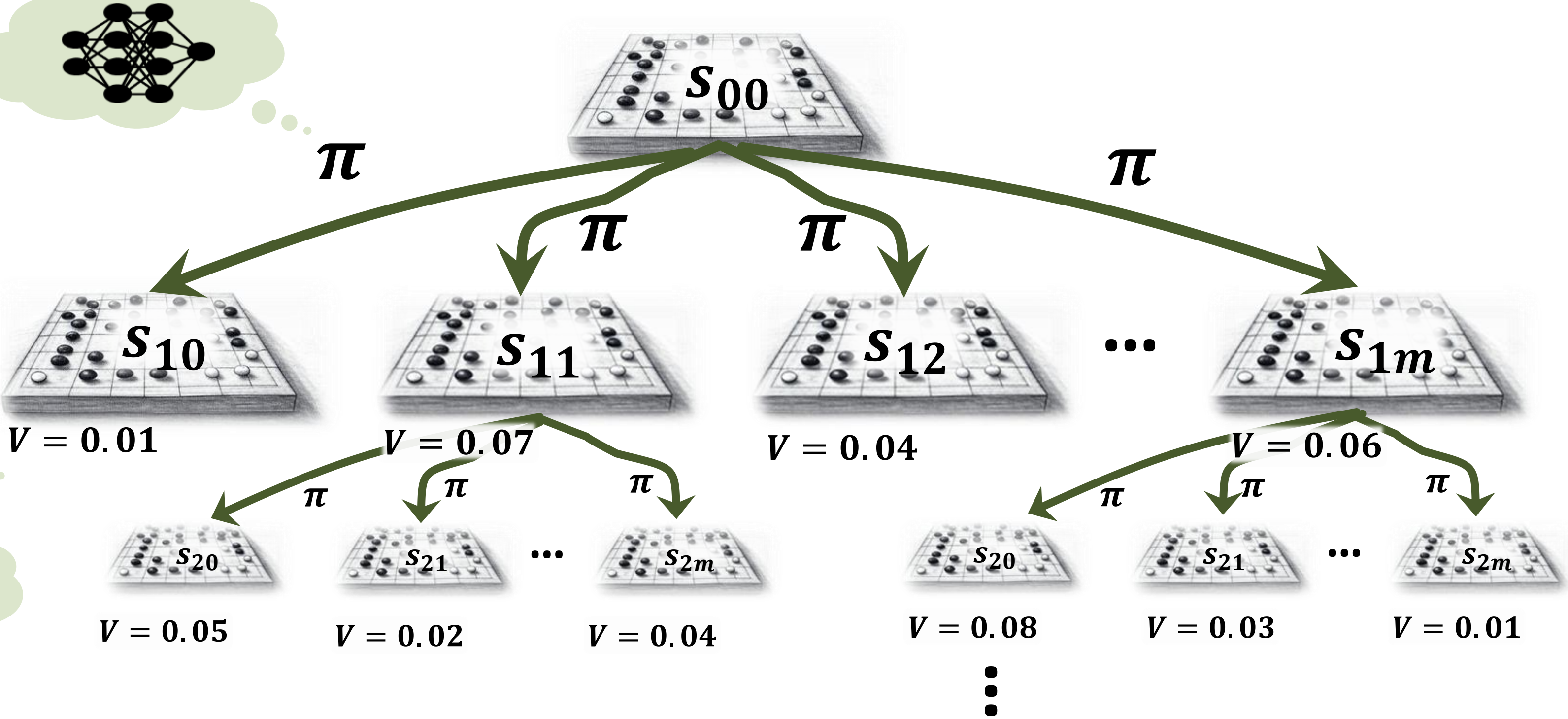
2022 – **ChatGPT** (RLHF, 2023, DPO)
"Training language models to follow instructions with human feedback"
14k+ citations

ChatGPT

2023 – **Chain of Thought**
Reasoning (SC-CoT, ToT, GoT, etc.)
"Chain-of-Thought Prompting Elicits Reasoning in Large Language Models"
8k+ citations

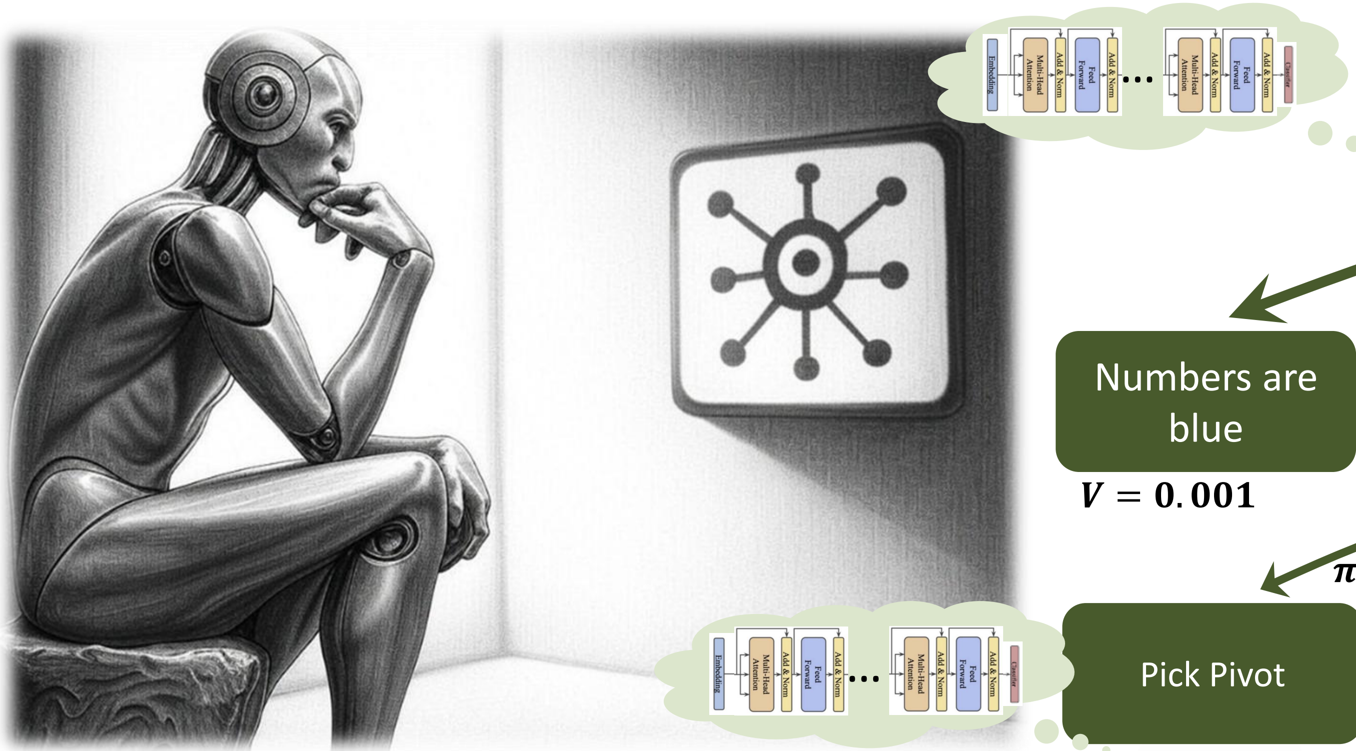


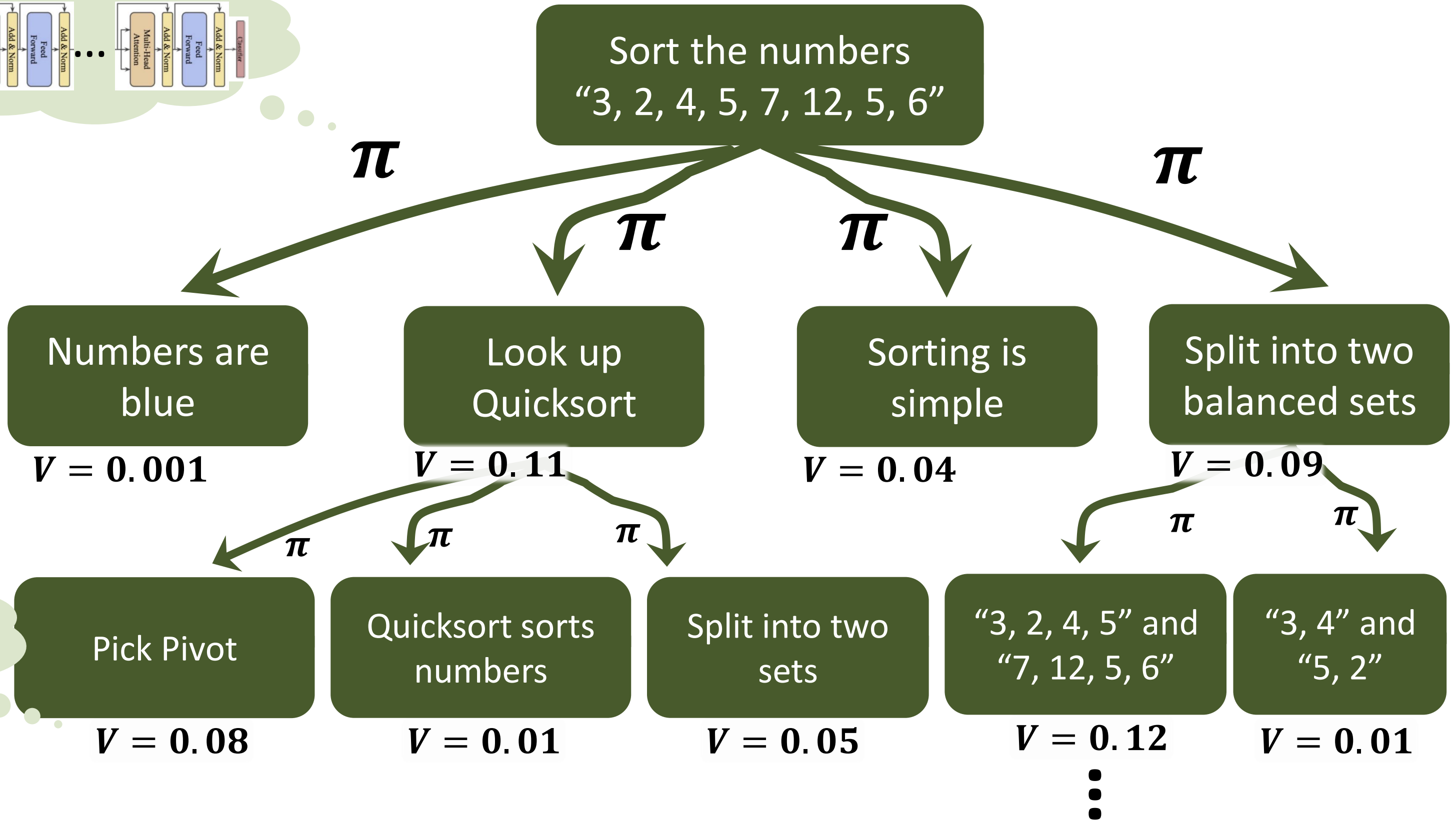
A Detour to Go Playing – AlphaGo vs. Lee Sedol (considered best Go player at the time)



- **(Monte Carlo) Tree Search (MCTS) samples multiple tree searches to some depth and propagates final values up the path, which keeps statistics for each state, action pair (edge)**
  - Up to 1,600 expansions per move for AlphaGo Zero
  - Depth is decided by the value network (no fixed depth rollout)
- **At the end, choose most promising action from root and prepare next move**

Unifying LLMs and Reinforcement Learning into Large Reasoning Models (LRMs)





- **Policy function is an LLM**
  - Fine-tuned with a special loss function to generate next best reasoning step (a bit tricky, needs multiple evals)
- **Value function is another LLM**
  - Replace final token output layer with a regression to a value (train on known examples, e.g., math tasks)
- **During inference, still do MCTS search to cover reasoning paths**
  - Extremely expensive! Up to thousands of inferences per reasoning step!

With RLMs to AHI

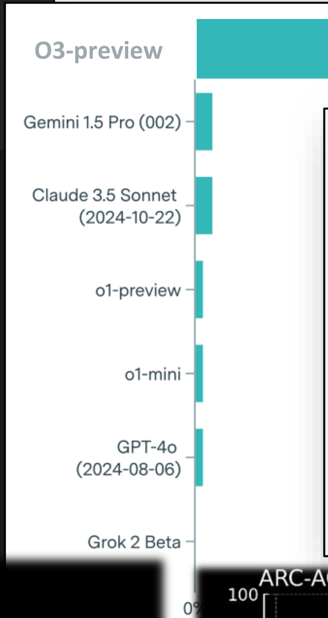
GPQA: A Graduate-Level Google-Proof Q&A Benchmark

David Rei... kland[1]
Jackson P... Dirani[1]

Human PhDs:
34% outside their field
81% inside their field
o3:
87% in all fields

We present GPQA, a challenging dataset of 448 multiple-choice questions written by domain experts in biology, physics, and chemistry. We ensure that the questions are high-quality and extremely difficult: experts who have or are pursuing PhDs in the corresponding domains reach 65% accuracy (74% when discounting clear mistakes the experts identified in retrospect), while highly skilled non-expert validators only

FRONTIERMATH: A BENCHMARK FOR EVALUATING ADVANCED MATHEMATICAL REASONING IN AI

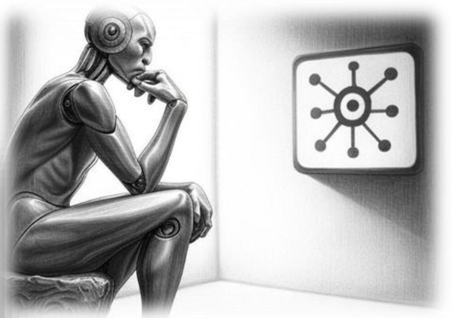


IMPRESSIONS OF OUR RESEARCH-LEVEL PROBLEMS

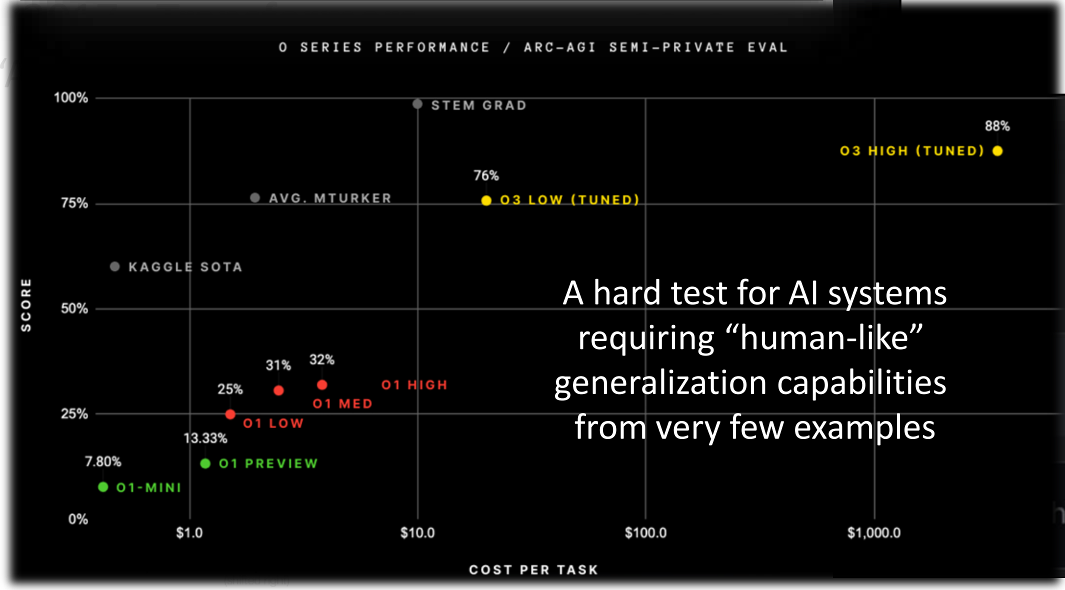
"These are extremely challenging... I think they will resist AIs for several years at least."

Nov.'24

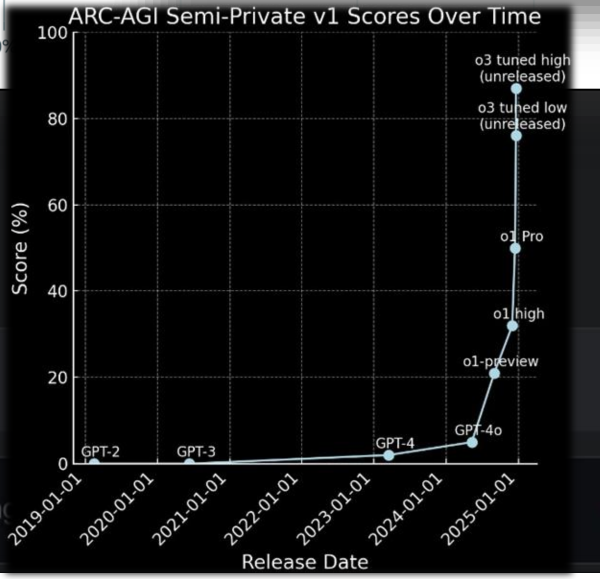
Terence Tao
Fields Medalist (2



2024 – **Strawberry**
RL (o1, o3, etc.)
"Learning to Reason with LLMs"



A hard test for AI systems requiring "human-like" generalization capabilities from very few examples

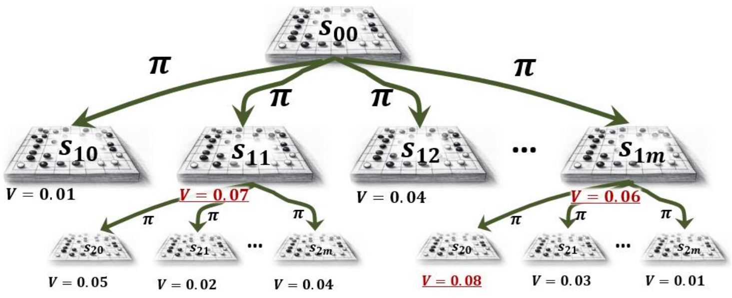


Codeforce Elo rating

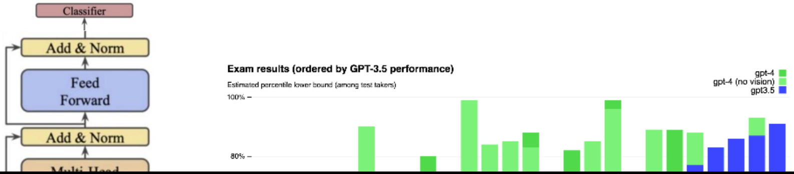
Elo-MMR	Title	Division	Number	Percentile	CF at same rank (spread)
3000+	Legendary Grandmaster	1	8	99.99	3382+
2700-2999	International Grandmaster	1	37	99.95	3010-3329 (372)
2400-2699	Grandmaster	1	255	99.7	2565-3010 (445)
2200-2399	International Master	1	560	99.1	2317-2565 (248)
2000-2199	Master	1	2089	97	2088-2317 (229)
1800-1999	Candidate Master	2			
1600-1799	Expert				
1400-1599	Special				
1200-1399	Apprentic				
1000-1199	Pupil				
Up to 999	Newbie	4	33923		Up to 818

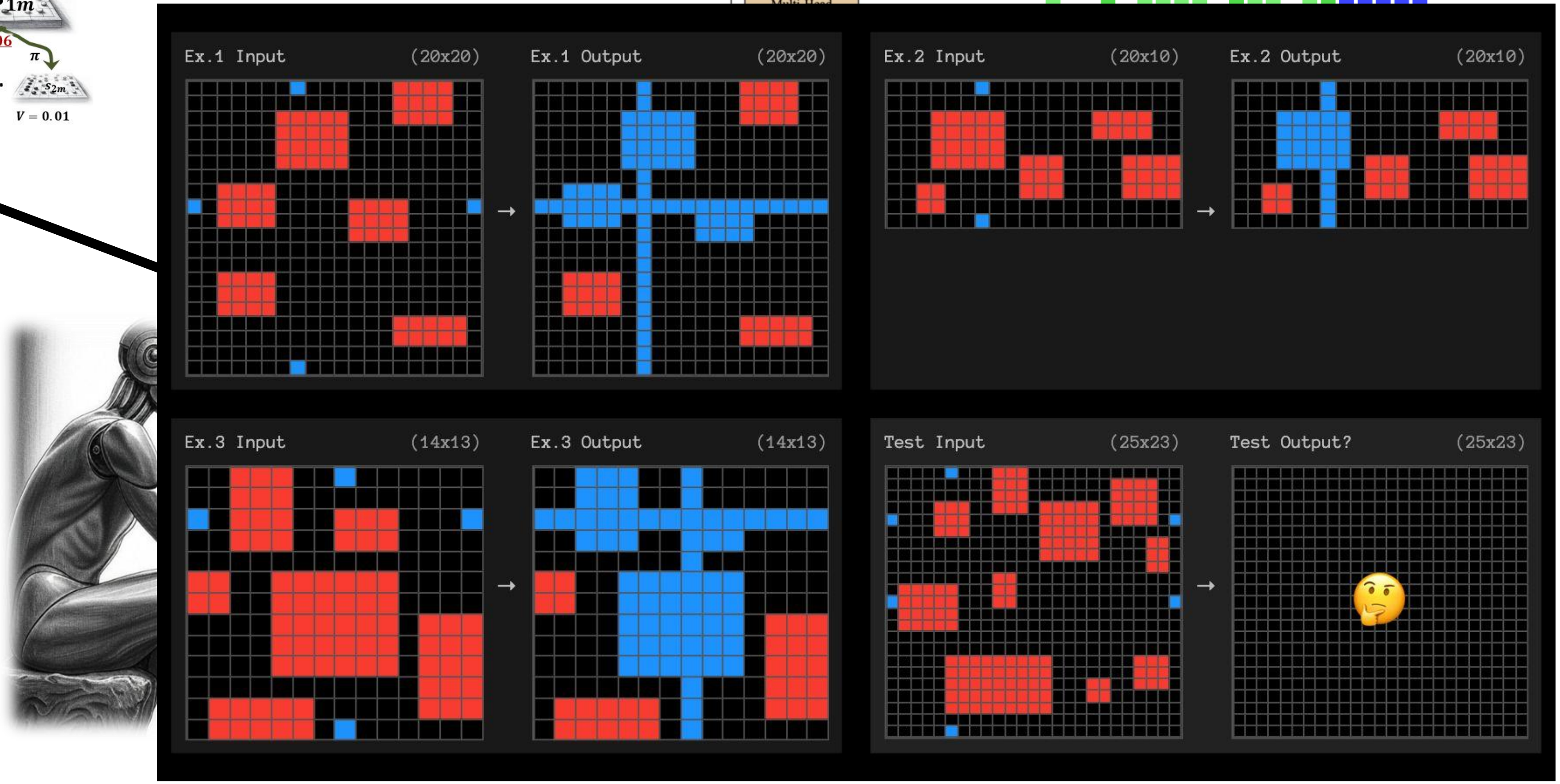
o3 achieves 2727 → 99.95^{th} percentile of competitive programmers!

Super-human Strategy



Super-human Knowledge





Reasoning Language Models (RLMs) start the

era of reasoning scaling

Chollet: **Calling something like o1 "an LLM" is about as accurate as calling AlphaGo "a convnet"**

We are NOT done yet!

If you want to know more how this works or want to build one yourself!

1

Reasoning Language Models: A Blueprint

Maciej Besta[1†], Julia Barth[1], Eric Schreiber[1], Ales Kubicek[1], Afonso Catarino[1], Robert Gerstenberger[1], Piotr Nyczyk[2], Patrick Iff[1], Yueling Li[3], Sam Houliston[1], Tomasz Sternal[1], Marcin Copik[1], Grzegorz Kwaśniewski[1], Jürgen Müller[3], Łukasz Flis[4], Hannes Eberhard[1], Hubert Niewiadomski[2], Torsten Hoefler[1]

† Corresponding author [1]ETH Zurich [2]Cledar [3]BASF SE [4]Cyfronet AGH

Abstract—Reasoning language models (RLMs), also known as Large Reasoning Models (LRMs), such as OpenAI's o1 and o3, DeepSeek-V3, and Alibaba's QwQ, have redefined AI's problem-solving capabilities by extending large language models (LLMs) with advanced reasoning mechanisms. Yet, their high costs, proprietary nature, and complex architectures—uniquely combining Reinforcement Learning (RL), search heuristics, and LLMs—present accessibility and scalability challenges. To address these, we propose a comprehensive blueprint that organizes RLM components into a modular framework, based on a survey and analysis of all RLM works. This blueprint incorporates diverse reasoning structures (chains, trees, graphs, and nested forms), reasoning strategies (e.g., Monte Carlo Tree Search, Beam Search), RL concepts (policy, value models and others), supervision schemes (Outcome-Based and Process-Based Supervision), and other related concepts (e.g., Test-Time Compute, Retrieval-Augmented Generation, agent tools). We also provide detailed mathematical formulations and algorithmic specifications to simplify RLM implementation. By showing how schemes like LLaMA-Berry, QwQ, Journey Learning, and Graph of Thoughts fit as special cases, we demonstrate the blueprint's versatility and unifying potential. To illustrate its utility, we introduce **x1**, a modular implementation for rapid RLM prototyping and experimentation. Using **x1** and a literature review, we provide key insights, such as multi-phase training for policy and value models, and the importance of familiar training distributions. Finally, we discuss scalable RLM cloud deployments and we outline how RLMs can integrate with a broader LLM ecosystem. Our work demystifies RLM c… fosters innovation, aiming to mitigate the gap between "rich AI" and "p…

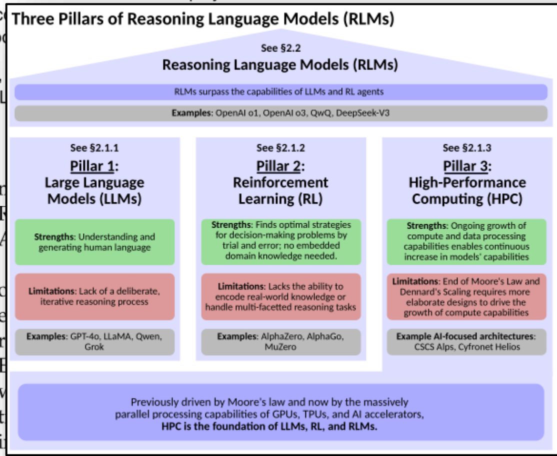
Index Terms—Reasoning Language Model, Large Reasoning Model, … LRM, Reasoning LLMs, Reinforcement Learning for LLMs, MCTS for L…

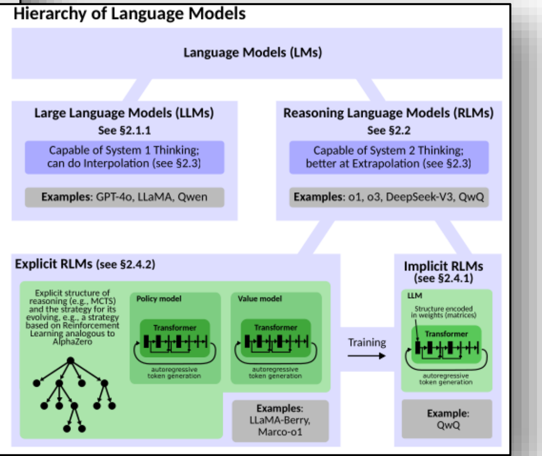
arXiv:2501.11223v3 [cs.AI] 23 Jan 2025

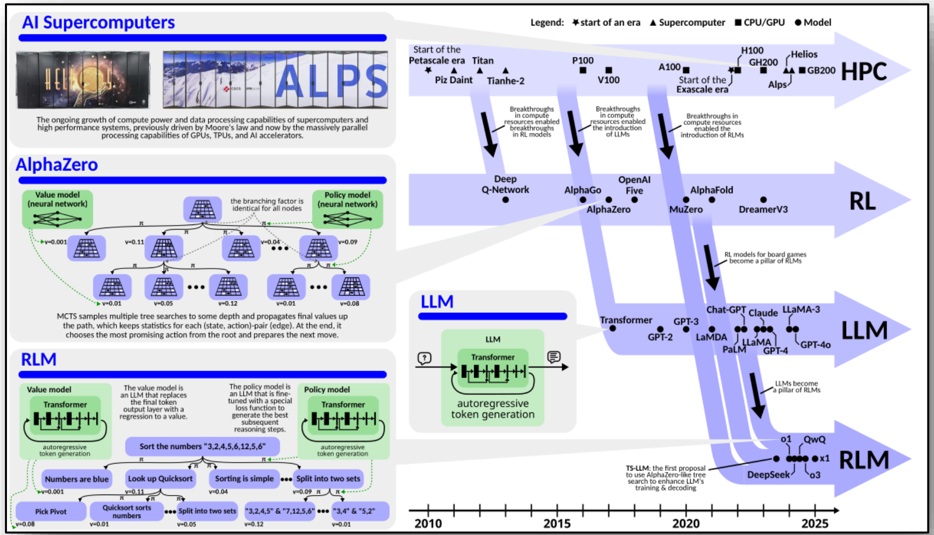
1 Introduction

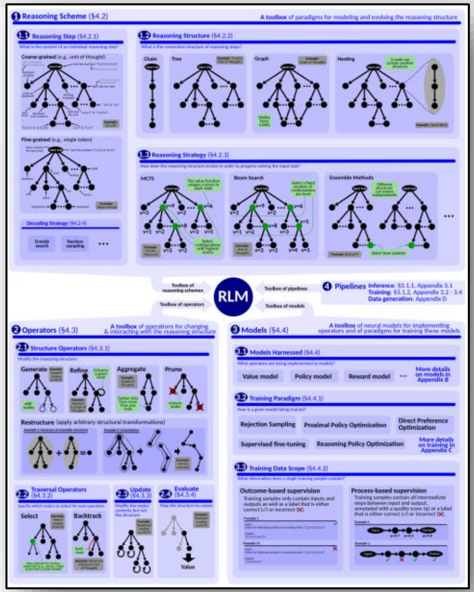
Reasoning Language Models (RLMs), such as OpenAI's o1 [116], o3 [76], and Alibaba's QwQ [148], also referred to as Large Reasoning Models (LRMs)[1], represent a transformative breakthrough in AI, on par with the advent of ChatGPT [114]. These advanced systems have fundamentally redefined AI's problem-solving capabilities, enabling nuanced reasoning, improved contextual understanding, and robust decision-making across a wide array of domains, reshaping science [45], industries [21], governance [52], and numerous other aspects of human life [46], [75], [80], [143], [144]. By extending the capabilities of standard large language

… growing disadvantage, threatening to stifle innovation and reinforce systemic inequities. As RLMs become integral to



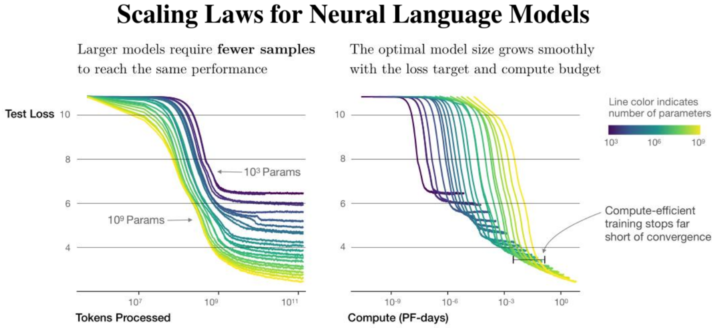






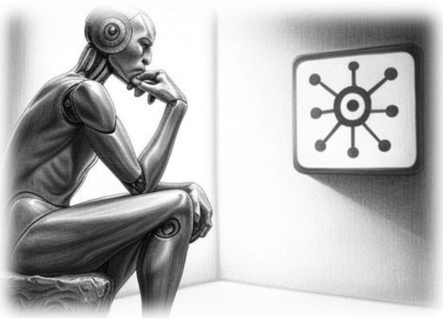
With RLMs to AHI

2018 - **BERT**
"BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding"
122k+ citations



2020 - **GPT-3** (2020, scaling laws)
"Language Models are Few-Shot Learners"
37k+ citations

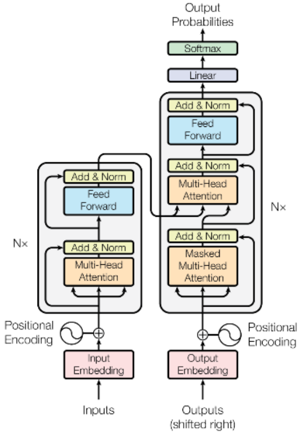
2023 – **Llama** (Qwen, Grok, etc.)
"LLaMA: Open and Efficient Foundation Language Models"
11k+ citations



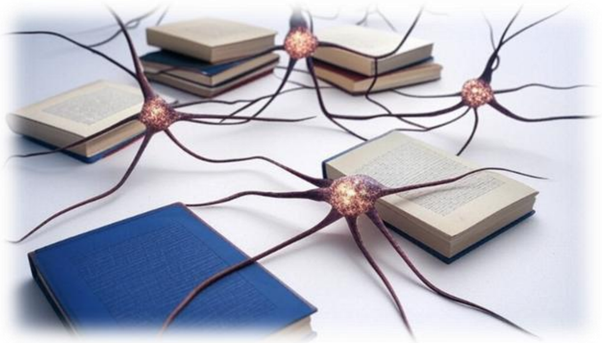
2024 – **Strawberry**
RL (o1, o3, etc.)
"Learning to Reason with LLMs"

era of model size scaling — **era of data scaling** — **era of reasoning scaling**

2017 - **Transformers**
"Attention is All you Need"
146k+ citations



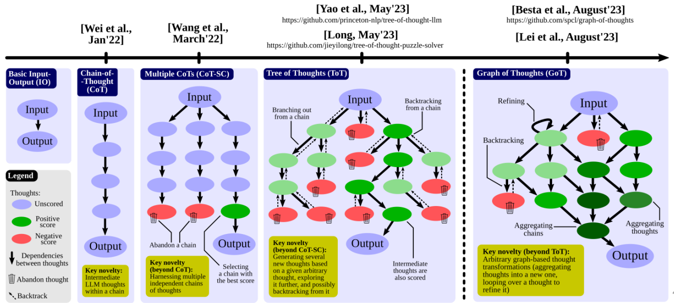
2019 - **GPT-2**
"Language Models are Unsupervised Multitask Learners"
14k+ citations



2022 – **ChatGPT** (RLHF, 2023, DPO)
"Training language models to follow instructions with human feedback"
14k+ citations

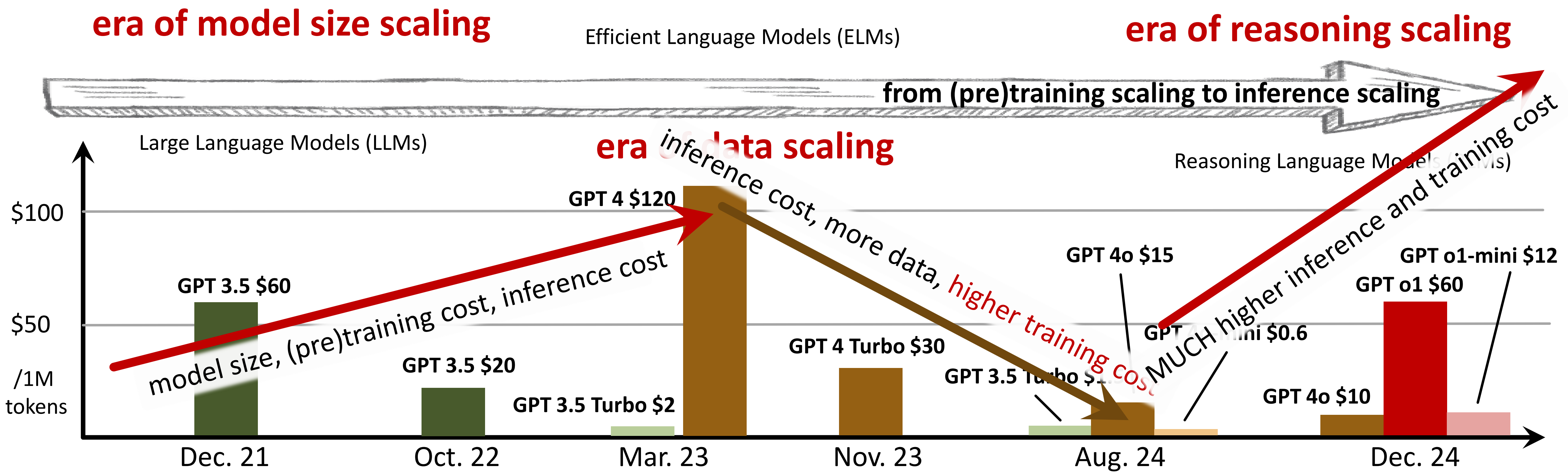
ChatGPT

2023 – **Chain of Thought**
Reasoning (SC-CoT, ToT, GoT, etc.)
"Chain-of-Thought Prompting Elicits Reasoning in Large Language Models"
8k+ citations

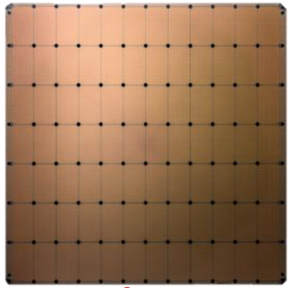


Development of Computation Requirement with RLMs

o3 > $5000 / task



We need cheaper compute



Principles: Datatypes, Sparsity, Spatial

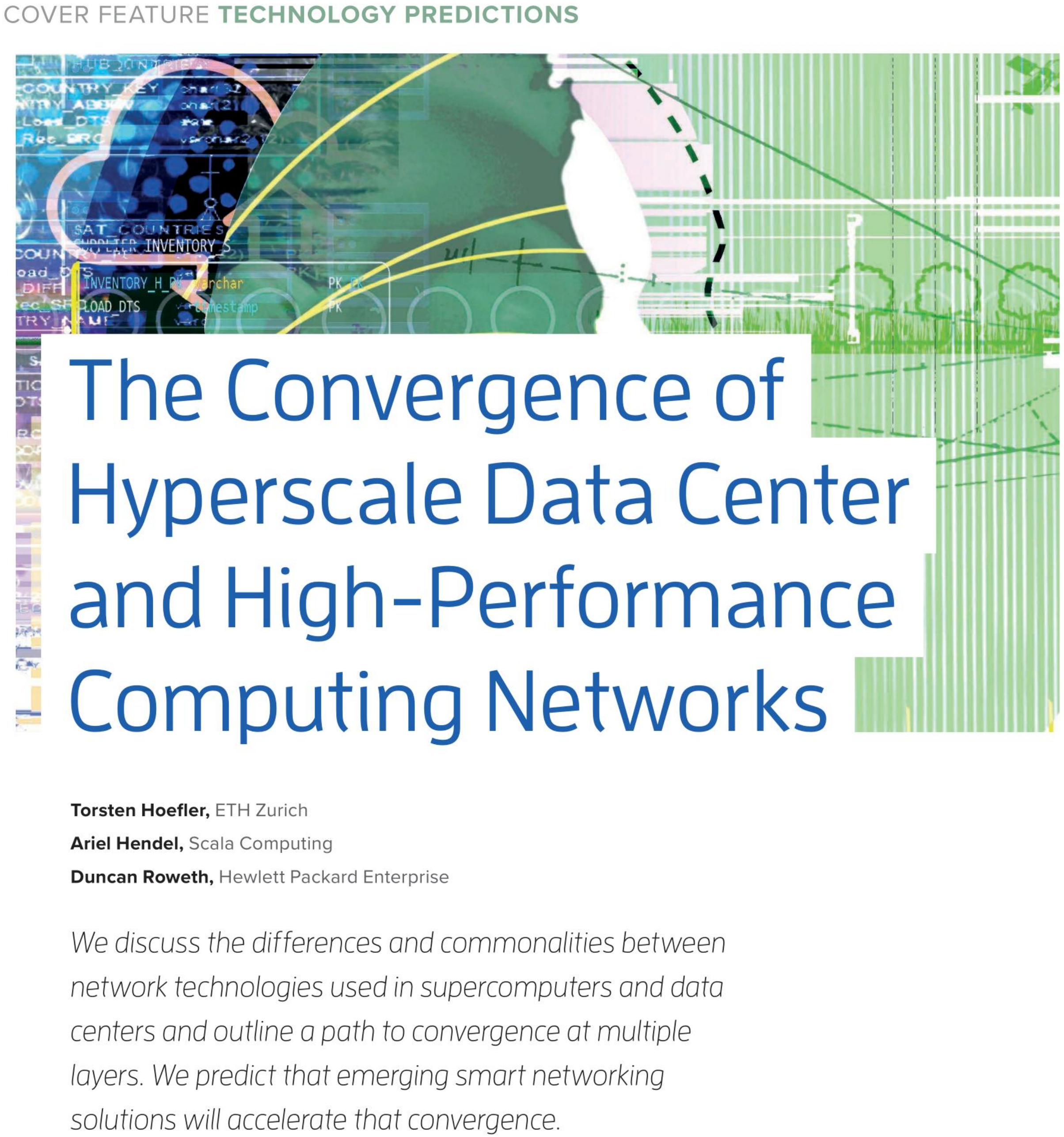
We need cheaper systems (networking!)

Ultra Ethernet Consortium

Principles: high local bandwidth, reliability, cost

Networks Converge

The Datacenter will be a Supercomputer

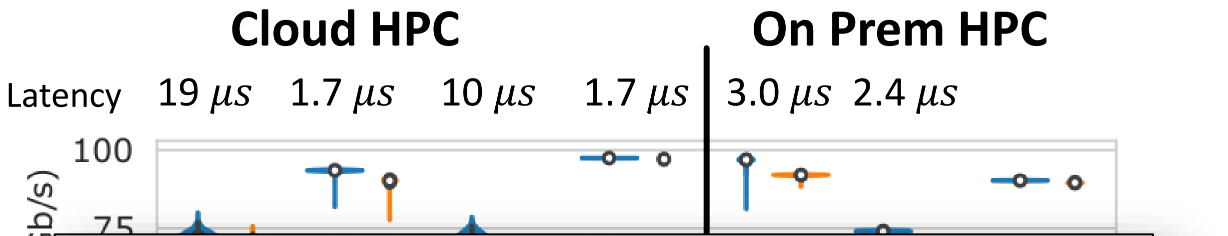


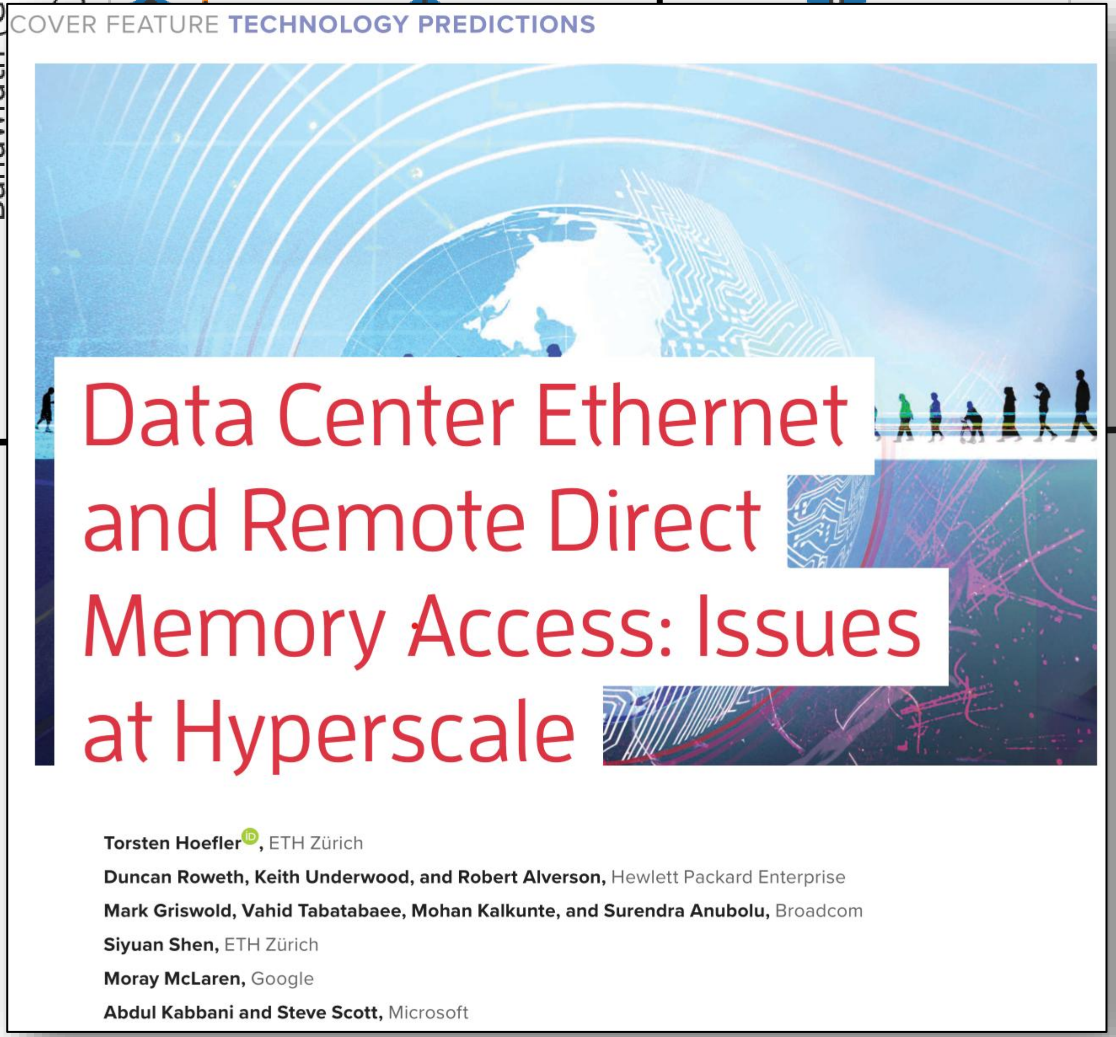
COVER FEATURE **TECHNOLOGY PREDICTIONS**

The Convergence of Hyperscale Data Center and High-Performance Computing Networks

Torsten Hoefler, ETH Zurich
Ariel Hendel, Scala Computing
Duncan Roweth, Hewlett Packard Enterprise

We discuss the differences and commonalities between network technologies used in supercomputers and data centers and outline a path to convergence at multiple layers. We predict that emerging smart networking solutions will accelerate that convergence.





COVER FEATURE **TECHNOLOGY PREDICTIONS**

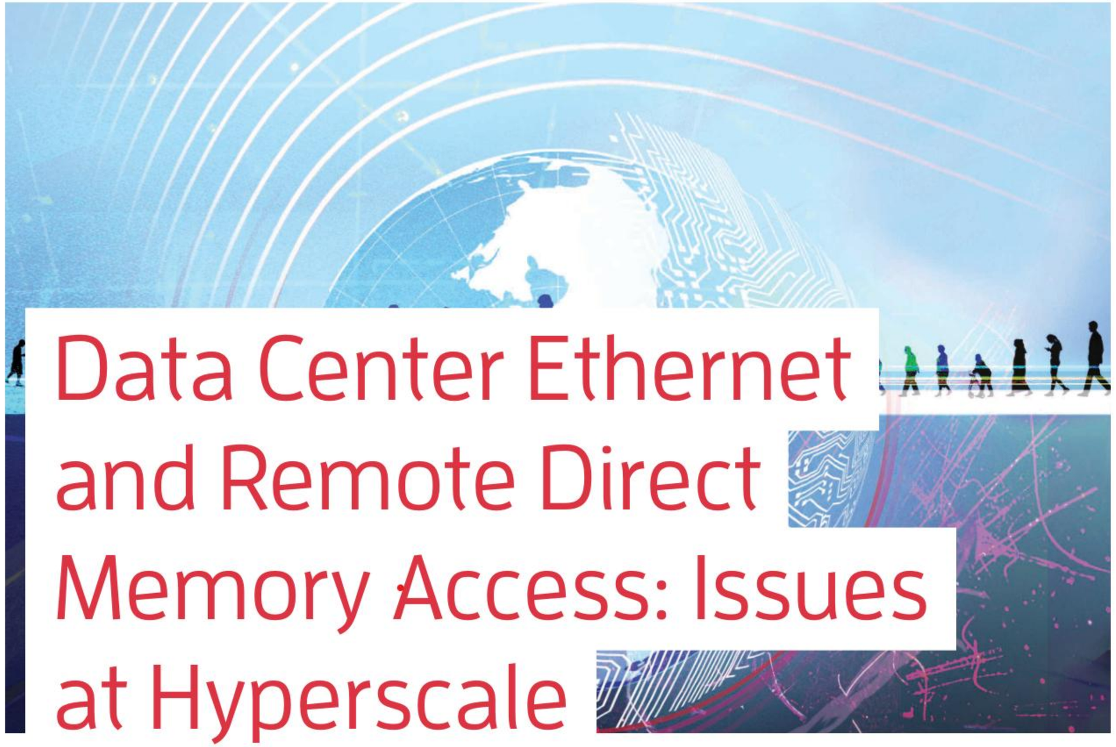
Data Center Ethernet and Remote Direct Memory Access: Issues at Hyperscale

Torsten Hoefler, ETH Zürich
Duncan Roweth, Keith Underwood, and Robert Alverson, Hewlett Packard Enterprise
Mark Griswold, Vahid Tabatabaee, Mohan Kalkunte, and Surendra Anubolu, Broadcom
Siyuan Shen, ETH Zürich
Moray McLaren, Google
Abdul Kabbani and Steve Scott, Microsoft

[1] De Sensi et al.: "Noise in the Clouds: Influence of Network Performance Variability on Application Scalability", SIGMETRICS'23

Ultra Ethernet Set Out to Create the Best AI/ML and HPC Interconnect!

COVER FEATURE **TECHNOLOGY PREDICTIONS**



Data Center Ethernet and Remote Direct Memory Access: Issues at Hyperscale

Torsten Hoefler, ETH Zürich
Duncan Roweth, Keith Underwood, and Robert Alverson, Hewlett Packard Enterprise
Mark Griswold, Vahid Tabatabaee, Mohan Kalkunte, and Surendra Anubolu, Broadcom
Siyuan Shen, ETH Zürich
Moray McLaren, Google
Abdul Kabbani and Steve Scott, Microsoft

Ultra Ethernet Consortium

Founding Members

AMD, ARISTA, BROADCOM, CISCO, EVIDEN an atos business, Hewlett Packard Enterprise, intel, Meta, Microsoft

Ultra Ethernet Consortium

white Paper on ultraethernet.org

Overview of and Motivation for the Forthcoming Ultra Ethernet Consortium Specification

Networking Demands of Modern AI Jobs

Networking is increasingly important for efficient and cost-effective training of AI models. Large Language Models (LLMs) such as GPT-3, Chinchilla, and PALM, as well as recommendation systems like DLRM and DHEN, are trained on clusters of thousands of GPUs.

Ecosystem is quicky growing

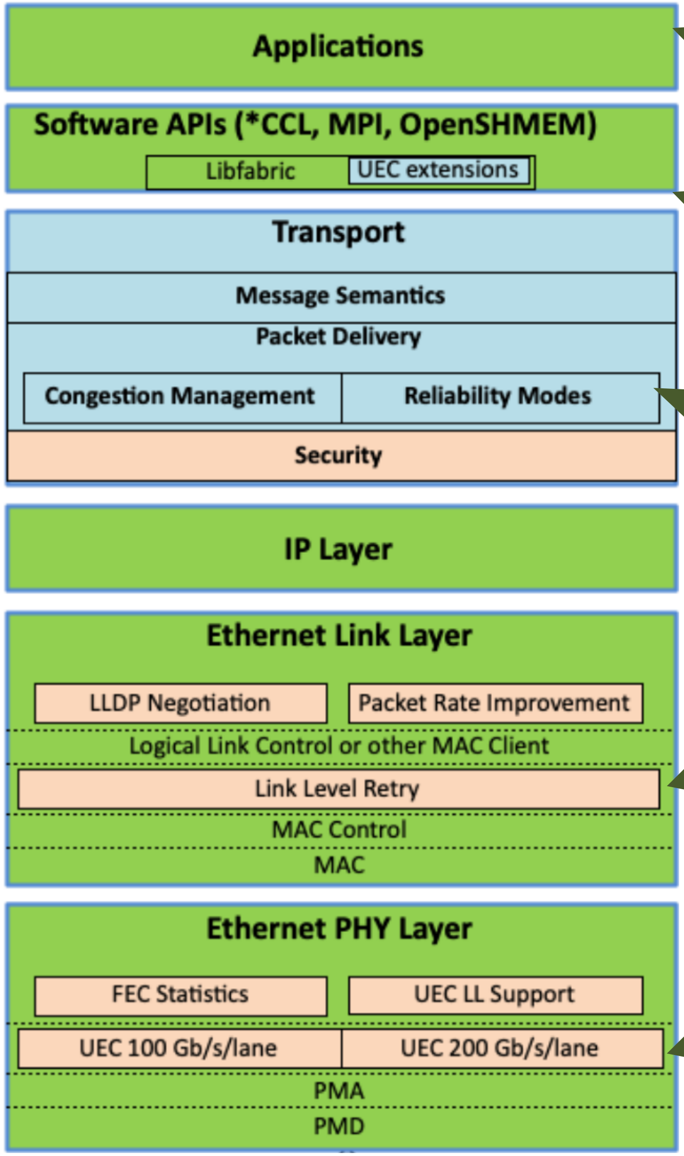
Ultra Ethernet Consortium

Today 10 steering companies, 26 general member companies, 54 contributor members

Chair’s view of the Transport WG Meeting in March’24 (60+ members on site, 800+ total now)

Ultra Ethernet's philosophy

Ultra Ethernet Consortium



(largely) unchanged libfabric applications

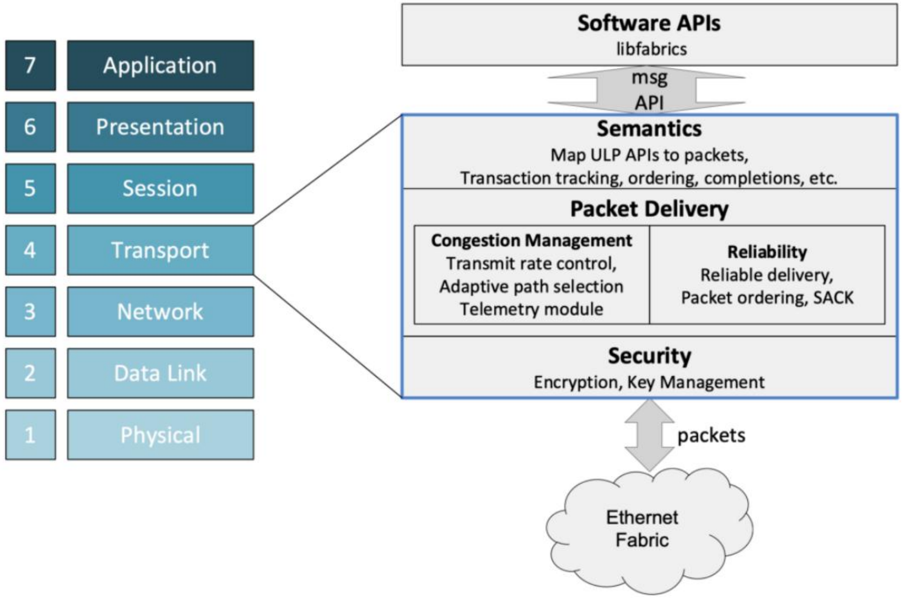
Libfabric + UE extensions

Here is where most of the innovation lies!

Some Ethernet extensions to benefit AI/HPC workloads

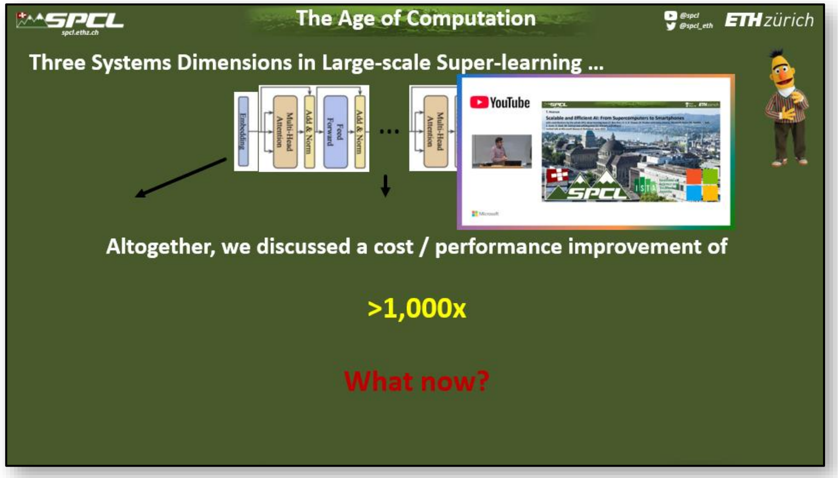
Close collaboration with IEEE 802.3 standards body

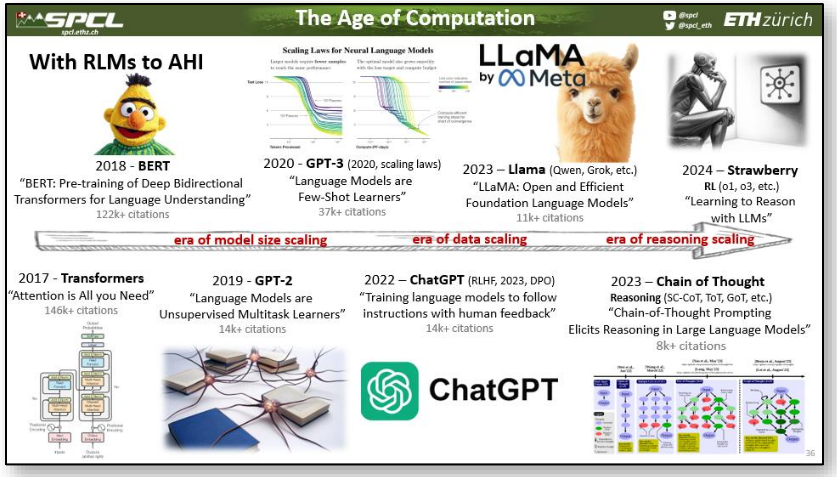
UE enables cheap high-performance **hardware implementations** of an **optimized transport** over (legacy) **Ethernet networks** while **enabling vendor innovation**

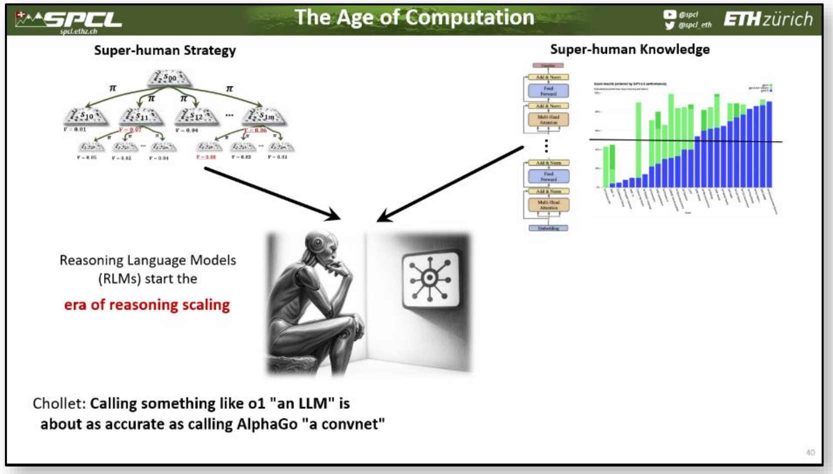


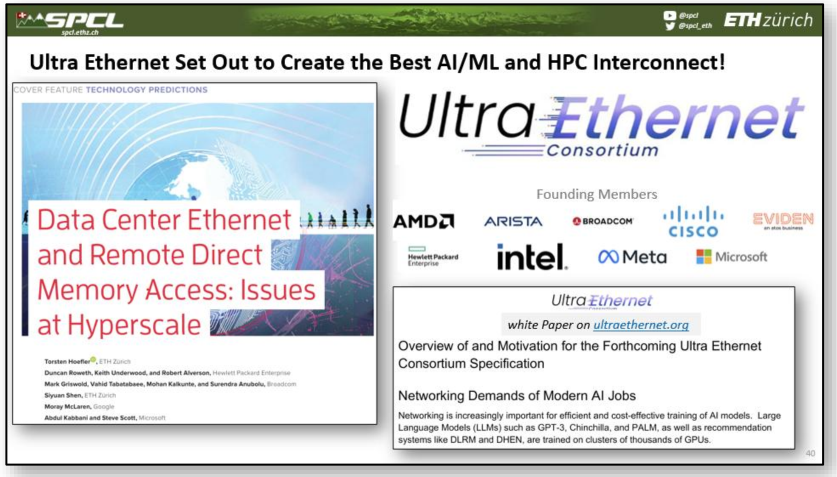
https://ultraethernet.org/uec-progresses-towards-v1-0-set-of-specifications/

Key Points and Conclusions









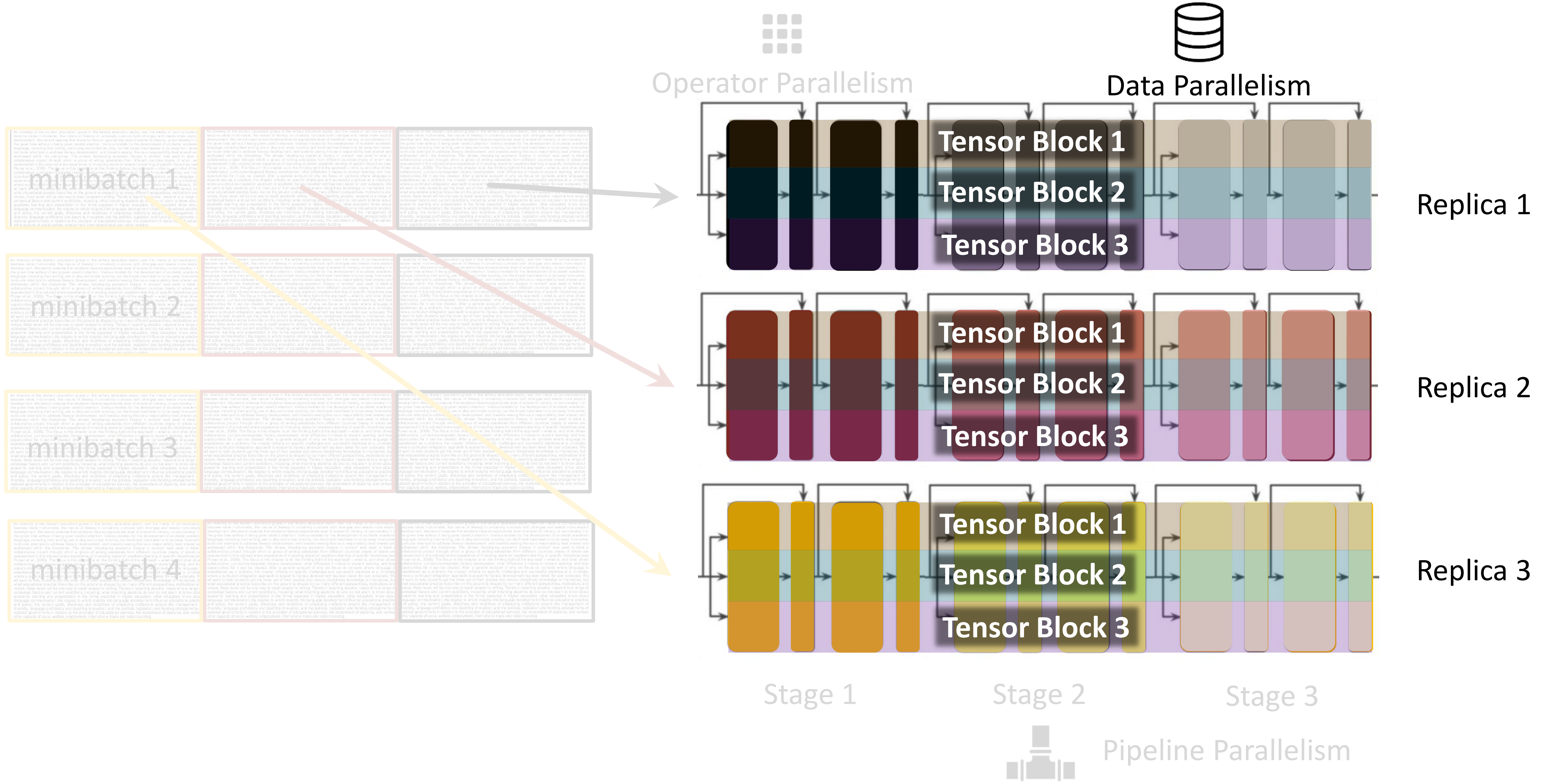
More of SPCL's research:

- youtube.com/@spcl — 210+ Talks
- twitter.com/spcl_eth — 1.4K+ Followers
- github.com/spcl — 2K+ Stars

… or spcl.ethz.ch

Want to join our efforts?
We're looking for excellent
Postdocs, PhD students, and Visitors.
Talk to me!

The Three Dimensions of Parallelism in Deep Learning (arXiv:1802.09941)

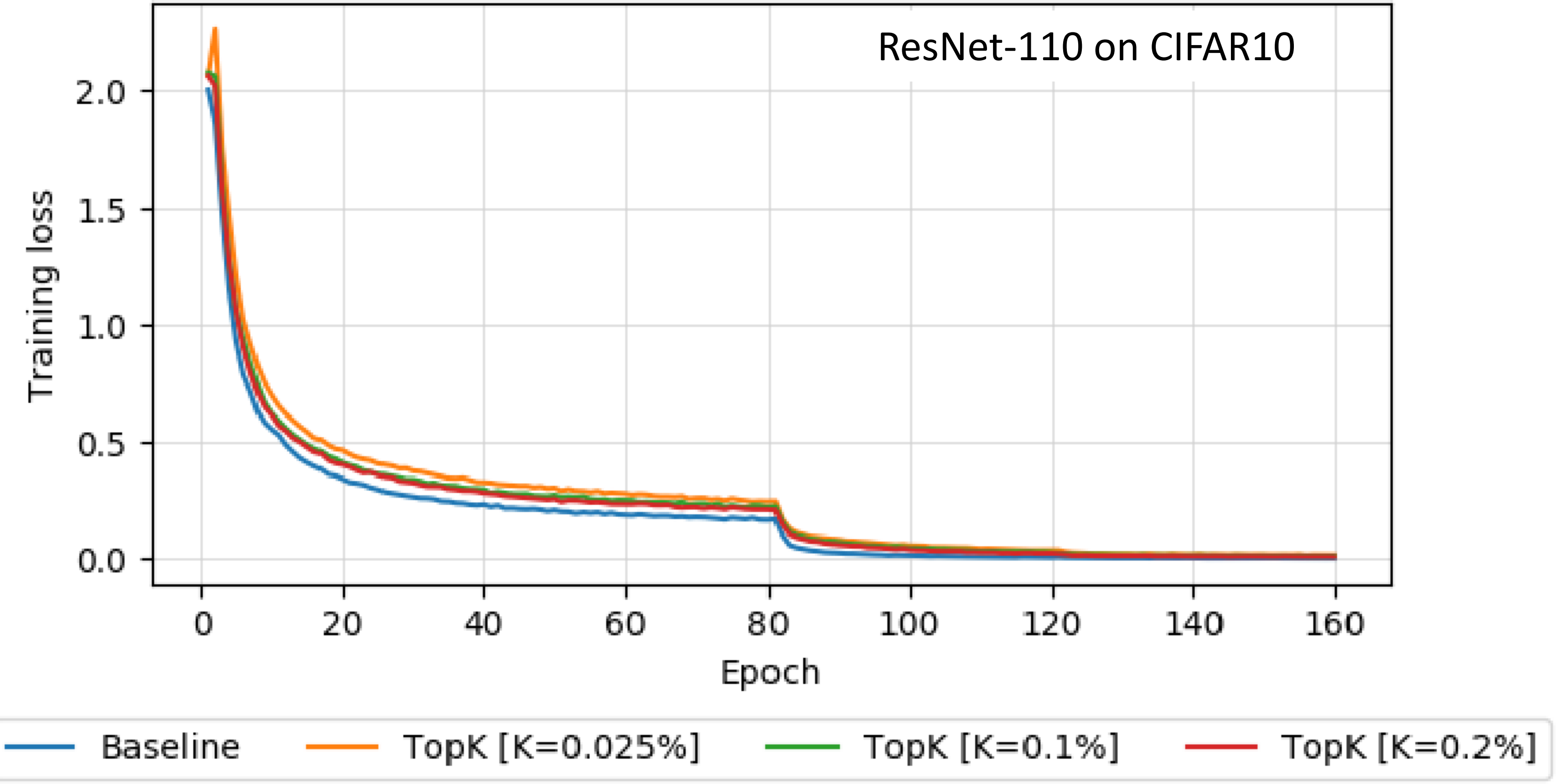


Data-parallel Gradient Sparsification – Top-k SGD (arXiv:1809.10505)

- **Turns out 90-99.9% of the smallest gradient values can be skipped in the summation – at similar accuracy**
  - **Accumulate** the **skipped values locally** (convergence proof, similar to async. SGD with implicit staleness bounds [1])

Assump

Discuss
Section

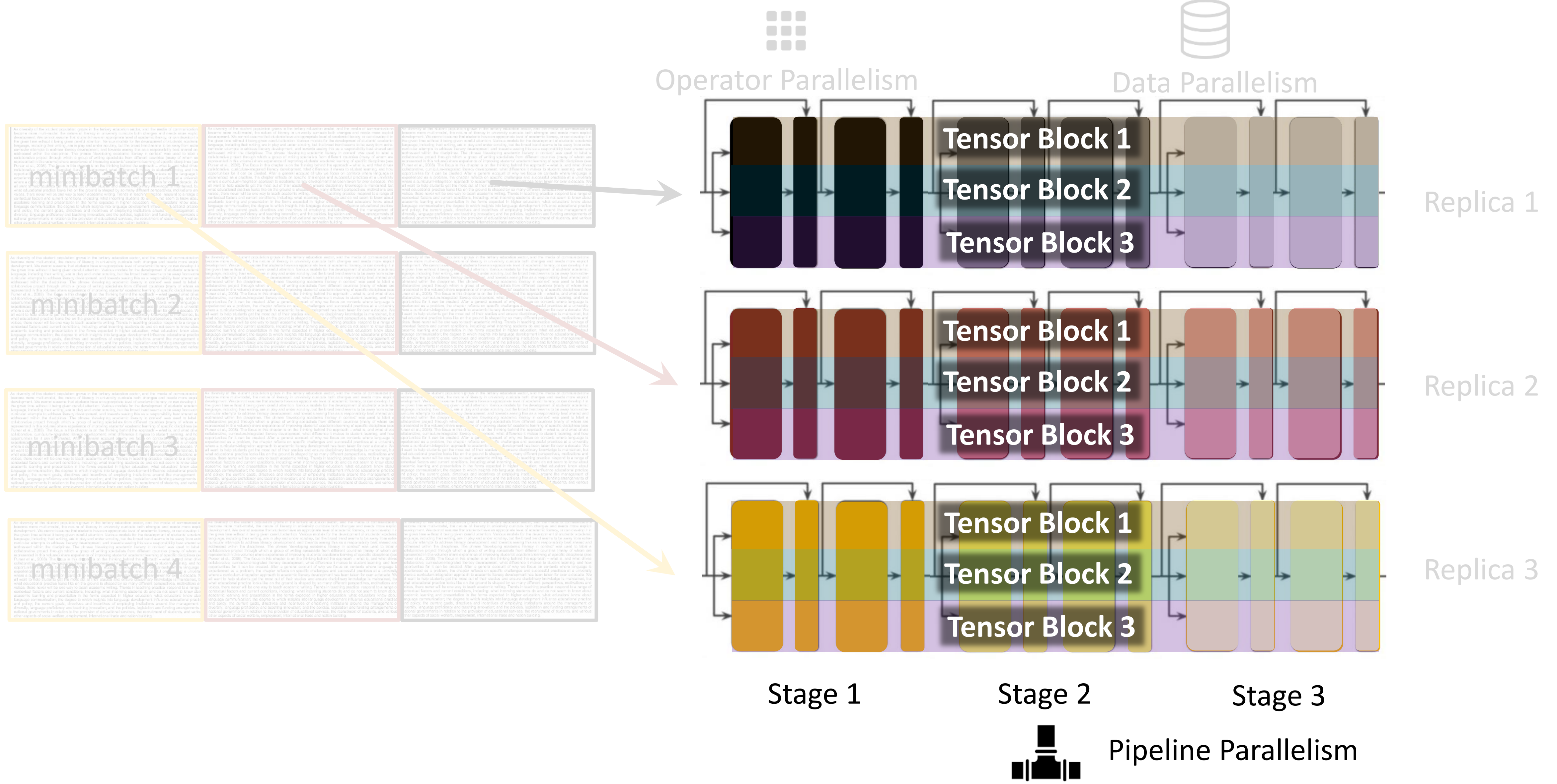


ve:

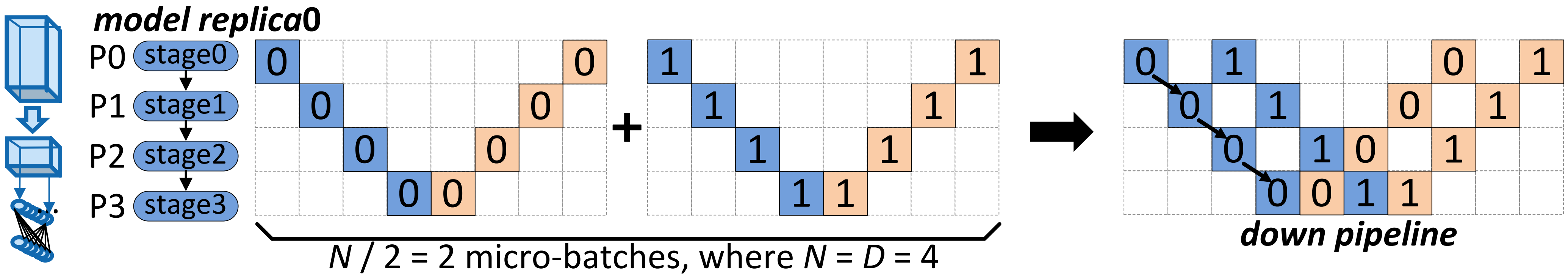
tasks in

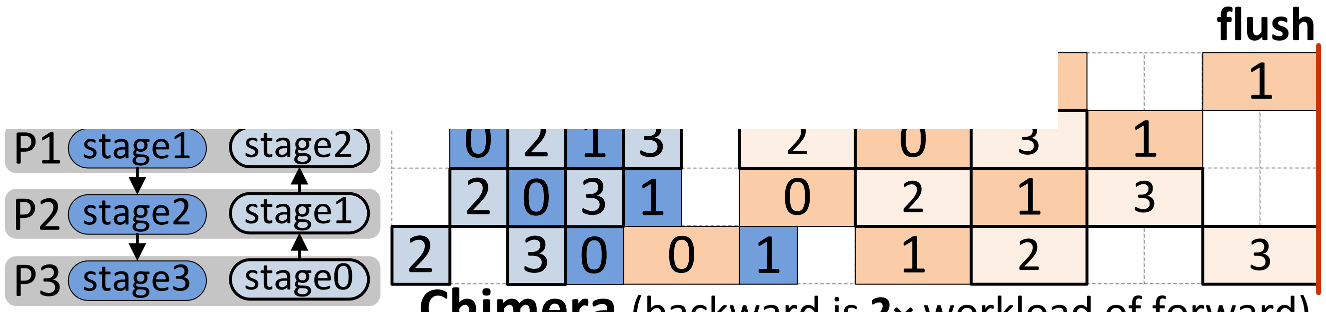
[1] Dan Alistarh, TH, et al.: "The Convergence of Sparsified Gradient Methods", NIPS'18

The Three Dimensions of Parallelism in Deep Learning (arXiv:1802.09941)



Bidirectional Pipelines – Meet Chimera (arXiv: 2107.06925v3)

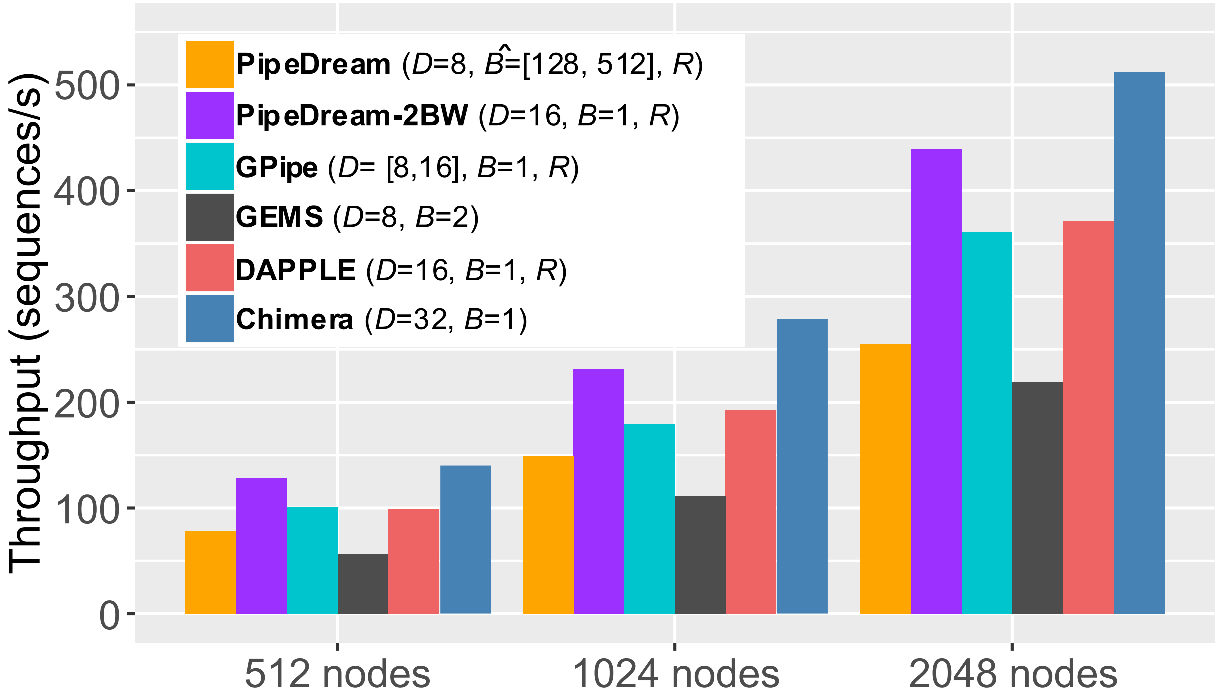




Chimera (backward is **2×** workload of forward)

S. Li, T. Hoefler: Chimera: Efficiently Training Large-Scale Neural Networks with Bidirectional Pipelines, best paper candidate at Supercomputing, SC21

Chimera Weak Scaling (arXiv: 2107.06925v3)

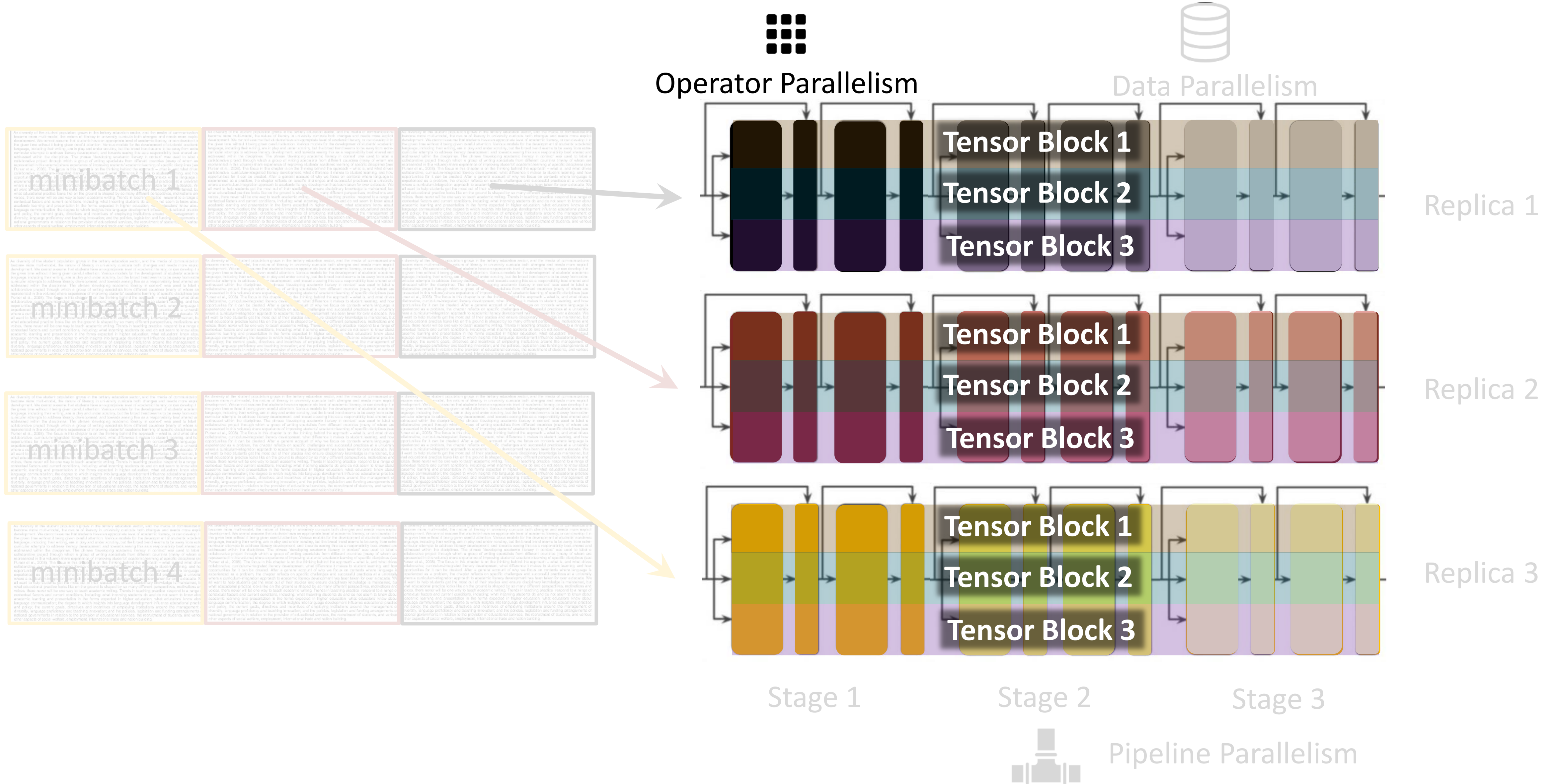


Weak scaling for GPT-2 on Piz Daint
(512 to 2048 GPU nodes)

- **1.38x - 2.34x speedup over synchronous approaches (GPipe, GEMS, DAPPLE)**
  - Less bubbles
  - More balanced memory thus no recomputation
- **1.16x - 2.01x speedup over asynchronous approaches (PipeDream-2BW, PipeDream)**
  - More balanced memory thus no recomputation
  - Gradient accumulation thus low synch frequency

S. Li, T. Hoefler: Chimera: Efficiently Training Large-Scale Neural Networks with Bidirectional Pipelines, best paper candidate at Supercomputing, SC21

The Three Dimensions of Parallelism in Deep Learning (arXiv:1802.09941)



T. Ben-Nun, T. Hoefler: Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis, arXiv Feb 2018

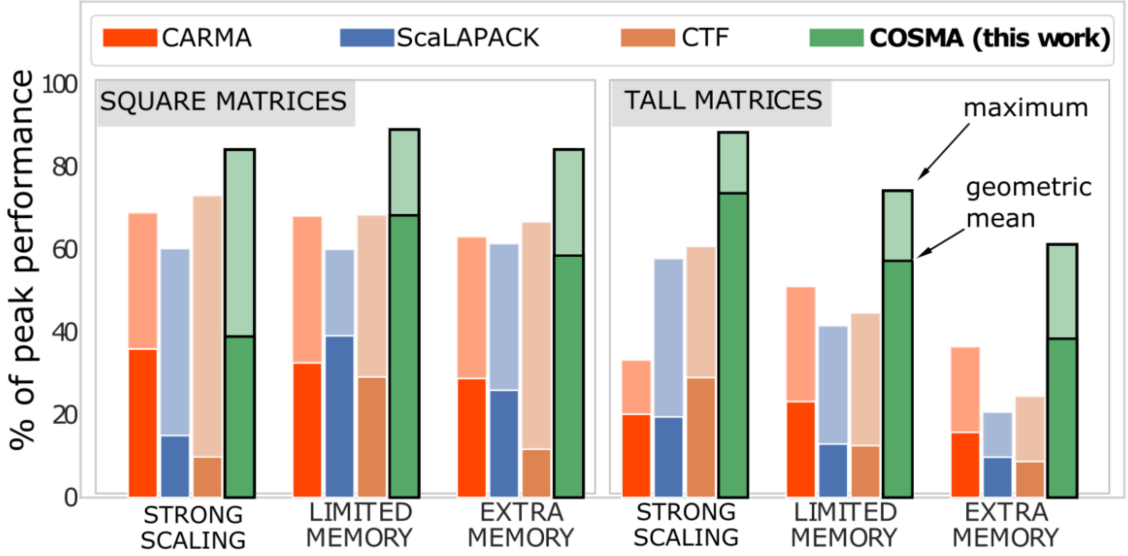
Operator Parallelism, i.e., Parallel Matrix Matrix Multiplication

Remember those?
All MMM!

- **Large MMMs dominate large language models!**
  - e.g., GPT-3 multiples 12,288x12,288 matrices
    600 MiB in fp32 and 1.9 Tflop
  - generative inference multiplies tall & skinny matrices

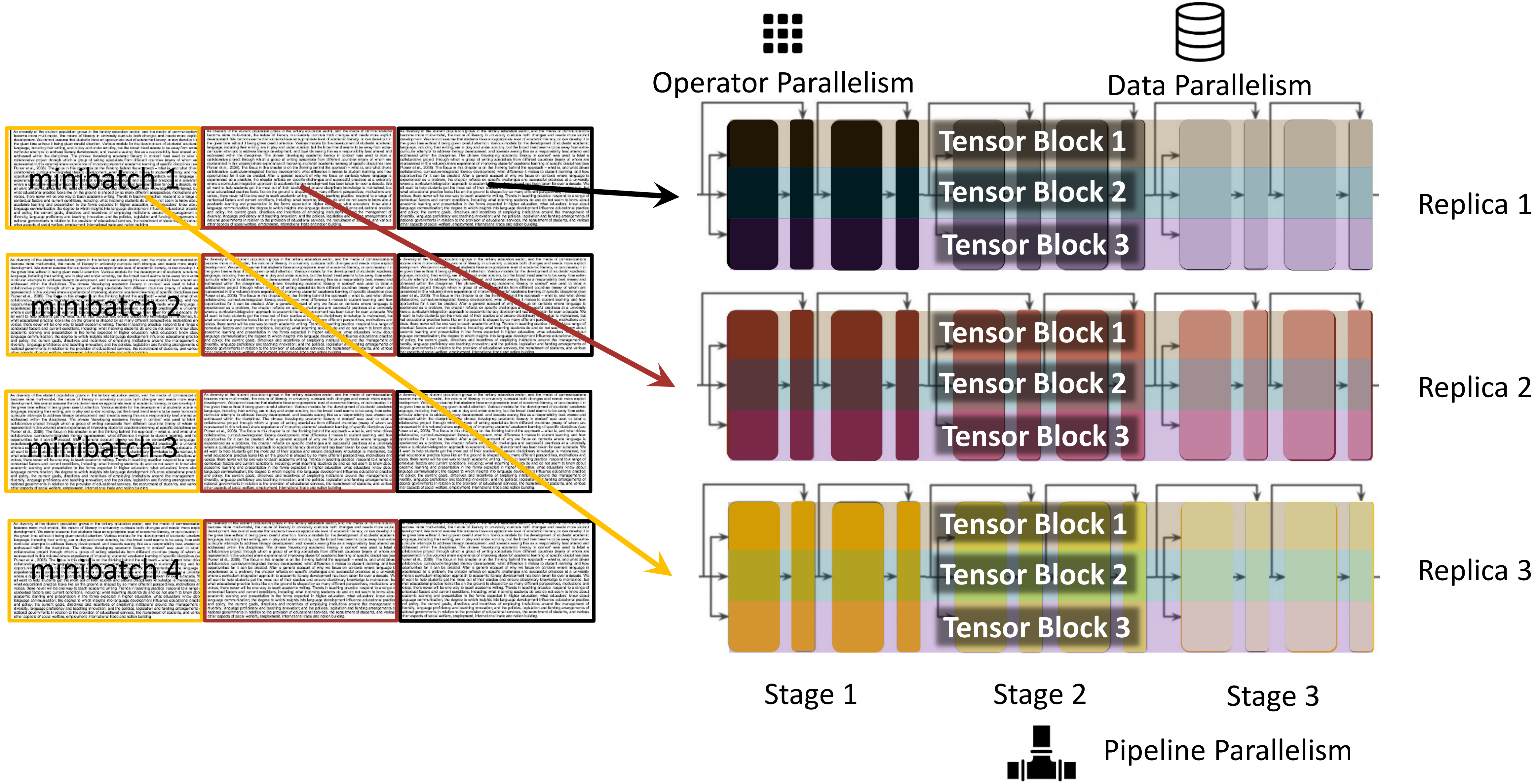
Operator class	% flop	% Runtime
Tensor contraction	99.80	61.0
Statistical normalization	0.17	25.5
Element-wise	0.03	13.5

- **Distribute as operator parallelism**
  - Heaviest communication dimension!
    Requires most optimization!
- **COSMA [1] communication-optimal distributed MMM**
  - Achieves tight I/O lower bound of $Q \geq \min\left\{\frac{2mnk}{p\sqrt{S}} + S, 3\left(\frac{mnk}{p}\right)^{\frac{2}{3}}\right\}$
  - Uses partial replication with an outer-product schedule
    See paper for details and proofs!
- **AutoDDL [2] combines operator-parallel models into communication-avoiding data distribution**



[1] G. Kwasniewski et al.: "Red-Blue Pebbling Revisited: Near Optimal Parallel Matrix-Matrix Multiplication", best student paper at Supercomputing SC19
[2] J. Chen et al.: "AutoDDL: Automatic Distributed Deep Learning with Asymptotically Optimal Communication", arXiv

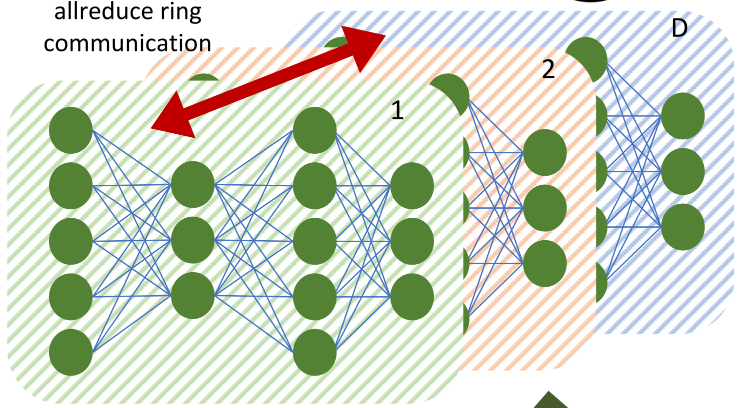
The Three Dimensions of Parallelism in Deep Learning (arXiv:1802.09941)



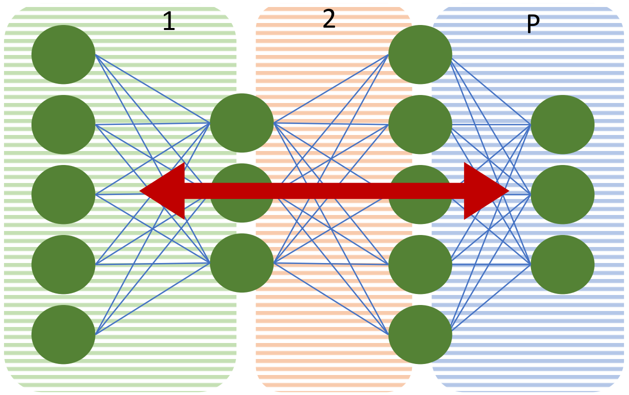
T. Ben-Nun, T. Hoefler: Demystifying Parallel and Distributed Deep Learning: An In-Depth Concurrency Analysis, arXiv Feb 2018

Communications in 3D Parallelism in Deep Learning (arXiv:2209.01346)

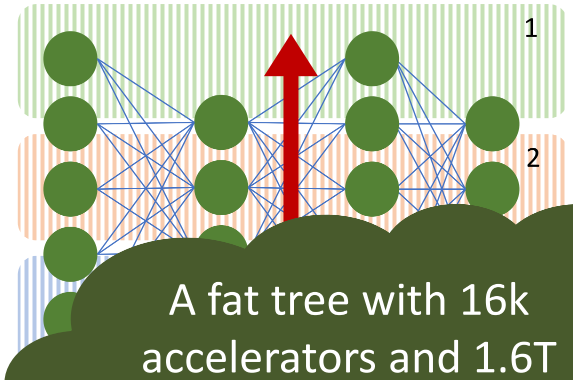
Data Parallelism



Pipeline Parallelism



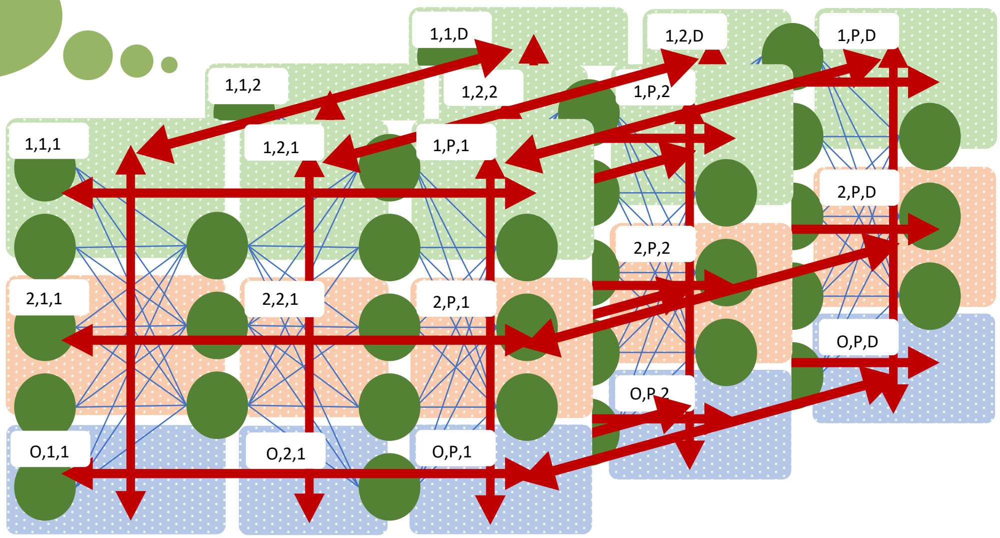
Operator Parallelism



A fat tree with 16k accelerators and 1.6T would cost $680M! ☹ ☹ ☹

Communication is (largely) a logical 3D Torus

3D - Data, Pipeline, and Operator Parallelism



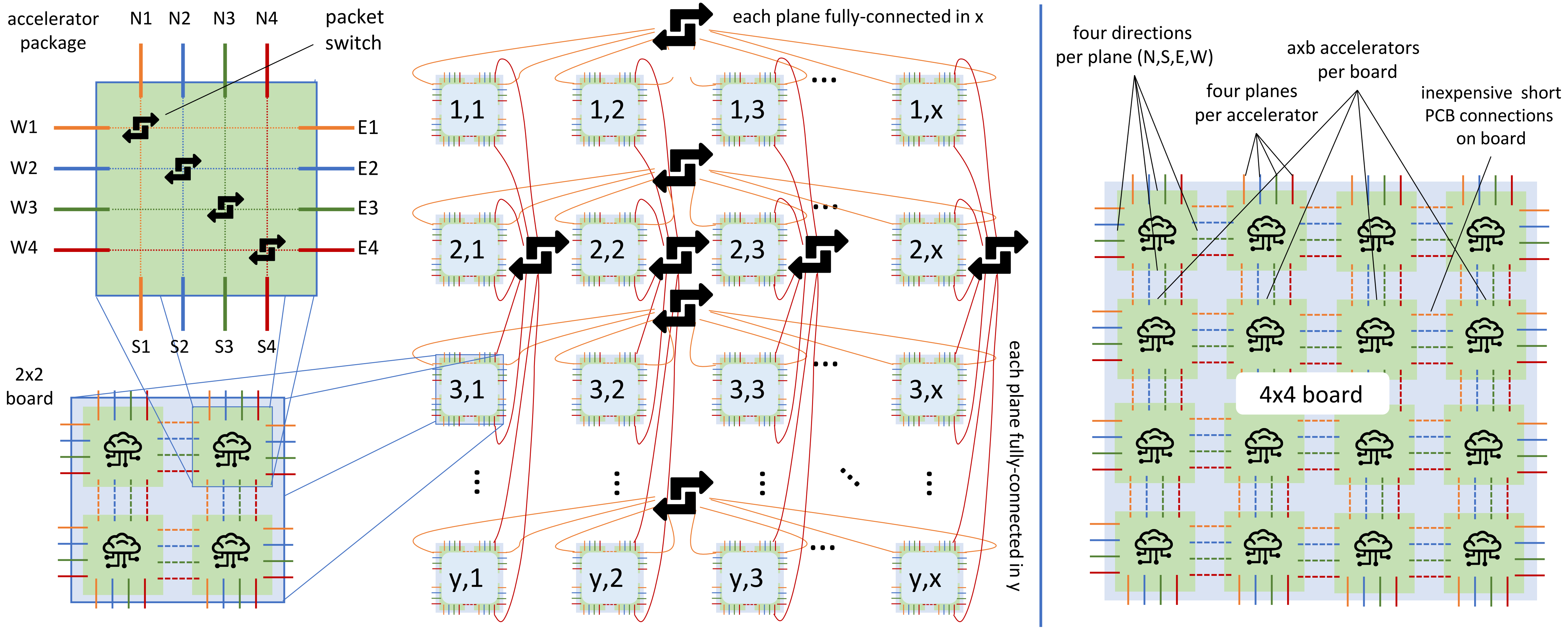
AI bandwidth today / yesterday (and growing!)

- Google TPUv2 (‘21): 1T
- AWS Trainium (‘21): 1.6T
- DGX-2 (A100, ‘21): 4.8T (islands of NVLINK)
- Tesla Dojo (‘22): 128T

→ Broadcom TH5 / NVIDIA Spectrum 4: 51.2T

TH et. al.: HammingMesh: A Network Topology for Large-Scale Deep Learning, SC22 (best reproducibility award recipient) and arXiv (2209.01346)

Co-designing an AI Supercomputer with Unprecedented and Cheap Bandwidth



TH et. al.: HammingMesh: A Network Topology for Large-Scale Deep Learning, SC22 (best reproducibility award recipient) and arXiv (2209.01346)