T. HOEFLER

From Big Large Language Models to Fast R dels

Three Eras in The Age of Computation.

with contributions by the whole SPCL deep learning team (M. Besta, J. Barth, E. Schr crosoft Azure (M. Heddes, J. Belk, S. Scott, D. Goel, M. Castro) and collaborators (D. Alistarh and others)

DP2E-AI, Paris, June, 2025

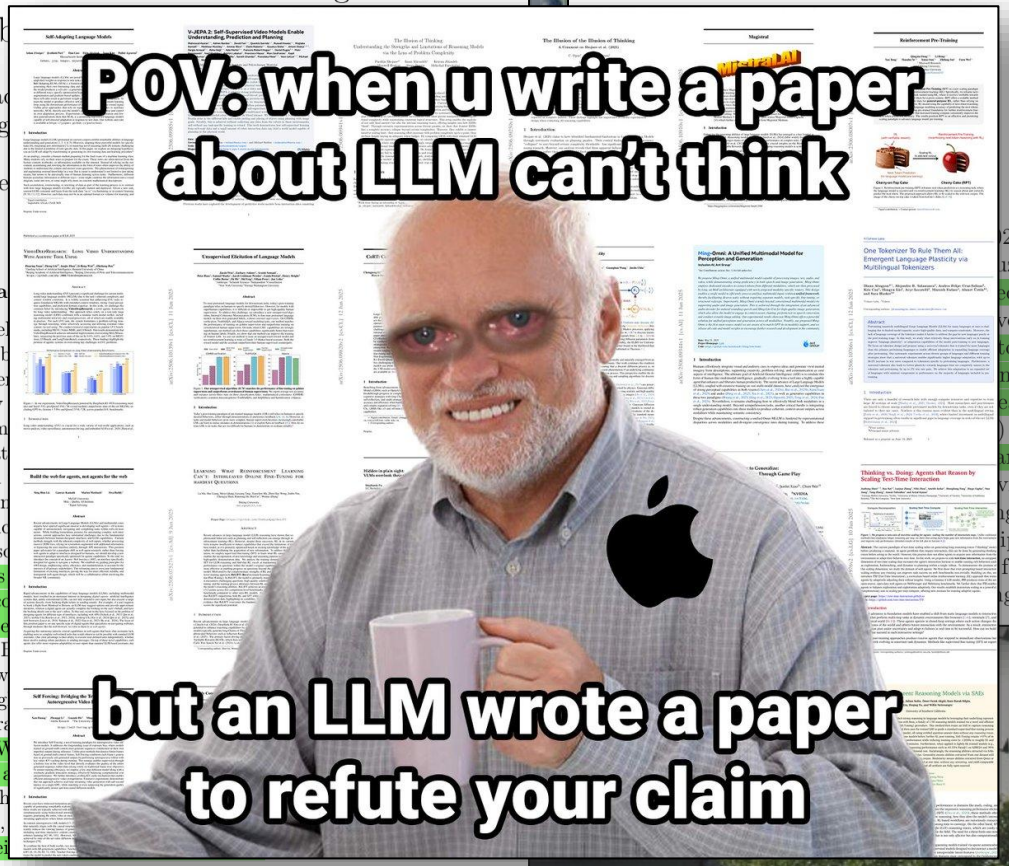
The Illusion of Thinking:
Understanding the Strengths and Limitations of Reasoning Models via the Lens of Prob

Parshin Shojaee*† Iman Mirzad
Maxwell Horton Samy Beng

Apple

Abstra

Recent generations of frontier language model (LRMs) that generate detailed thinking processes h demonstrate improved performance on reasoning be ing properties, and limitations remain insufficiently cus on established mathematical and coding benchm ever, this evaluation paradigm often suffers from dat into the reasoning traces' structure and quality. In gaps with the help of controllable puzzle environmer tional complexity while maintaining consistent logic of not only final answers but also the internal reasc "think". Through extensive experimentation across face a complete accuracy collapse beyond certain co intuitive scaling limit: their reasoning effort increase declines despite having an adequate token budget. I counterparts under equivalent inference compute, w complexity tasks where standard models surprising tasks where additional thinking in LRMs demonstra where both models experience complete collapse. W computation: they fail to use explicit algorithms a also investigate the reasoning traces in more depth and analyzing the models' computational behavior, and ultimately raising crucial questions about thei



The Illusion e Illusion of Thinking

A Comm on Shojaee et al. (2025)

C. Opus* A. Lawsen†

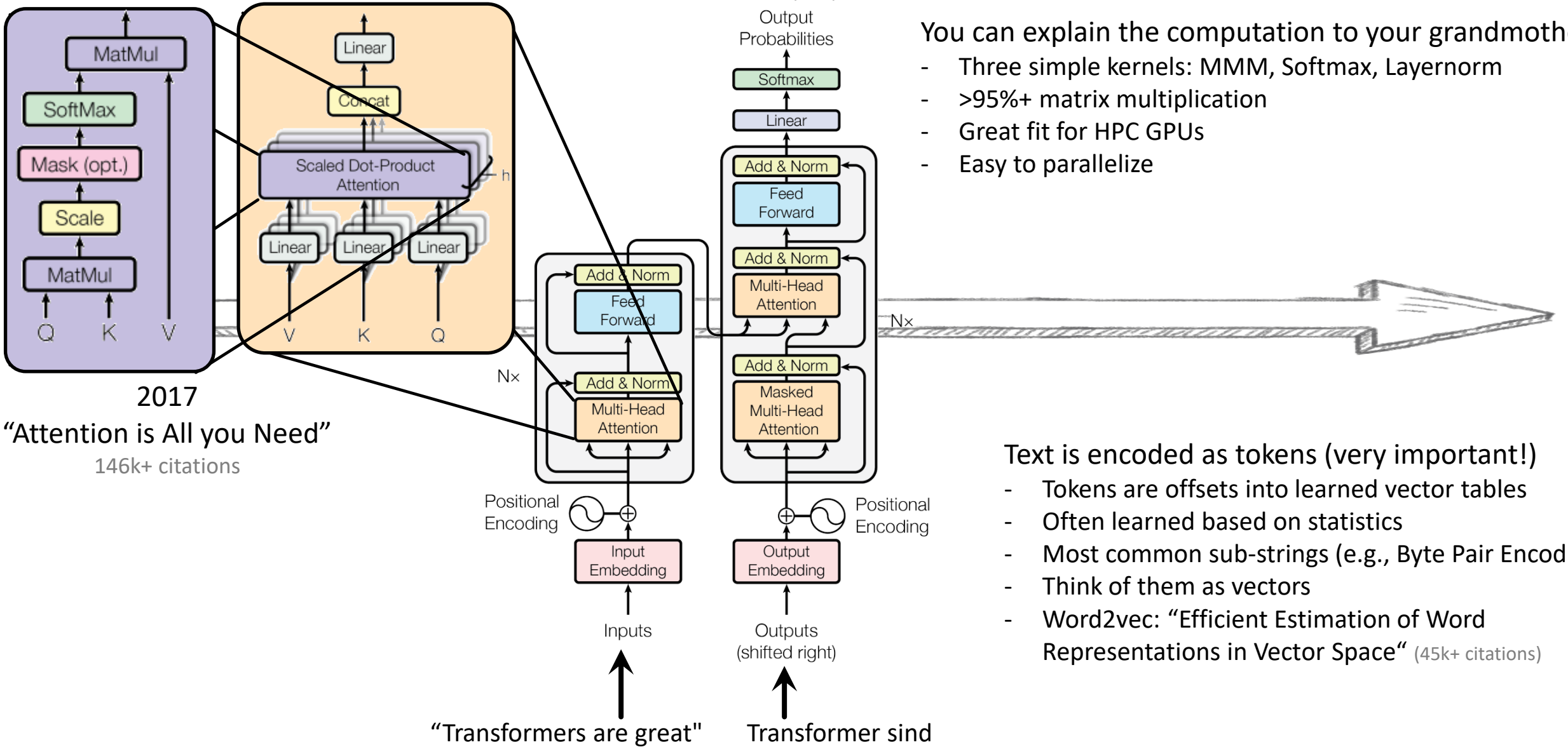
June 10, 2025

Abstract

25) report that Large Reasoning Models (LRMs) exhibit "accuracy col- uzzles beyond certain complexity thresholds. We demonstrate that their ct experimental design limitations rather than fundamental reasoning fail- veals three critical issues: (1) Tower of Hanoi experiments systematically oken limits at reported failure points, with models explicitly acknowledg- n their outputs; (2) The authors' automated evaluation framework fails to easoning failures and practical constraints, leading to misclassification of) Most concerningly, their River Crossing benchmarks include mathemat- nces for $N \geq 6$ due to insufficient boat capacity, yet models are scored ving these unsolvable problems. When we control for these experimental g generating functions instead of exhaustive move lists, preliminary ex- iple models indicate high accuracy on Tower of Hanoi instances previously failures. These findings highlight the importance of careful experimental g AI reasoning capabilities.

From LLMs to AHI

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



2017

"Attention is All you Need"

146k+ citations

You can explain the computation to your grandmother!

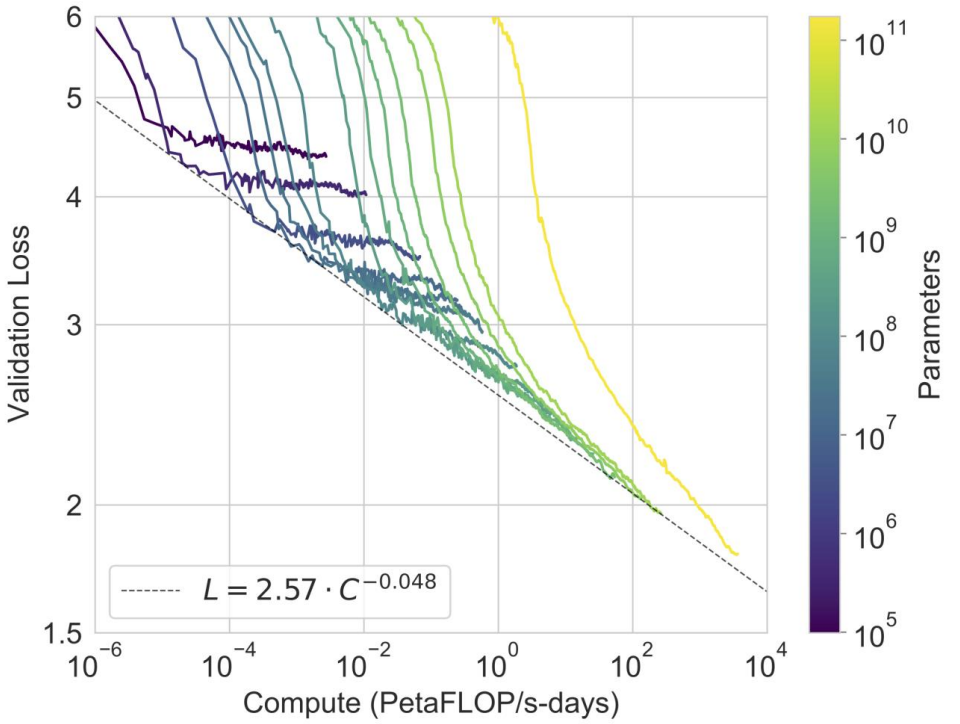
- Three simple kernels: MMM, Softmax, Layernorm
- >95%+ matrix multiplication
- Great fit for HPC GPUs
- Easy to parallelize

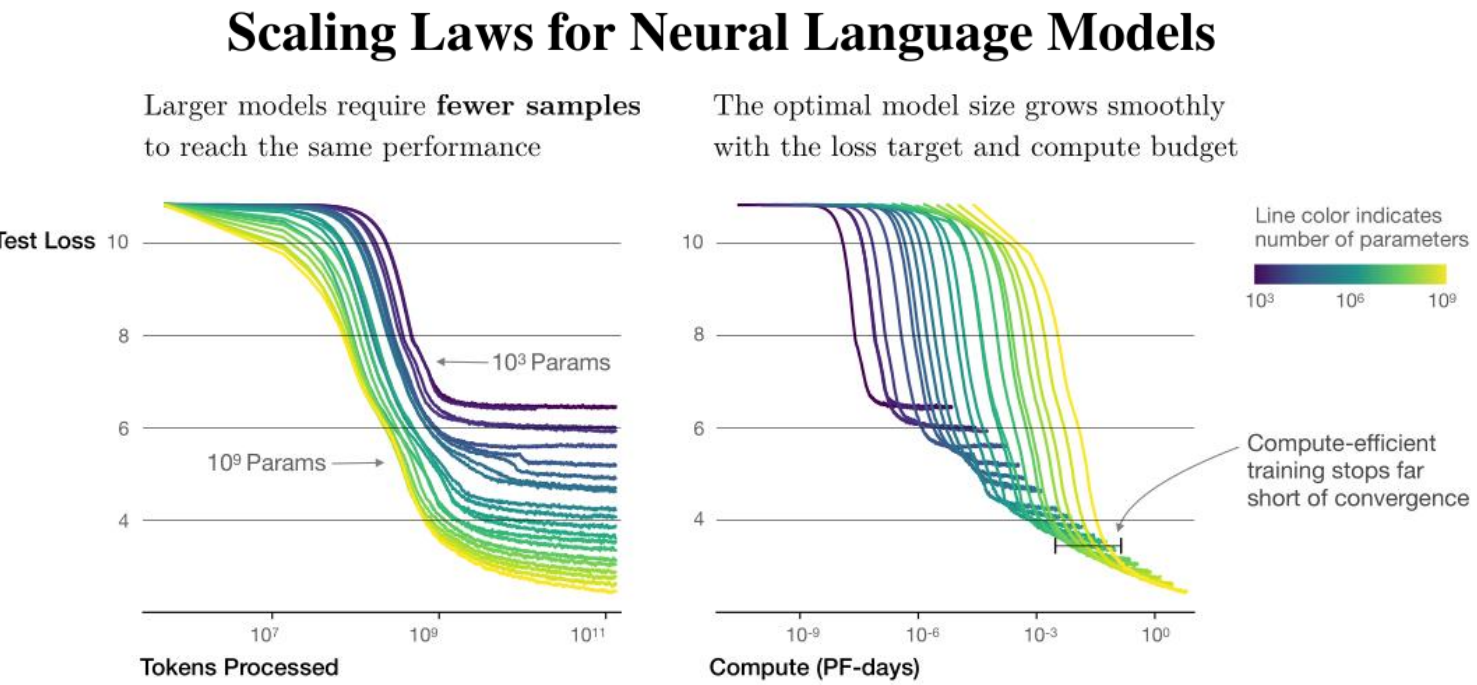
Text is encoded as tokens (very important!)

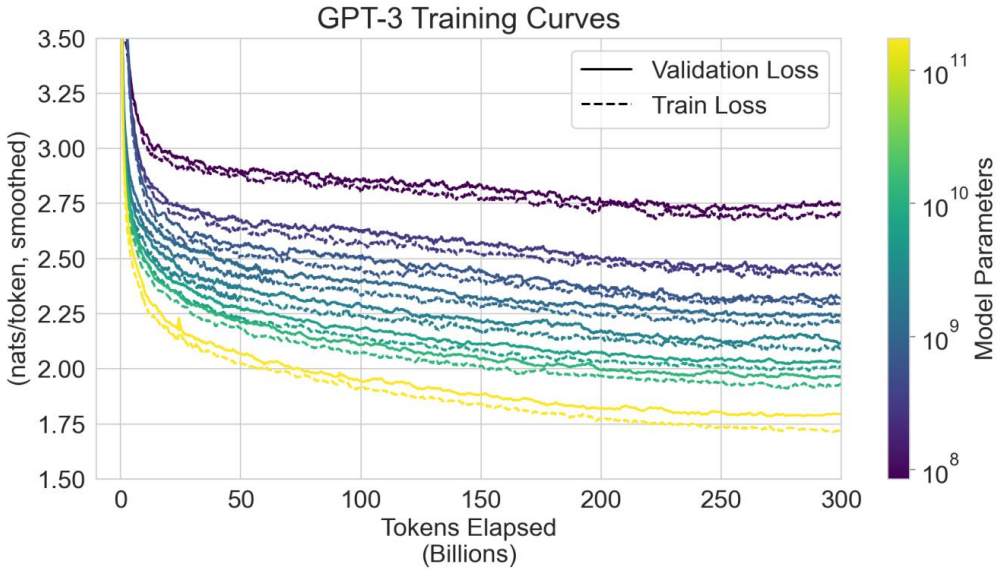
- Tokens are offsets into learned vector tables
- Often learned based on statistics
- Most common sub-strings (e.g., Byte Pair Encoding)
- Think of them as vectors
- Word2vec: "Efficient Estimation of Word Representations in Vector Space" (45k+ citations)

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"

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

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

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

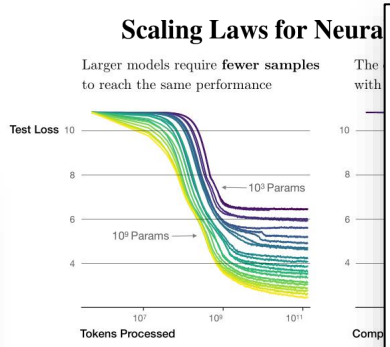
By The Associated Press
January 24, 2022, 10:33 PM

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



2020 - **GPT-3** (202
"Language Mo
Few-Shot Lea
37k+ citati

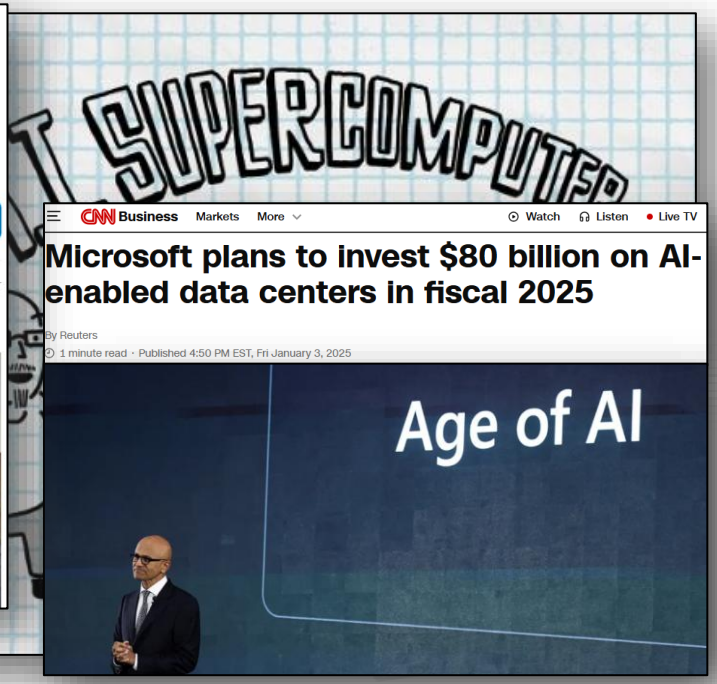
Trump's AI Push: Understanding The $500 Billion Stargate Initiative

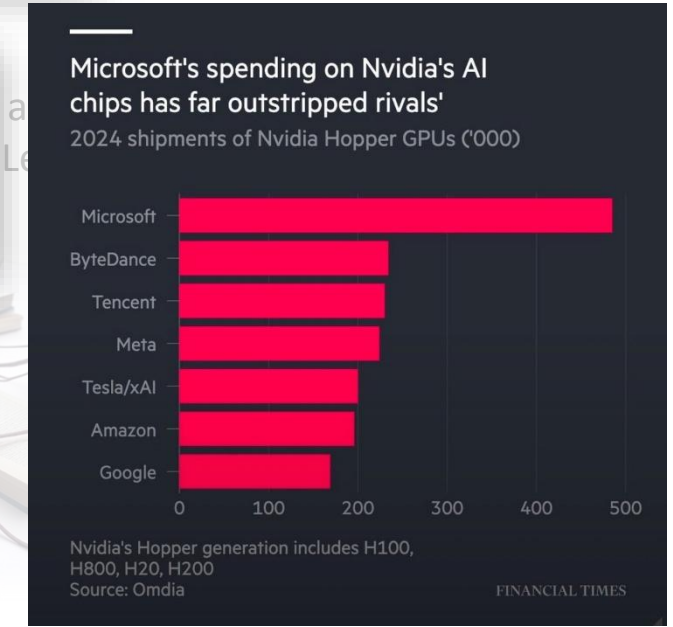
Garth Friesen Contributor
Specialist in global markets, economics and alternative investments.

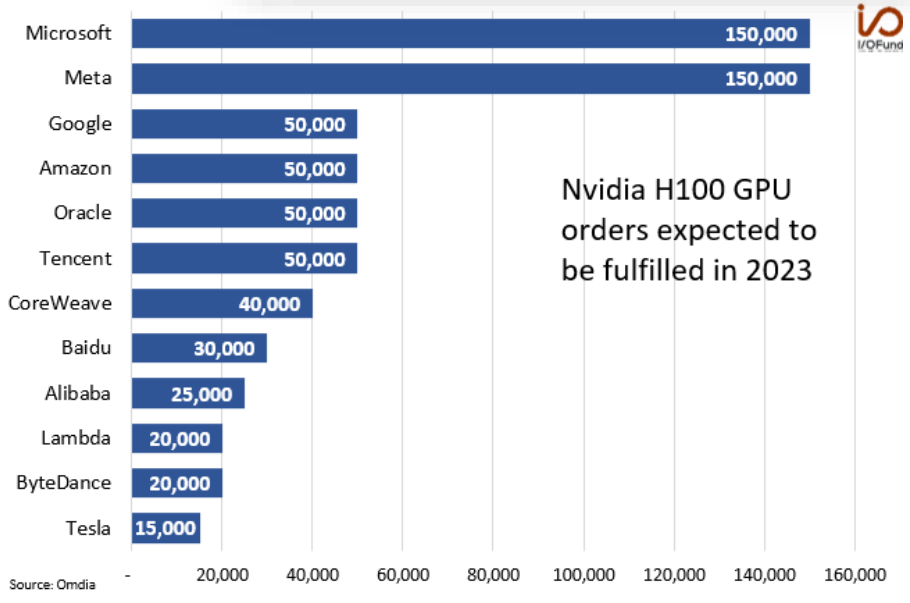
Follow

Jan 23, 2025, 08:00am EST

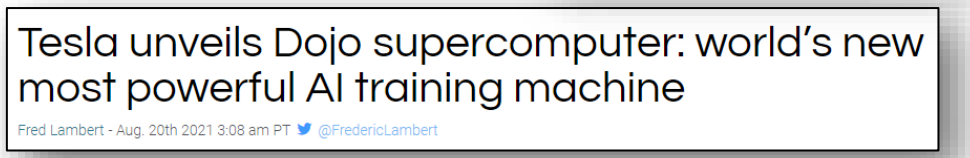
Updated Jan 24, 2025, 07:25am EST

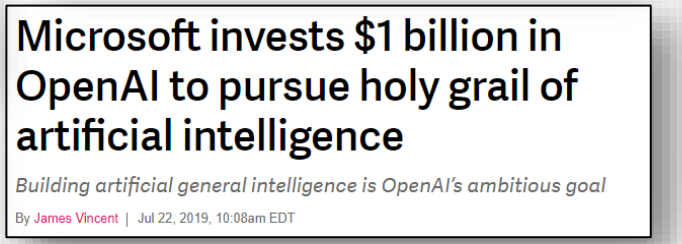


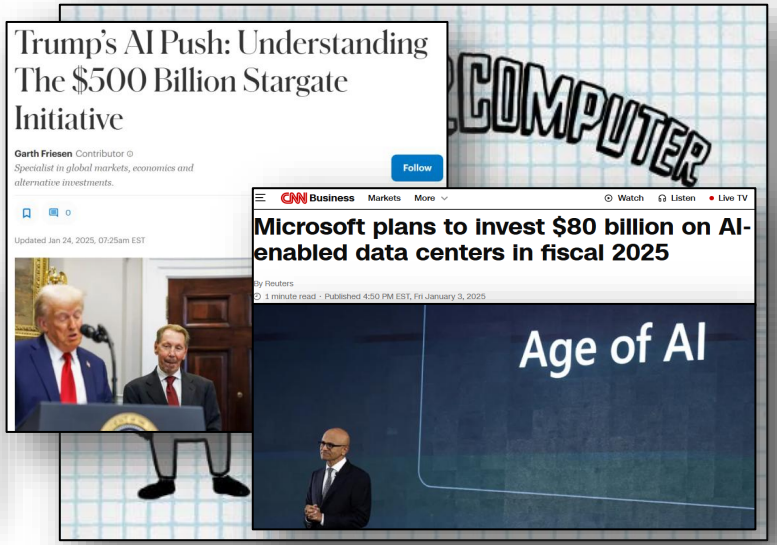




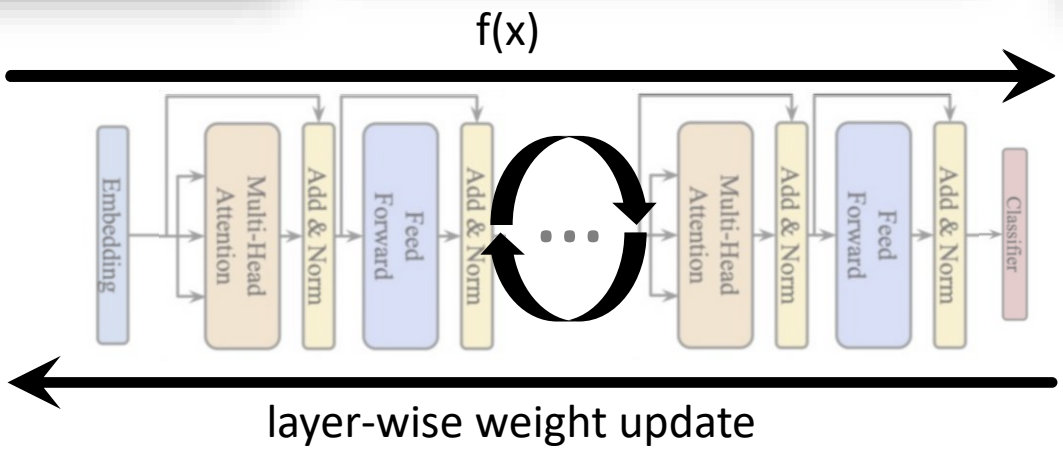
Supercomputers fuel Modern AI







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**!

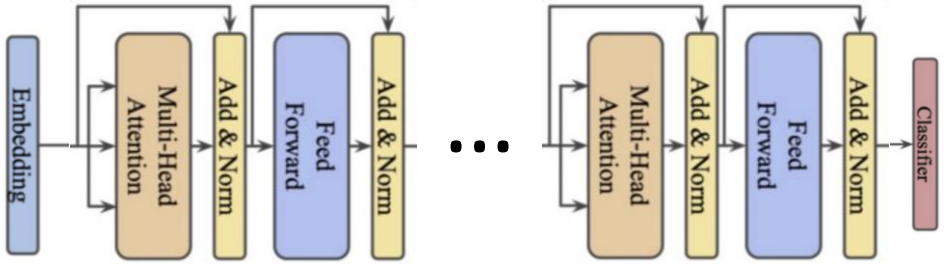
- PaLM-540B: 118 (complex) layers
  *540 bn parameters (**1 TiB** in fp16)*
  2048-token "sentences"

- 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

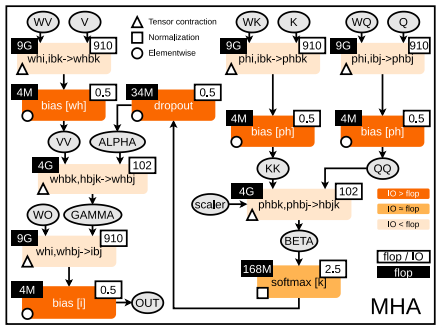
Data Movement Is All You Need: A Case Study on Optimizing Transformers (arXiv:2007.00072)

GPT/BERT encoder



Express as Dataflow

DaCe



highly optimized

Operator class	% flop	% Runtime
Tensor contraction	99.80	61.0
Statistical normalization	0.17	25.5
Element-wise	0.03	13.5
	0.2%	39%

Our performance improvement for BERT-large

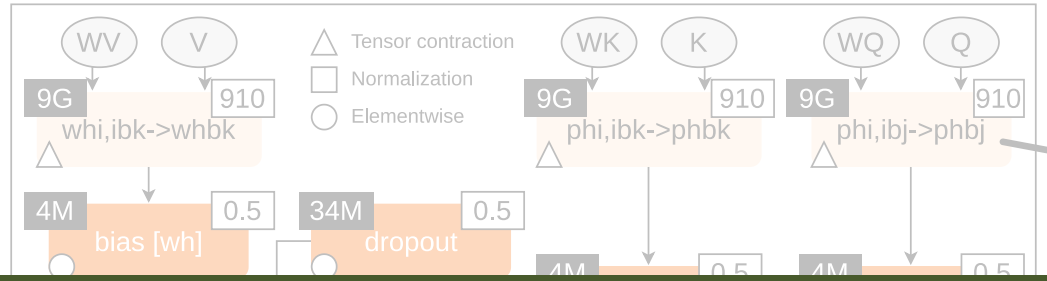
- 30% over PyTorch
- 20% over Tensorflow + XLA
- 8% over DeepSpeed

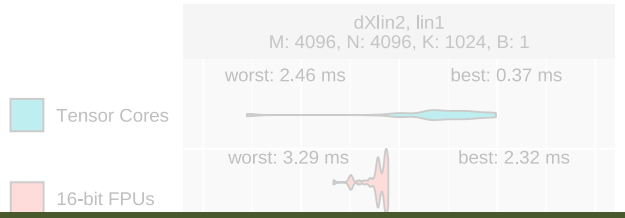
OpenAI booth at NeurIPS 2019 in Vancouver, Canada
Image Credit: Khari Johnson / VentureBeat

Last week, OpenAI published a paper detailing GPT-3, a machine learning model that achieves strong results on a number of natural language benchmarks. At 175 billion parameters, where a parameter affects data's prominence in an overall prediction, it's the largest of its kind. And with a memory size exceeding 350GB, it's one of the priciest, costing an estimated $12 million to train.

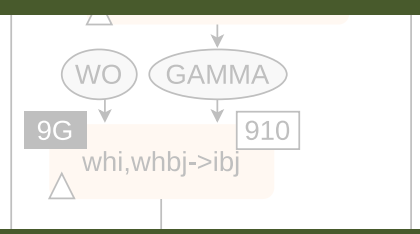
est. savings on AWS over PyTorch:
$85k for BERT, $3.6M GPT-3

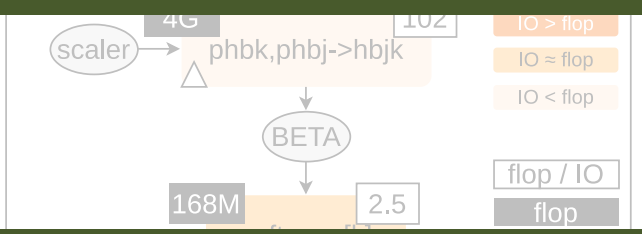
Data Movement Is All You Need: A Case Study on Optimizing Transformers (arXiv:2007.00072)

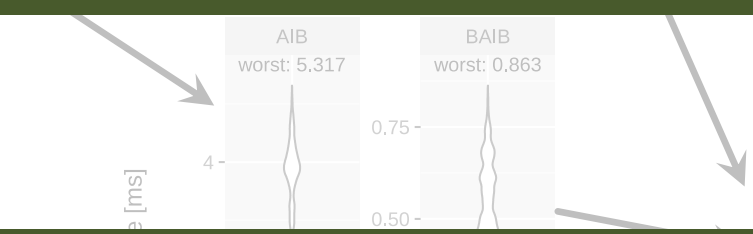


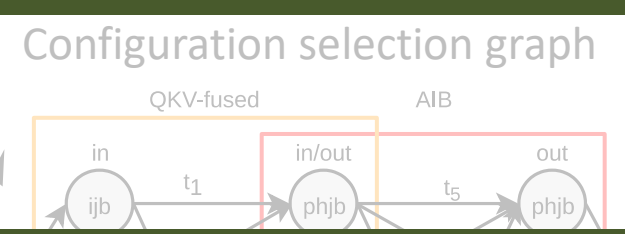


These Ideas are in Production Today





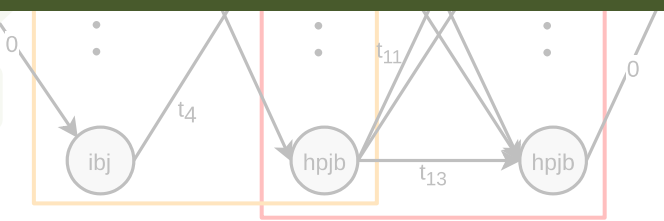




Reducing Cost Remains Imperative to Continue Scaling

Full BERT encoder layer performance (ms)

	TF+XLA	PyTorch	DeepSpeed	Ours
Forward	3.2	3.45	2.8	**2.63**
Backward	5.2	5.69	4.8	**4.38**

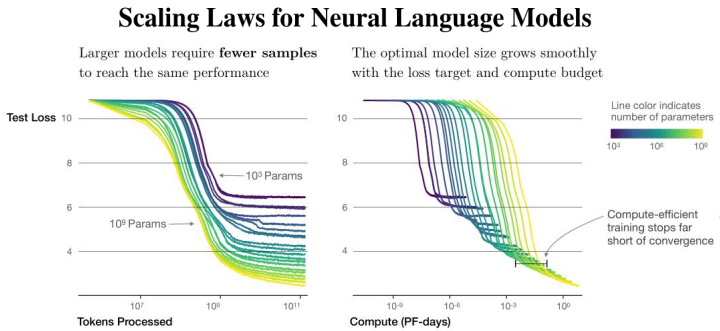


From LLMs to AHI

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

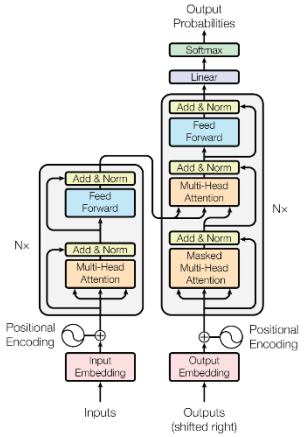
How to turn this into a business serving millions of customers?

era of model size scaling

2017 - **Transformers**

"Attention is All you Need"

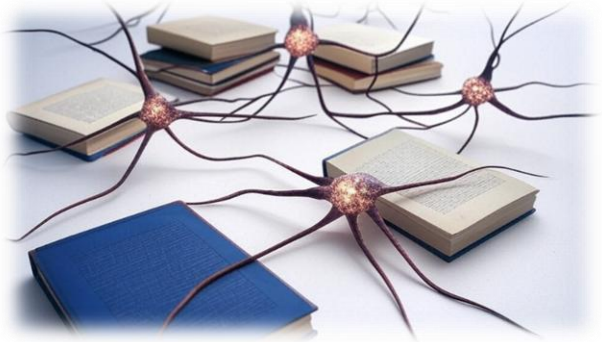
146k+ citations



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

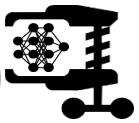
Next, let's see how to improve cost by

1,000x

Moving Data is Most Expensive!

Techniques to Shrink ML Data

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



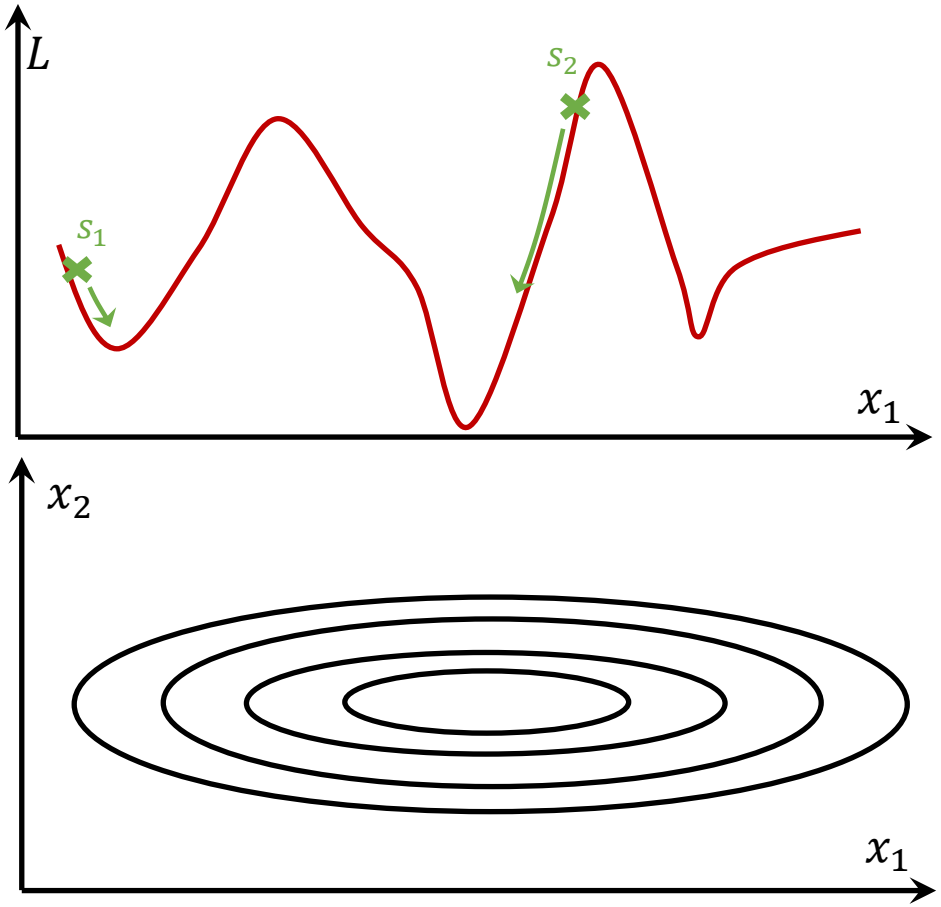
- **Brains have limited precision! Why are we computing with FP32?**
  - For technical reasons (SGD, optimization, how we quantize)
  - Neurons in Hippocampus can "reliably distinguish 24 strengths" [1]
    4.6 bits of information!
- **PaLM-540B has up to 540 billion parameters**
  - 1.08 TiB in FP16/BF16, 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)

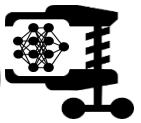
"Curvature" of error (aka. "sensitivity")

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)



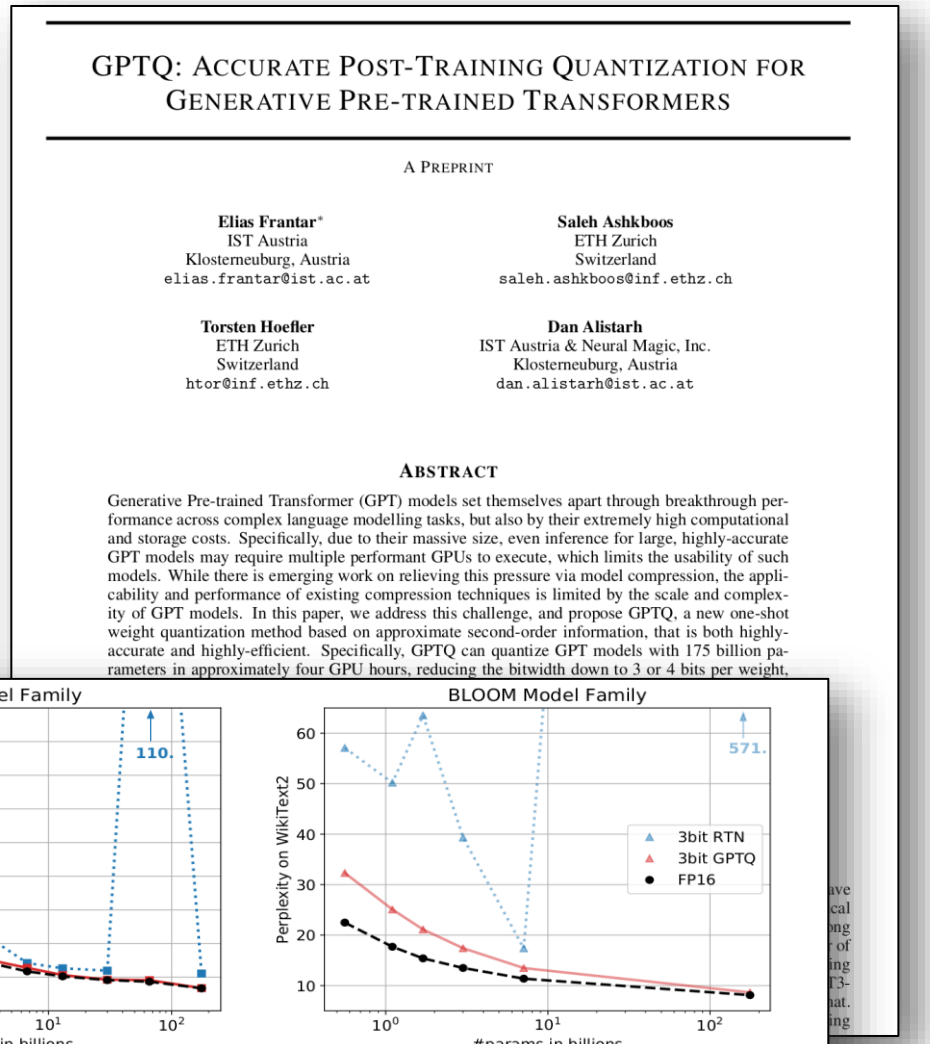
- **Quantization objective for low precision rounded weights $\hat{w}$**

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

- **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,

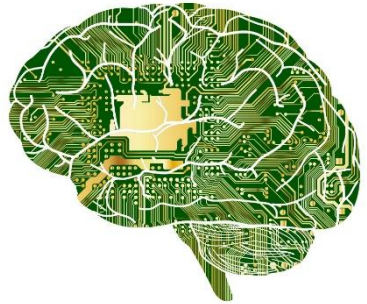
Figure 1: Quantizing OPT models to 4 and BLOOM models to 3 bit precision, comparing GPTQ with the FP16 baseline and round-to-nearest (RTN) [34, 5].

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!

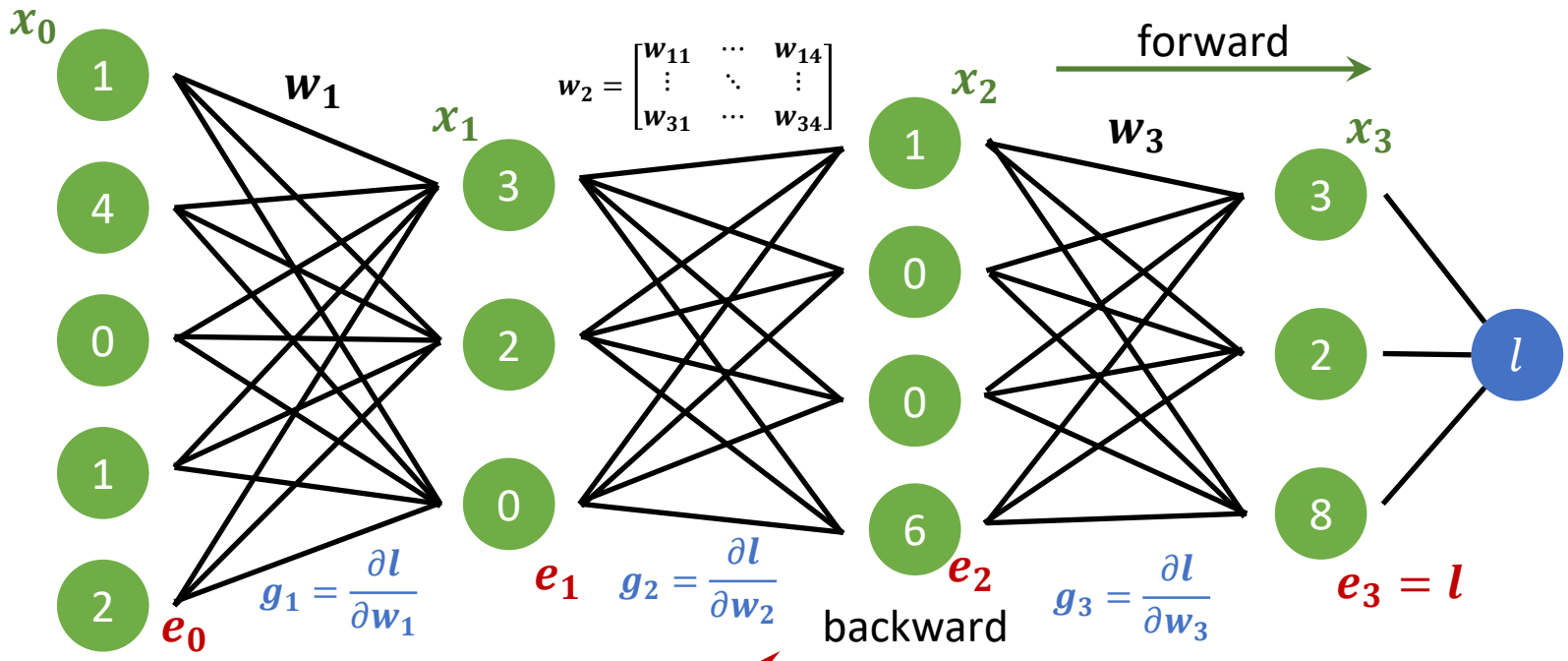
Intuition: not all features are always relevant!

- ○ 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!



Hoefler et al. "Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks", arXiv 2102.00554, Jan 2021

Much more (at least two hours more)

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

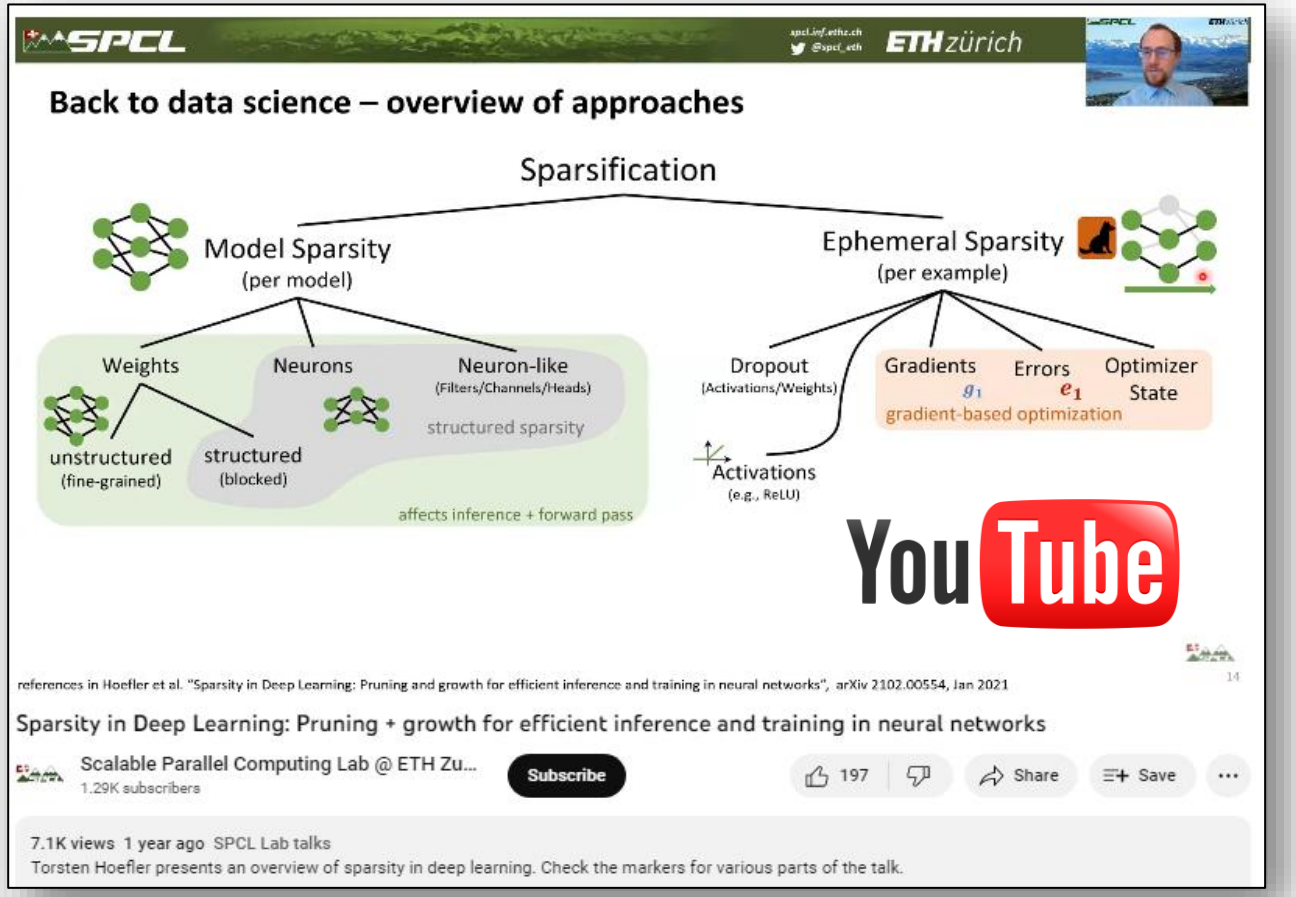
The growing energy and performance costs of deep learning have driven the community to reduce the size of neural networks by selectively pruning components. Similarly to their biological counterparts, sparse networks generalize just as well, if not better than, the original dense networks. Sparsity can reduce the memory footprint of regular networks to fit mobile devices, as well as shorten training time for ever growing networks. In this paper, we survey prior work on sparsity in deep learning and provide an extensive tutorial of sparsification for both inference and training. We describe approaches to remove and add elements of neural networks, different training strategies to achieve model sparsity, and mechanisms to exploit sparsity in practice. Our work distills ideas from more than 300 research papers and provides guidance to practitioners who wish to utilize sparsity today, as well as to researchers whose goal is to push the frontier forward. We include the necessary background on mathematical methods in sparsification, describe phenomena such as early structure adaptation, the intricate relations between sparsity and the training process, and show techniques for achieving acceleration on real hardware. We also define a metric of pruned parameter efficiency that could serve as a baseline for comparison of different sparse networks. We close by speculating on how sparsity can improve future workloads and outline major open problems in the field.

The supreme goal of all theory is to make the irreducible basic elements as simple and as few as possible without having to surrender the adequate representation of a single datum of experience -

Albert Einstein, 1933

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

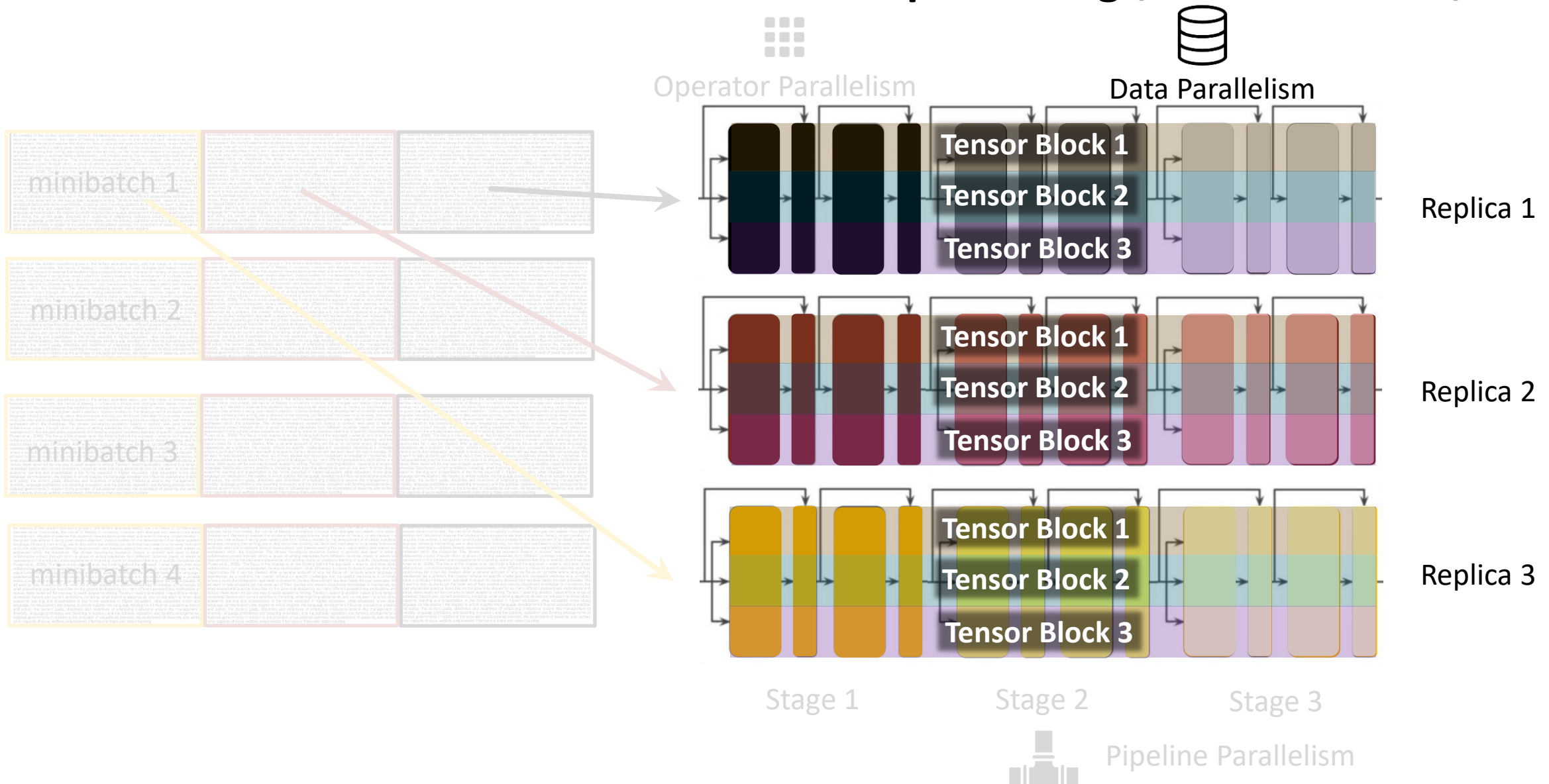
100x

More Efficient Processing

Which Makes Data Movement Even More Important!

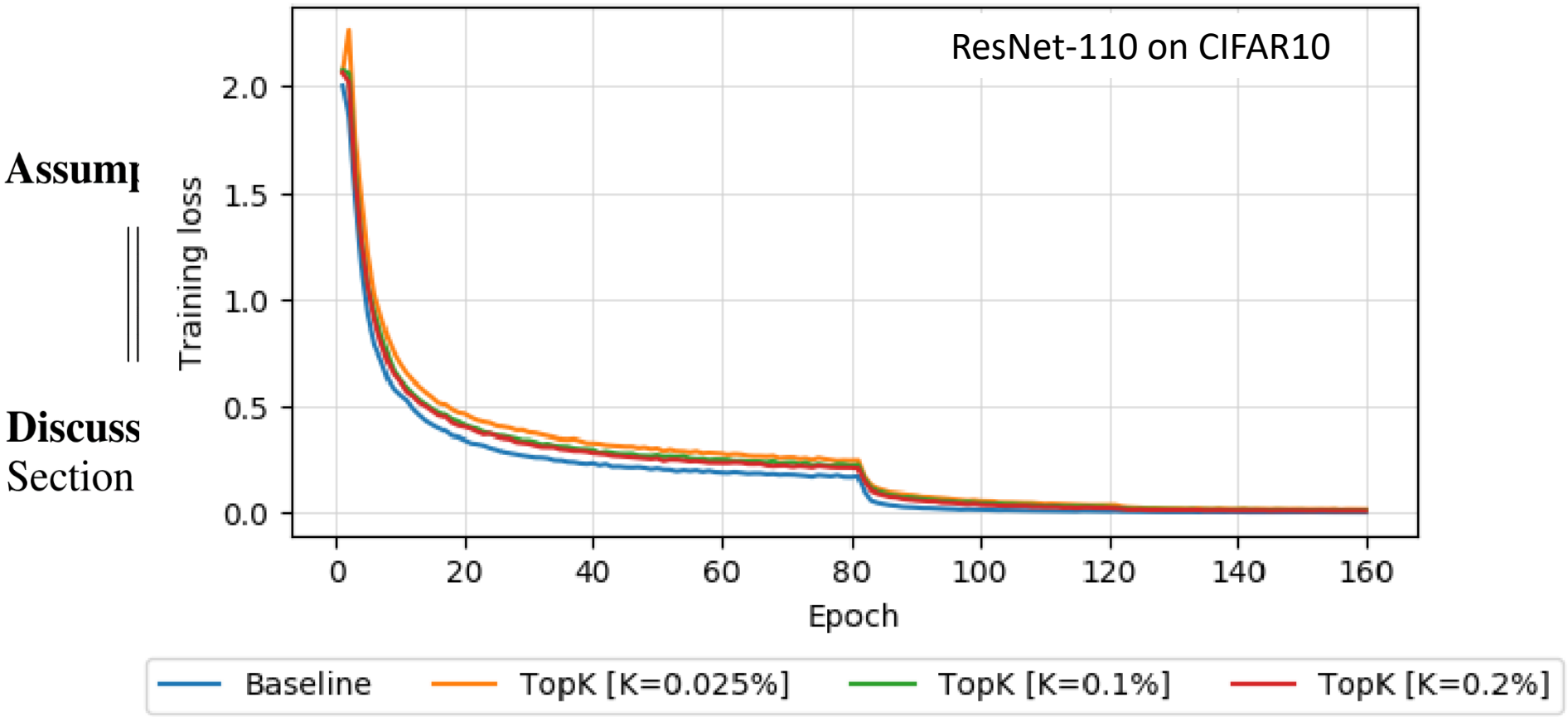
Especially in the Network!

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



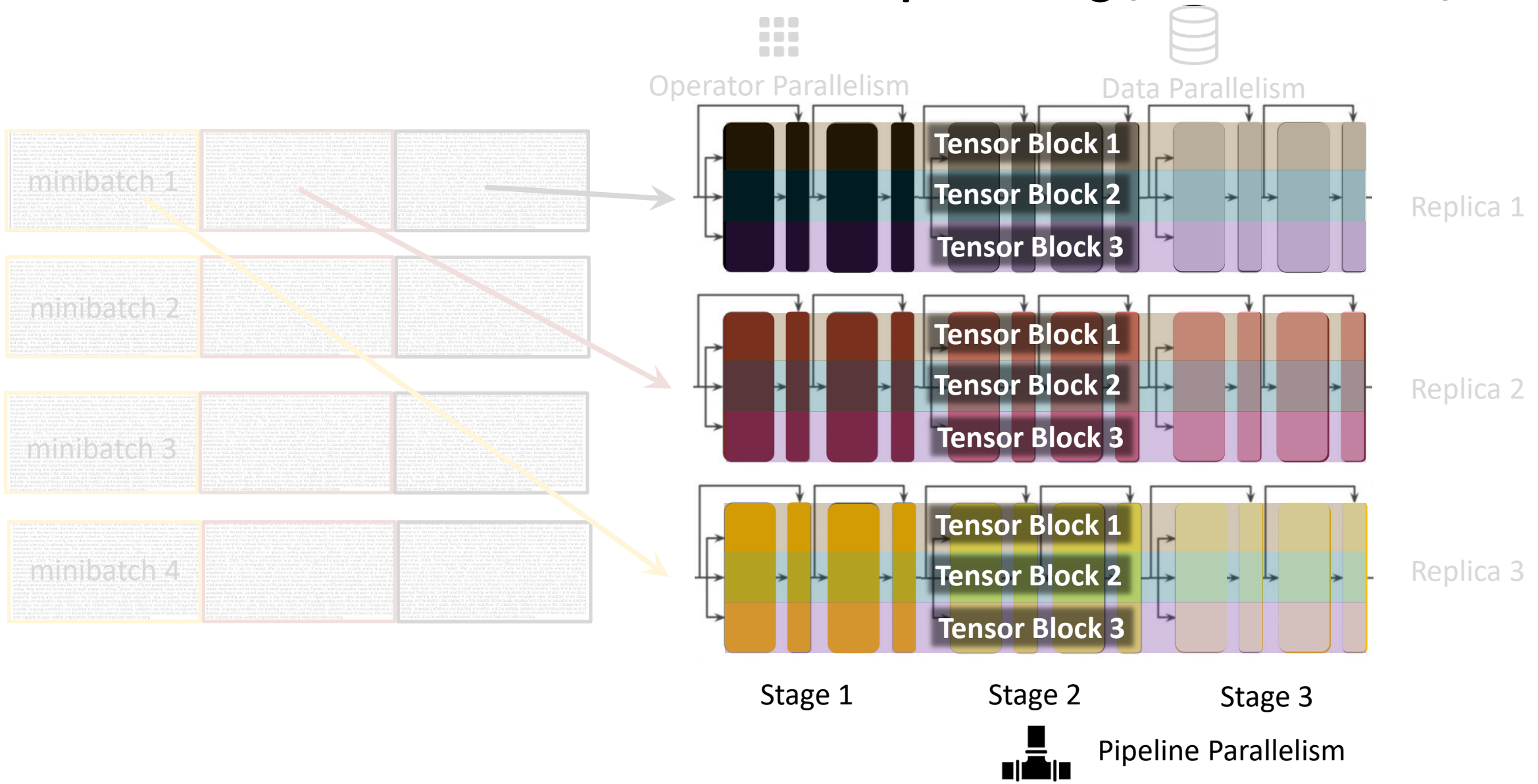
Data-parallel Gradient Sparsification – Top-k SGD (arXiv:1809.10505)

- **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])

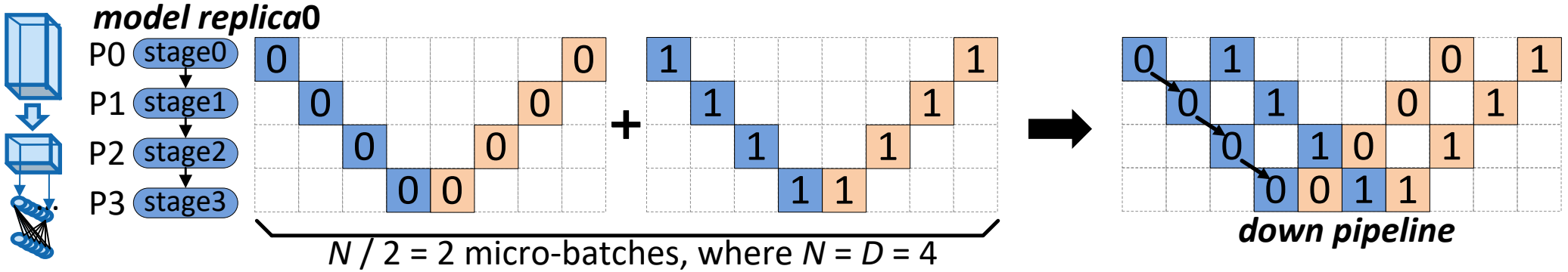


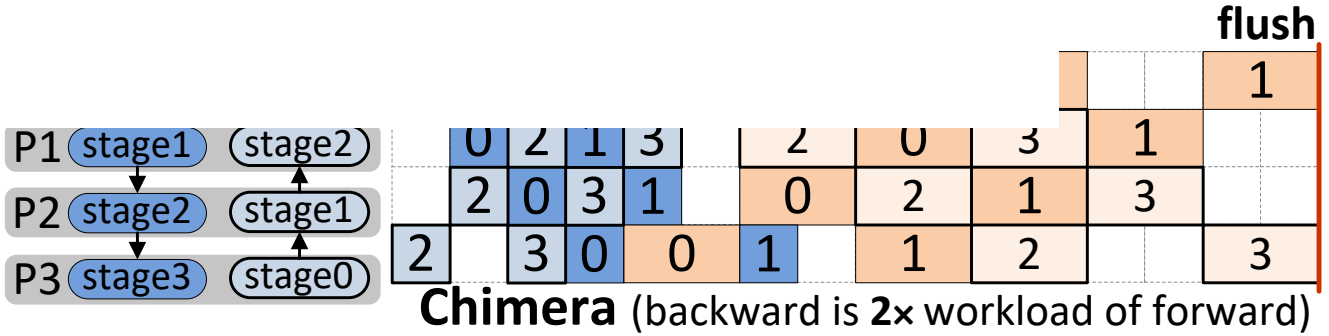
[1] Dan Alistarh, TH, et al.: "The Convergence of Sparsified Gradient Methods", NIPS'18

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



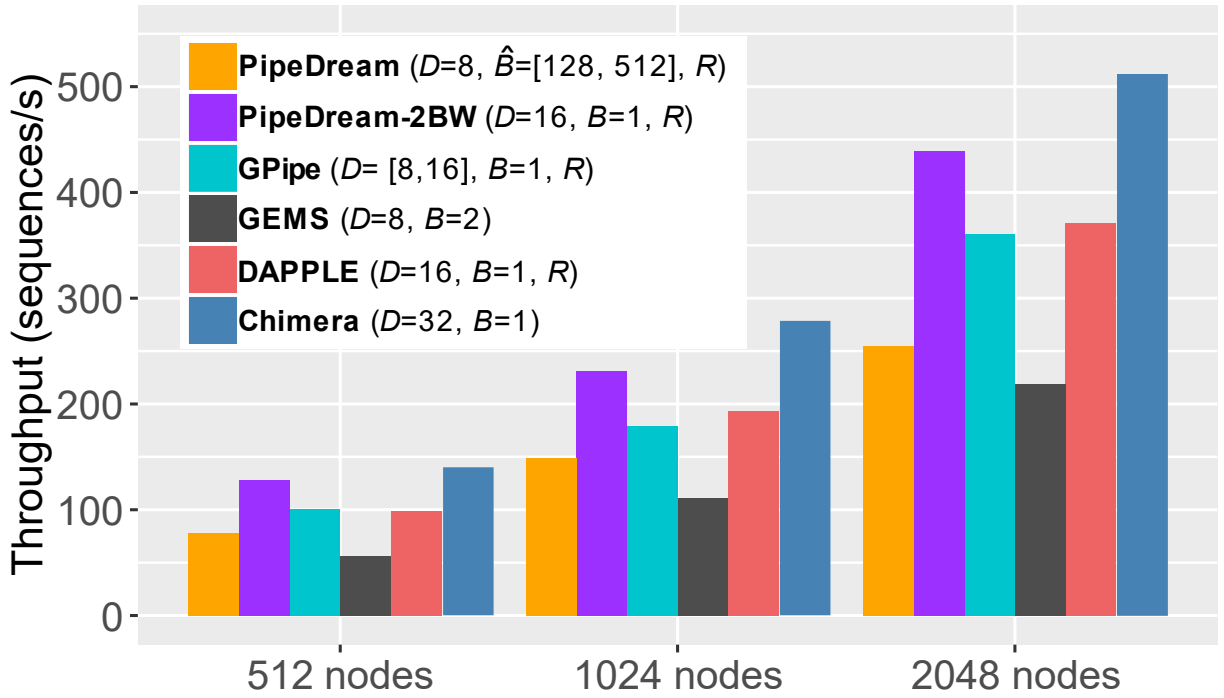
Bidirectional Pipelines – Meet Chimera (arXiv: 2107.06925v3)





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)

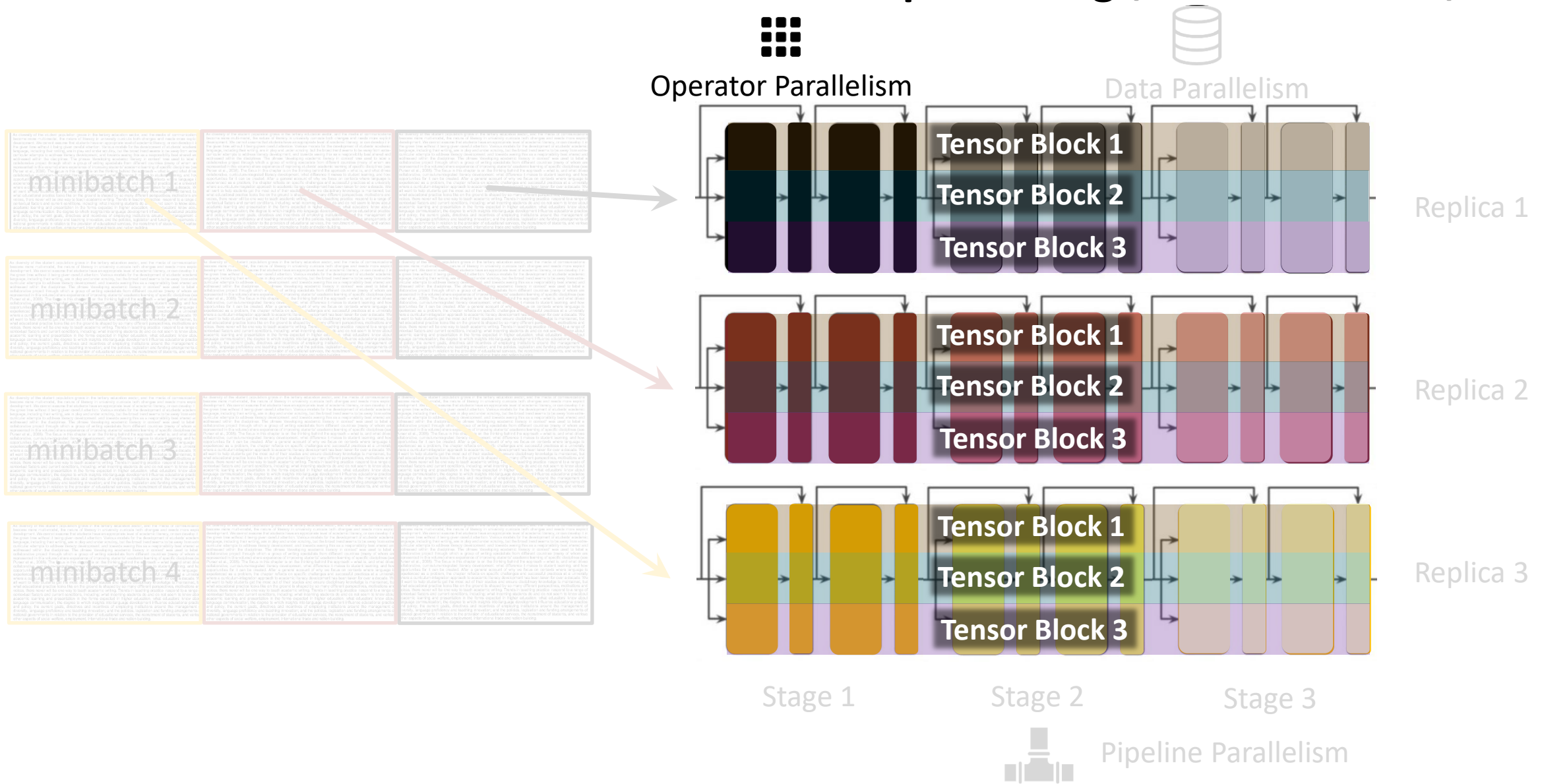


Weak scaling for GPT-2 on Piz Daint
(512 to 2048 GPU nodes)

- **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)



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

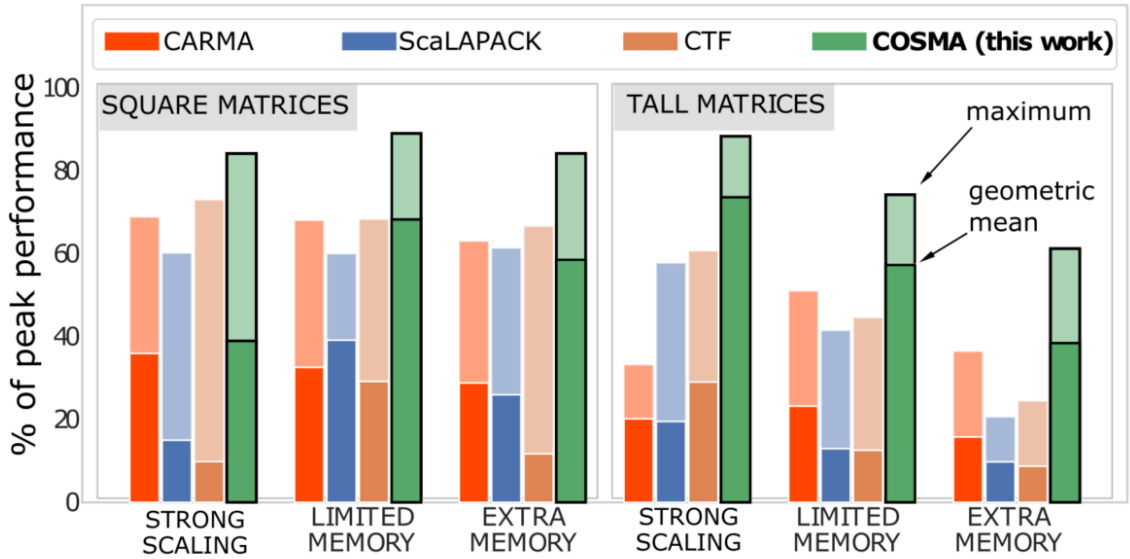
- **Large MMMs dominate large language models!**
  - e.g., GPT-3 multiples 12,288x12,288 matrices
    600 MiB in fp32 and 1.9 Tflop
  - generative inference multiplies tall & skinny matrices

- **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**

Remember those?
All MMM!

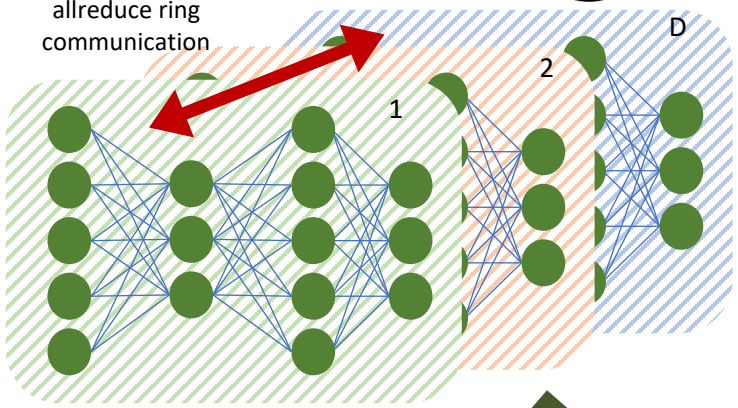
Operator class	% flop	% Runtime
Tensor contraction	99.80	61.0
Statistical normalization	0.17	25.5
Element-wise	0.03	13.5



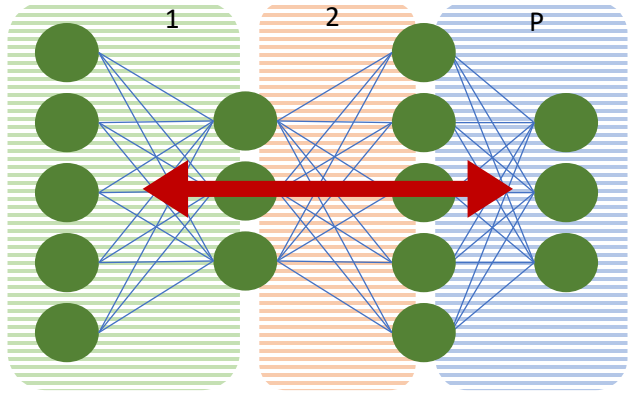
[1] G. Kwasniewski et al.: "Red-Blue Pebbling Revisited: Near Optimal Parallel Matrix-Matrix Multiplication", best student paper at Supercomputing SC19
[2] J. Chen et al.: "AutoDDL: Automatic Distributed Deep Learning with Asymptotically Optimal Communication", arXiv

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

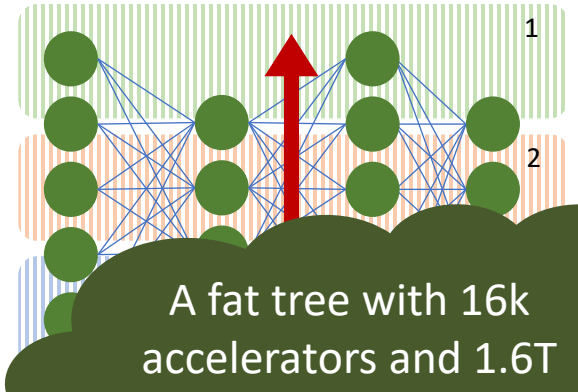
Data Parallelism



Pipeline Parallelism



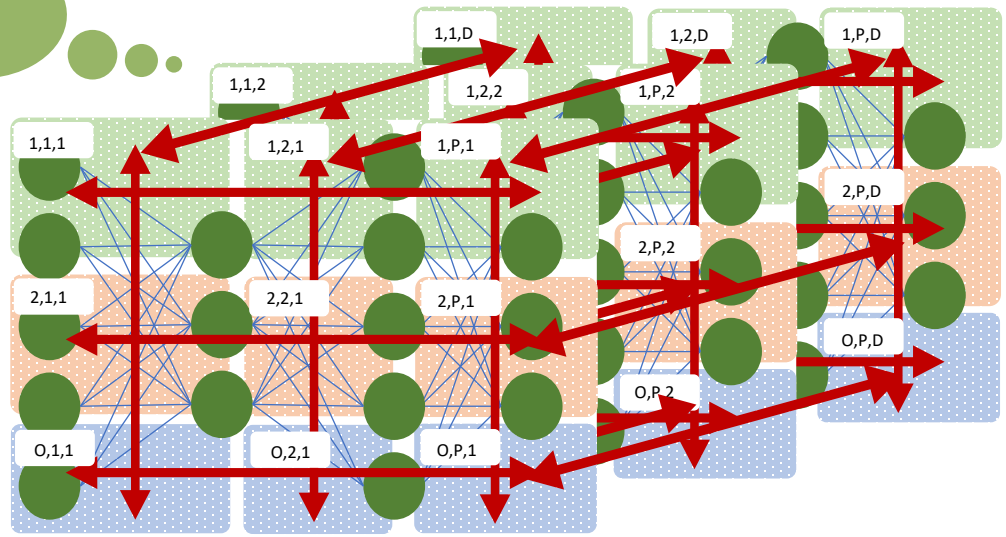
Operator Parallelism



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



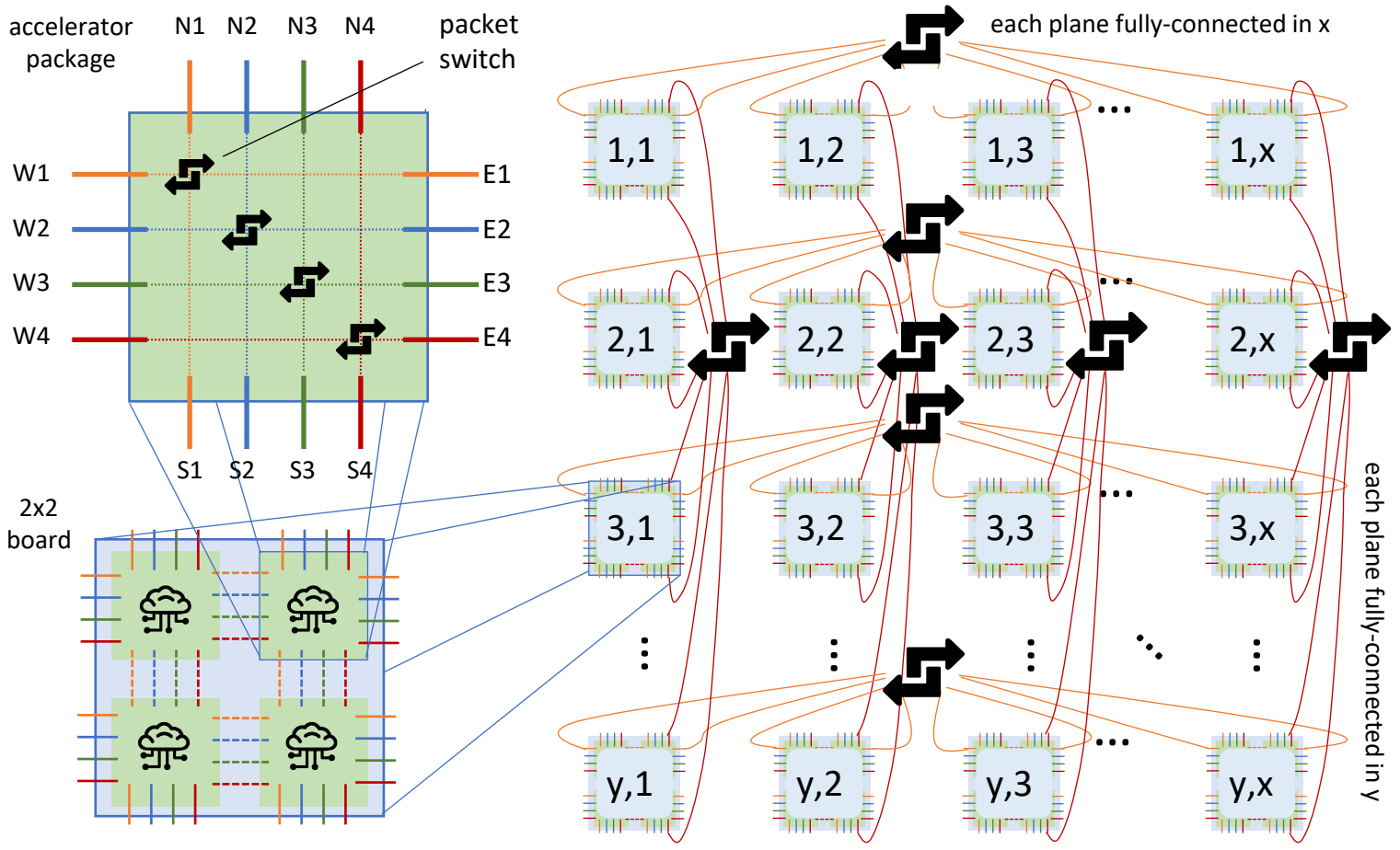
AI bandwidth today / yesterday (and growing!)

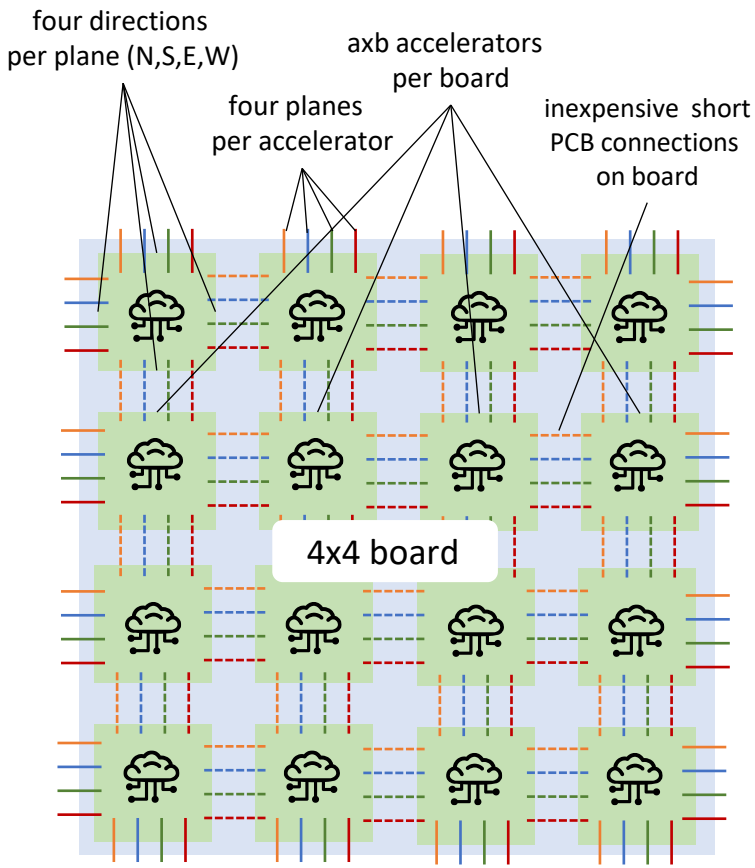
- Google TPUv2 (‘21): 1T
- AWS Trainium (‘21): 1.6T
- DGX-2 (A100, ‘21): 4.8T (islands of NVLINK)
- Tesla Dojo (‘22): 128T

→ 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)

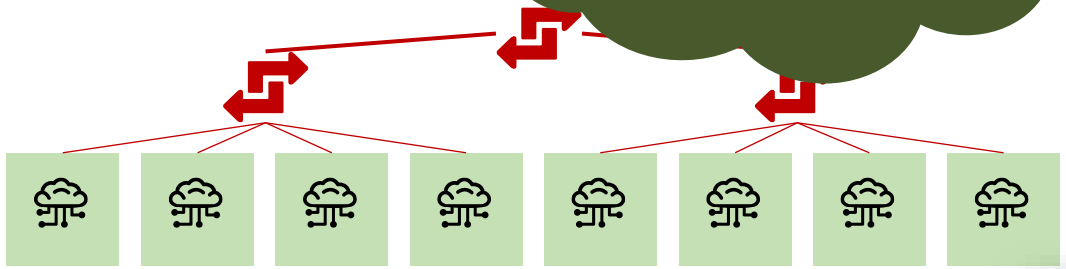
Bandwidth-cost-flexibility Tradeoffs (arXiv:2209.01346)

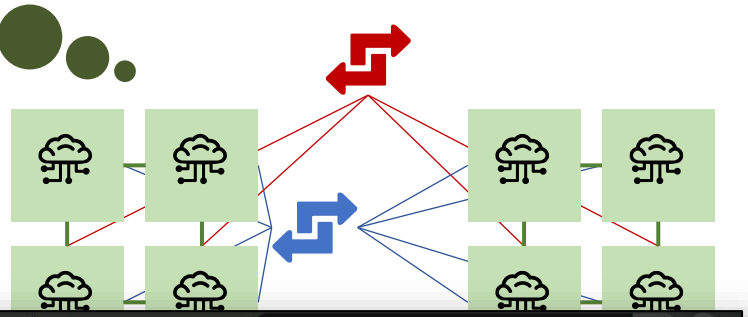
Global T
(e.g.

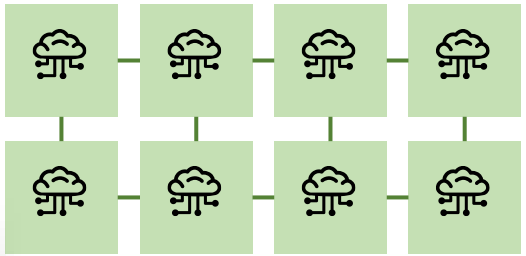
10-15x cheaper bandwidth than today's topologies!

HammingMesh
(many configurations)

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





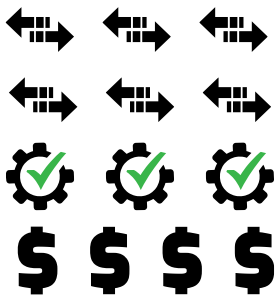


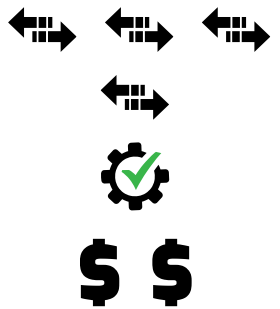
(large) reduce bandwidth

global bandwidth

placement flexibility

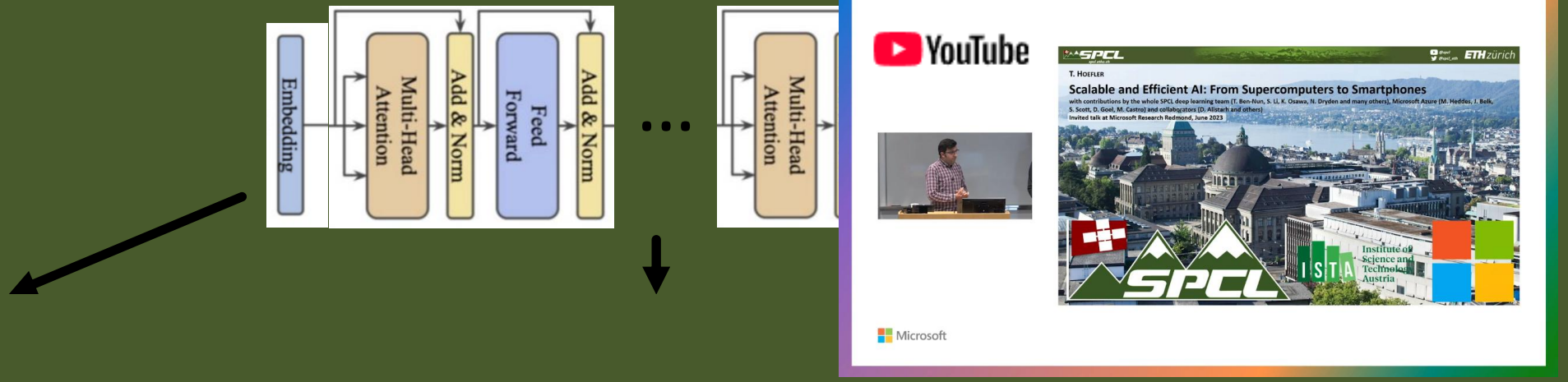
injection bandwidth





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

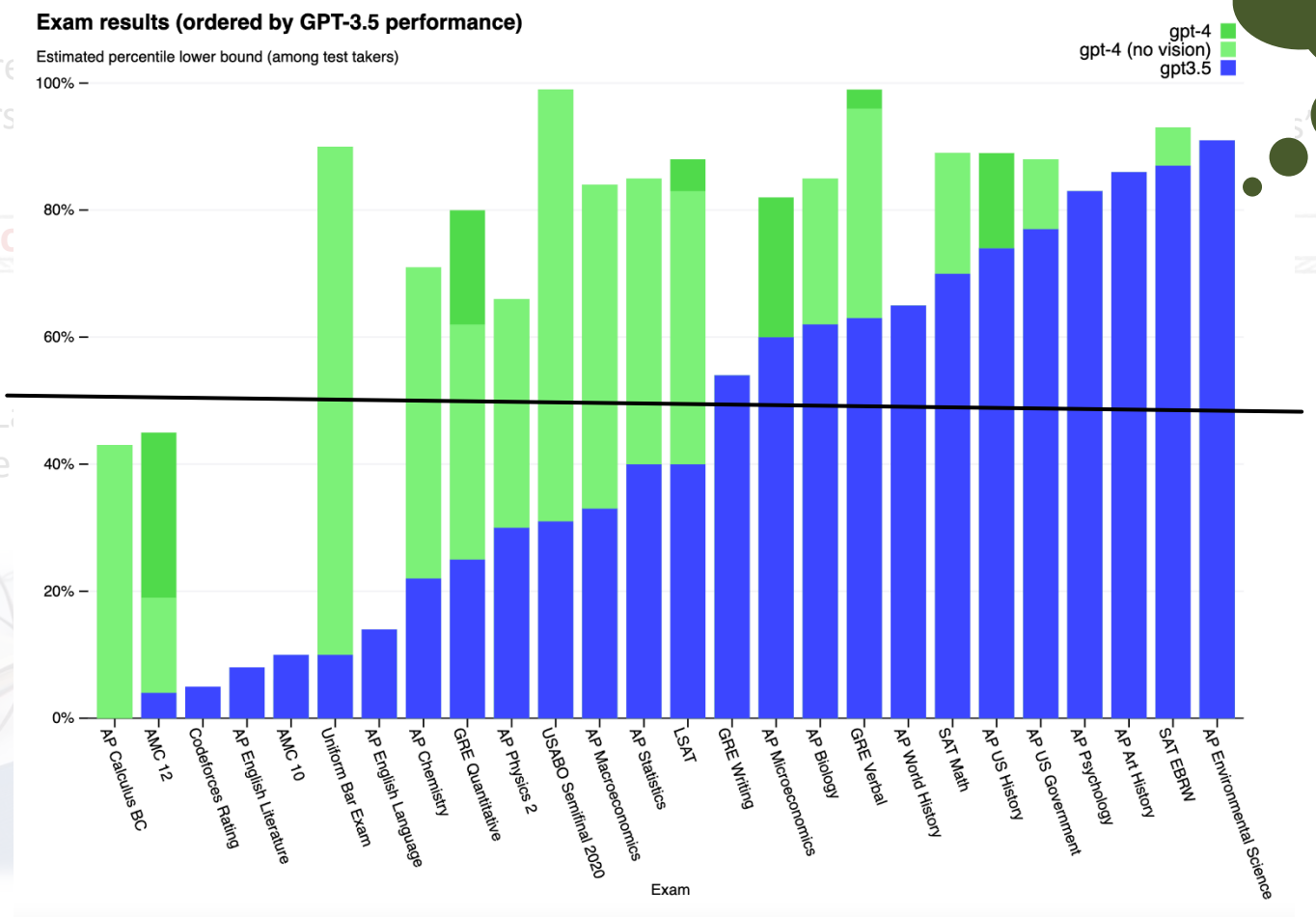
What now?

From LLMs to AHI

Ilya Sutskever

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

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

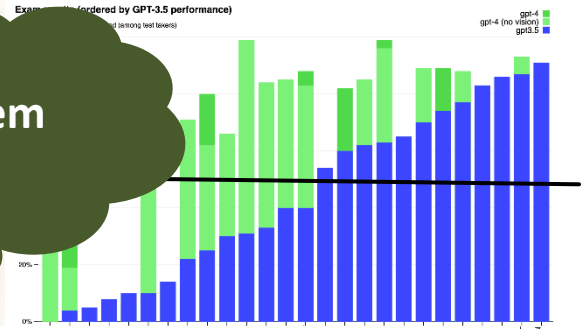


From LLMs to AHI

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"

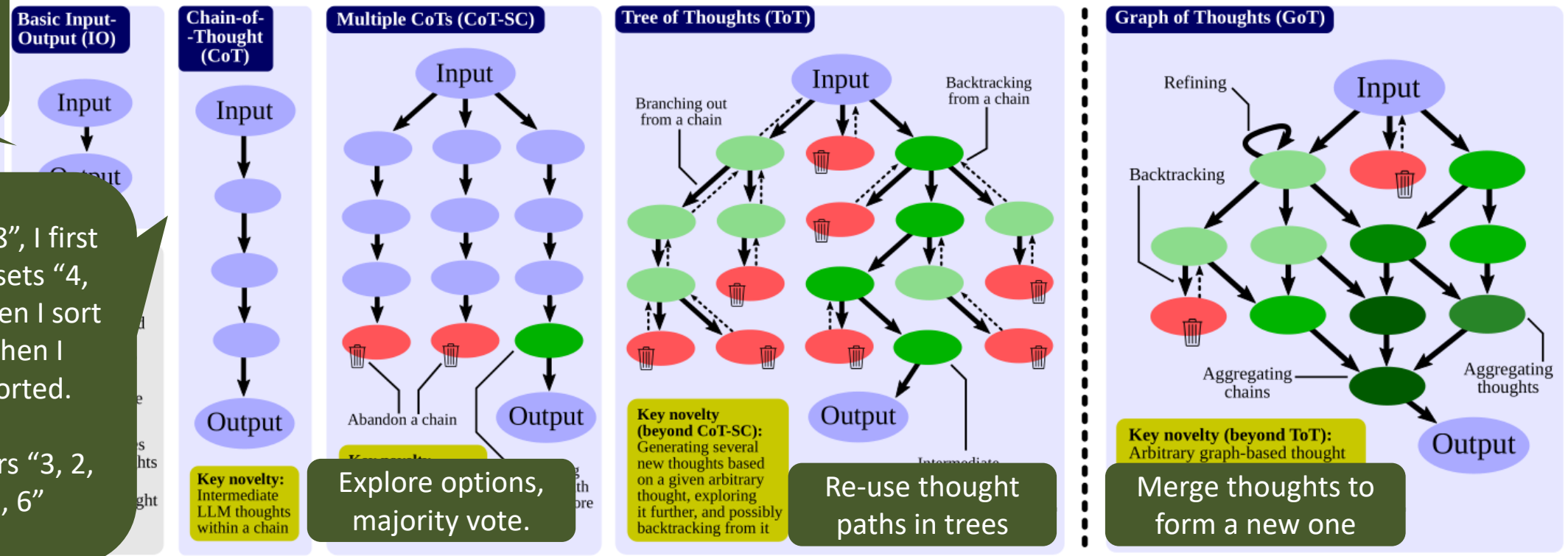
[Wang et al., March'22]

[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

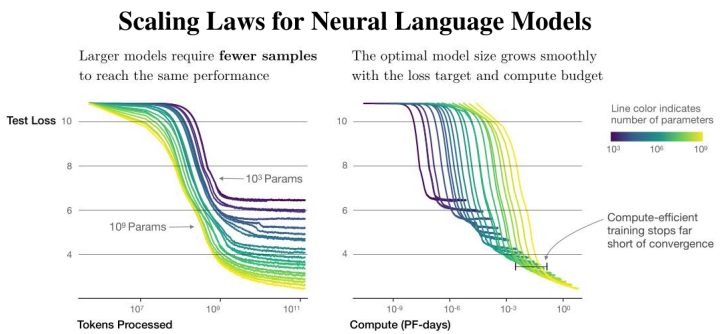
[Lei et al., August'23]



The Age of Computation

From LLMs to AHI

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



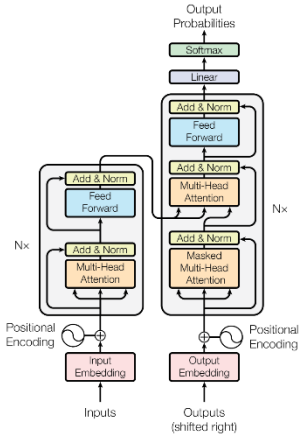
2020 - **GPT-3** (2020, scaling laws)
"Language Models are Few-Shot Learners"
37k+ citations

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

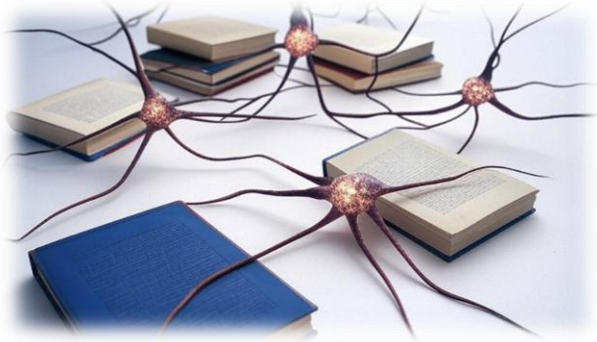
How to get to those "thoughts" (reasoning steps)

era of model size scaling → **era of data scaling**

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



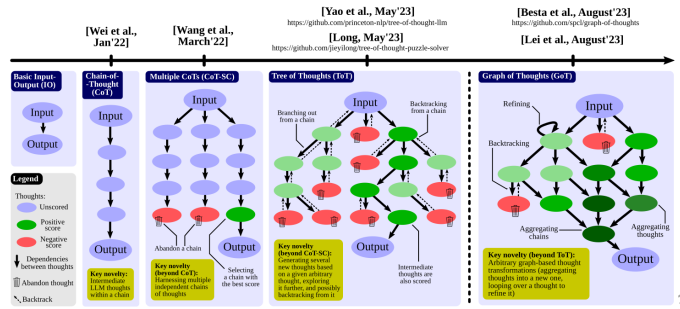
2019 - **GPT-2**
"Language Models are Unsupervised Multitask Learners"
14k+ citations



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)

How Google's AlphaGo Beat a Go World Champion

Inside a man-versus-machine showdown

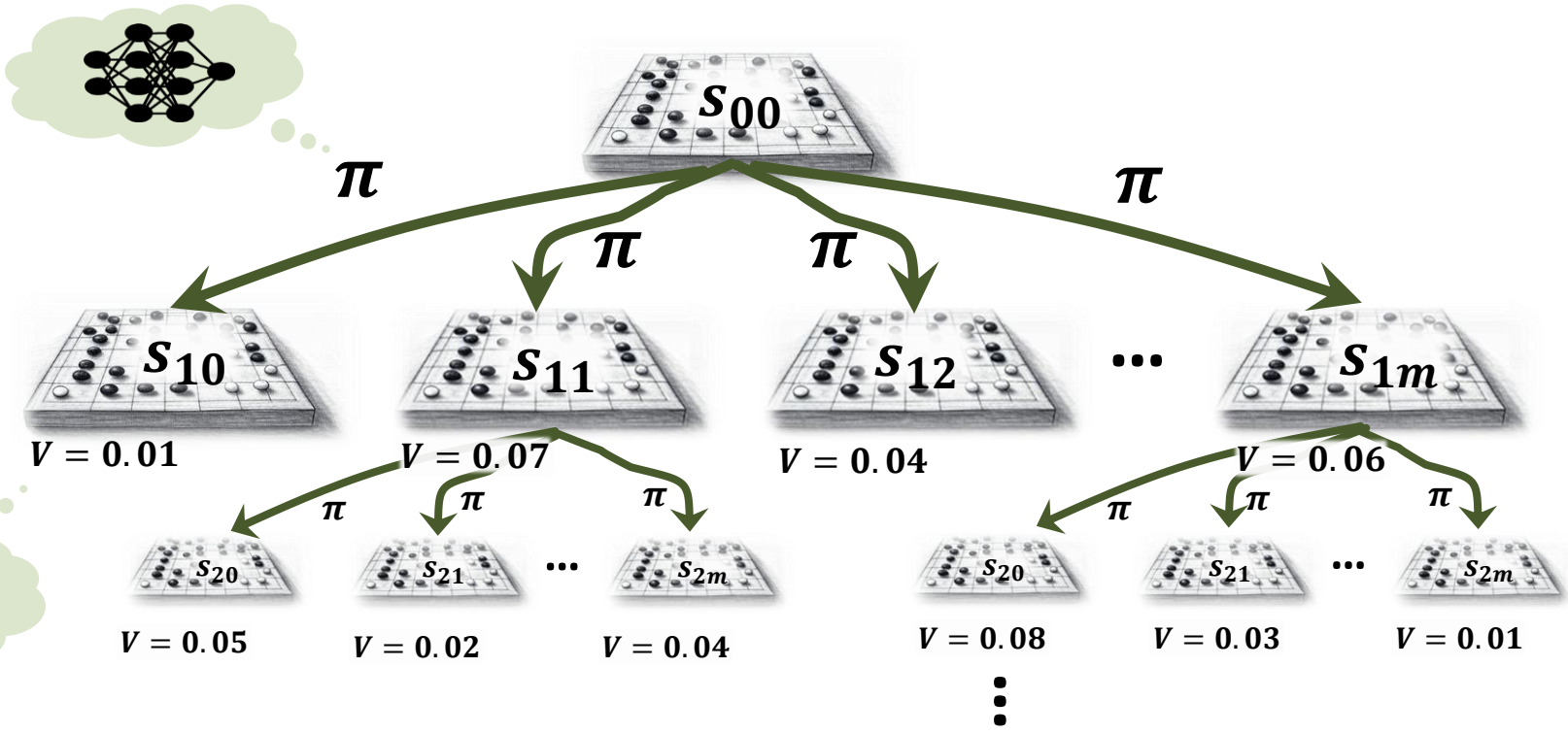
By Christopher Moyer

The Atlantic

18x champion

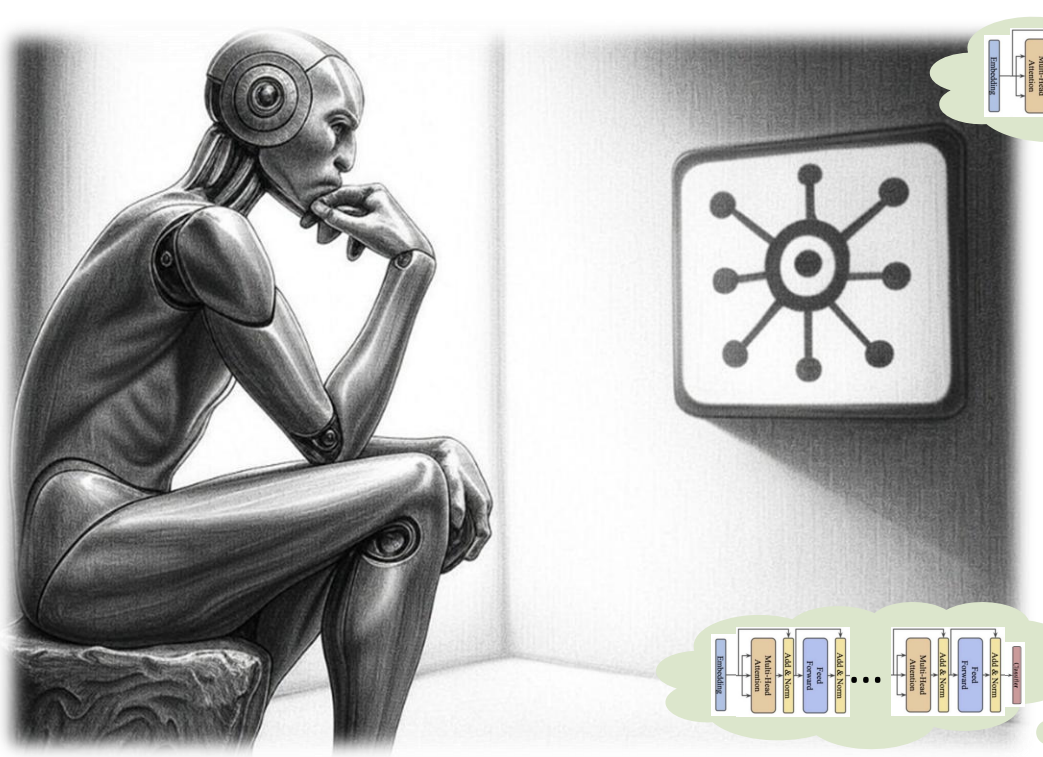
The South Korean professional Go player Lee Sedol reviews the match after finishing against Google's artificial AlphaGo. (Lee Jin-man / AP)

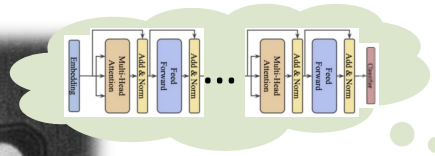
MARCH 28, 2016 SHARE SAVE



- **(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 Re 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

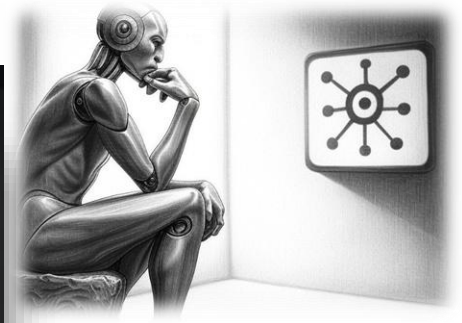
FrontierMath: A Benchmark for Evaluating Advanced Mathematical Reasoning in AI

IMPRESSIONS OF OUR RESEARCH-LEVEL PROBLEMS

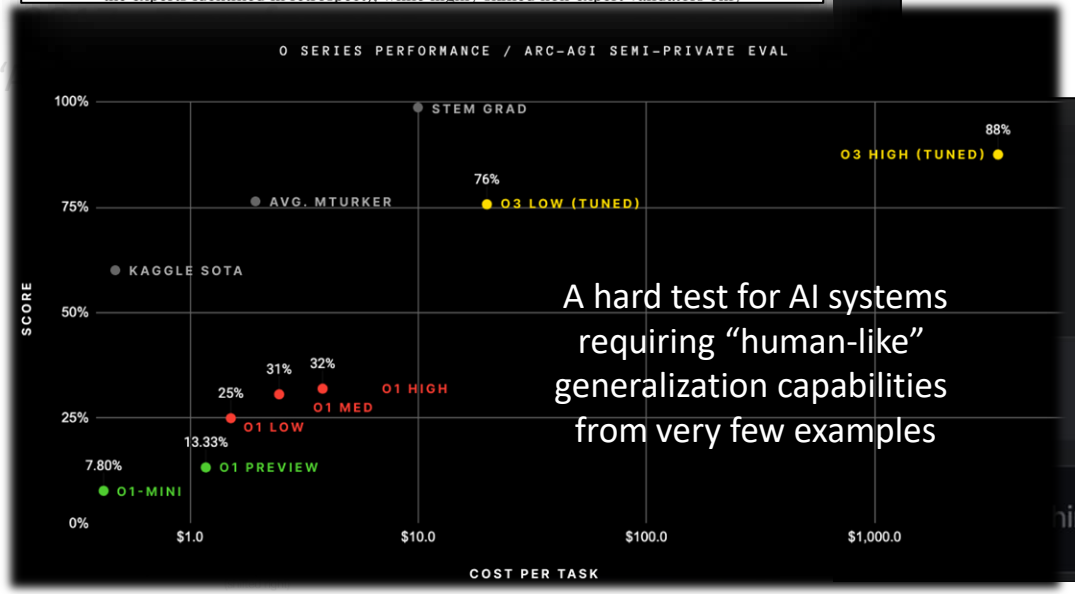
"These are extremely challenging... I think they will resist AIs for several years at least."

Nov.'24

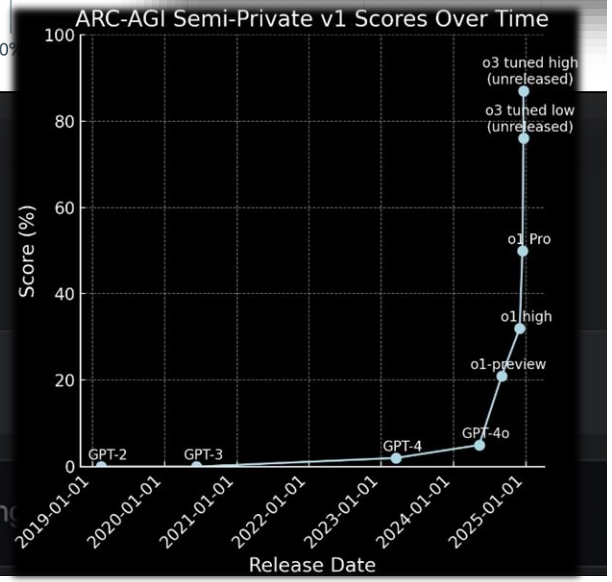
Terence Tao
Fields Medalist (2



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

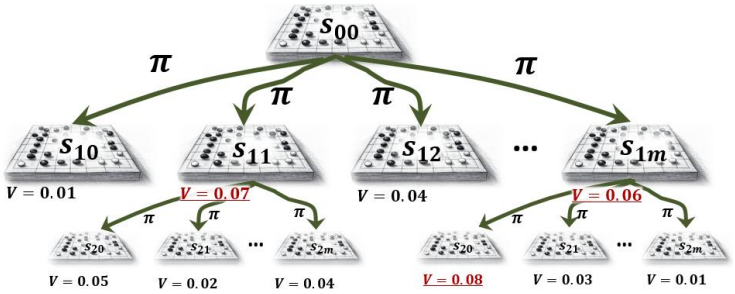


Codeforce Elo rating

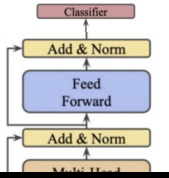
Elo-MMR	Title	Division	Number	Percentile	CF at same rank (spread)
3000+	Legendary Grandmaster	1	8	99.99	3382+
2700-2999	International Grandmaster	1	37	99.95	3010-3329 (372)
2400-2699	Grandmaster	1	255	99.7	2565-3010 (445)
2200-2399	International Master	1	560	99.1	2317-2565 (248)
2000-2199	Master	1	2089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

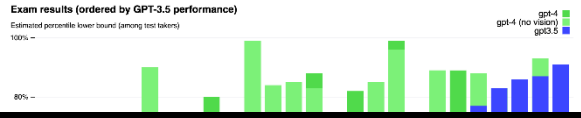
o3 achieves 2727 → 99.95th percentile of competitive programmers!

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!

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

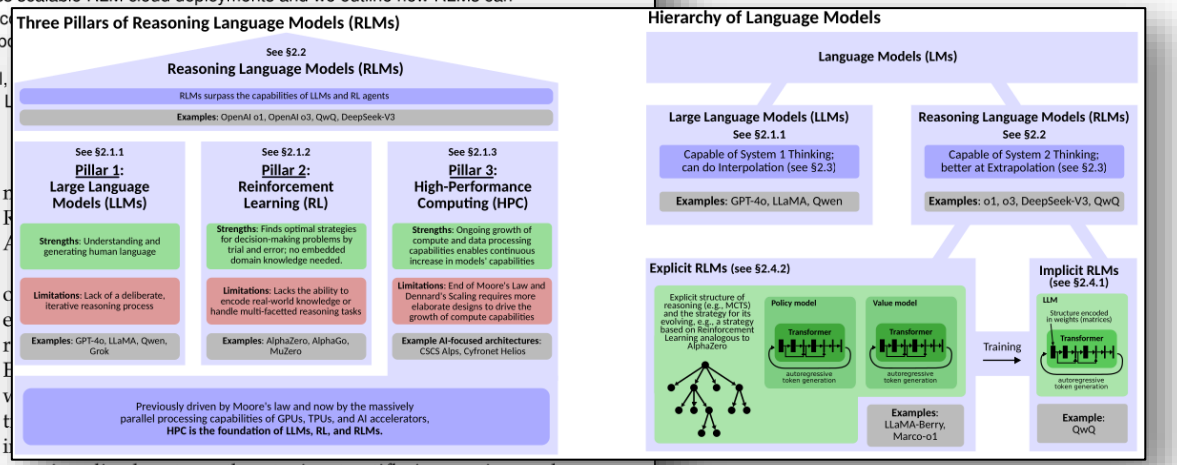
Abstract—Reasoning language models (RLMs), also known as Large Reasoning Models (LRMs), such as OpenAI's o1 and o3, DeepSeek-V3, and Alibaba's QwQ, have redefined AI's problem-solving capabilities by extending large language models (LLMs) with advanced reasoning mechanisms. Yet, their high costs, proprietary 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…

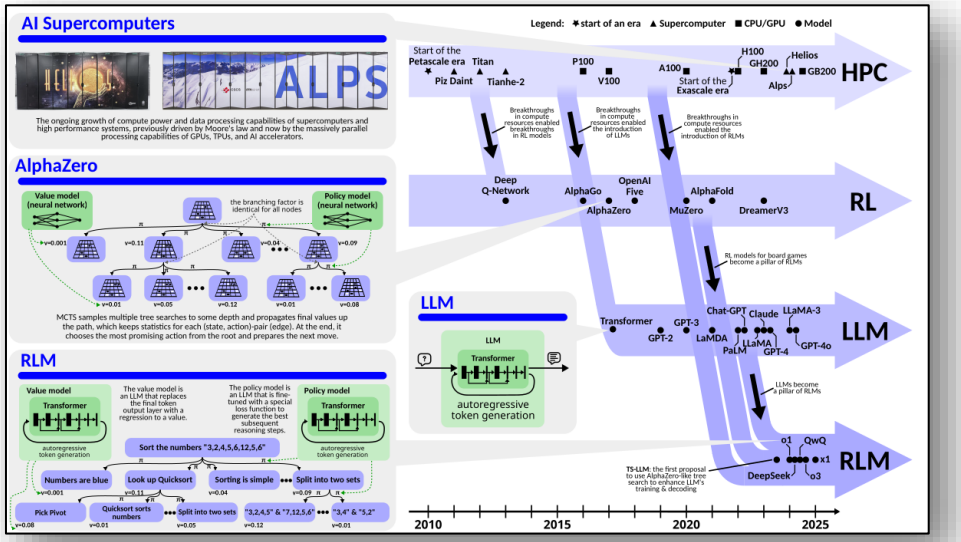
Index Terms—Reasoning Language Model, Large Reasoning Model, LRM, Reasoning LLMs, Reinforcement Learning for LLMs, MCTS for L…

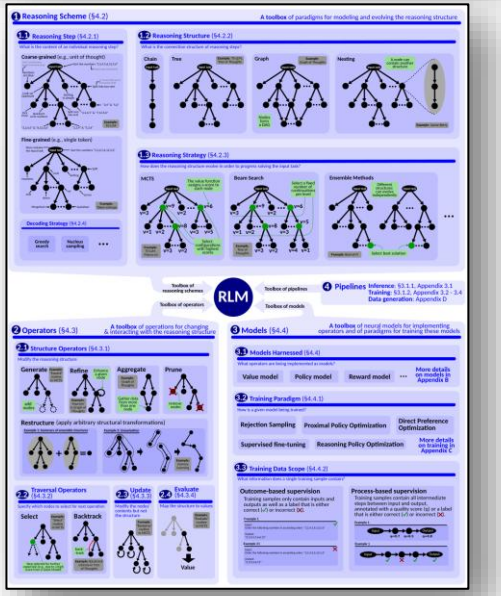
1 Introduction

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



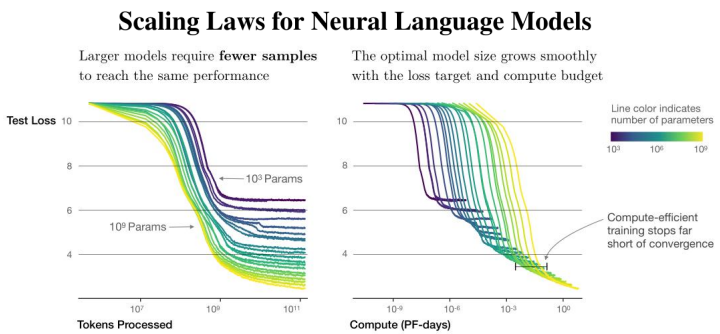




The Age of Computation

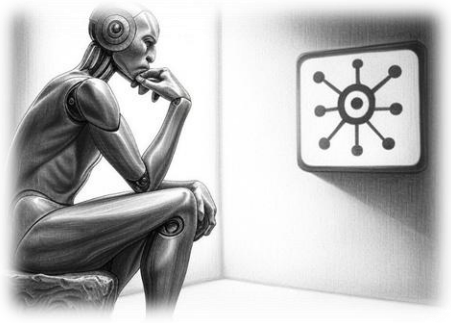
With RLMs to AHI

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

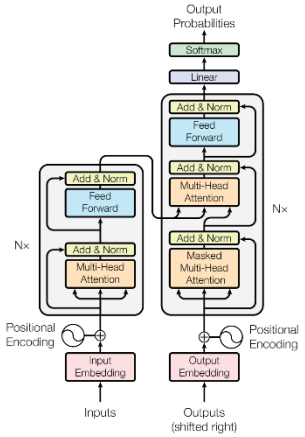
2023 – **Llama** (Qwen, Grok, etc.)
"LLaMA: Open and Efficient Foundation Language Models"
11k+ citations



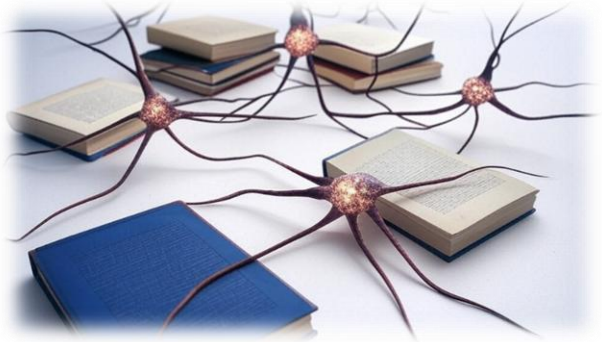
2024 – **Strawberry**
RL (o1, o3, etc.)
"Learning to Reason with LLMs"

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

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



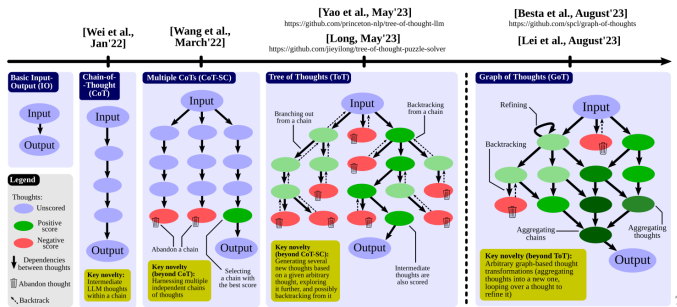
2019 - **GPT-2**
"Language Models are Unsupervised Multitask Learners"
14k+ citations



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

era of model size scaling

Efficient Language Models (ELMs)

era of reasoning scaling

from (pre)training scaling to inference scaling

Large Language Models (LLMs)

era of data scaling

Reasoning Language Models (RLMs)

$100

$50

/1M tokens

model size, (pre)training cost, inference cost

inference cost, more data, higher training cost

MUCH higher inference and training cost

GPT 3.5 $60

GPT 3.5 $20

GPT 3.5 Turbo $2

GPT 4 $120

GPT 4 Turbo $30

GPT 3.5 Turbo $1

GPT 4o $15

GPT 4o mini $0.6

GPT 4o $10

GPT o1 $60

GPT o1-mini $12

Dec. 21

Oct. 22

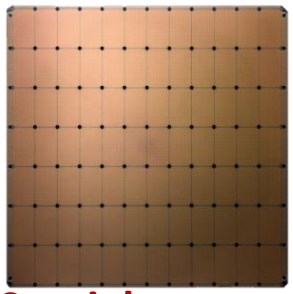
Mar. 23

Nov. 23

Aug. 24

Dec. 24

We need cheaper compute



Principles: Datatypes, Sparsity, Spatial

We need cheaper systems (networking!)

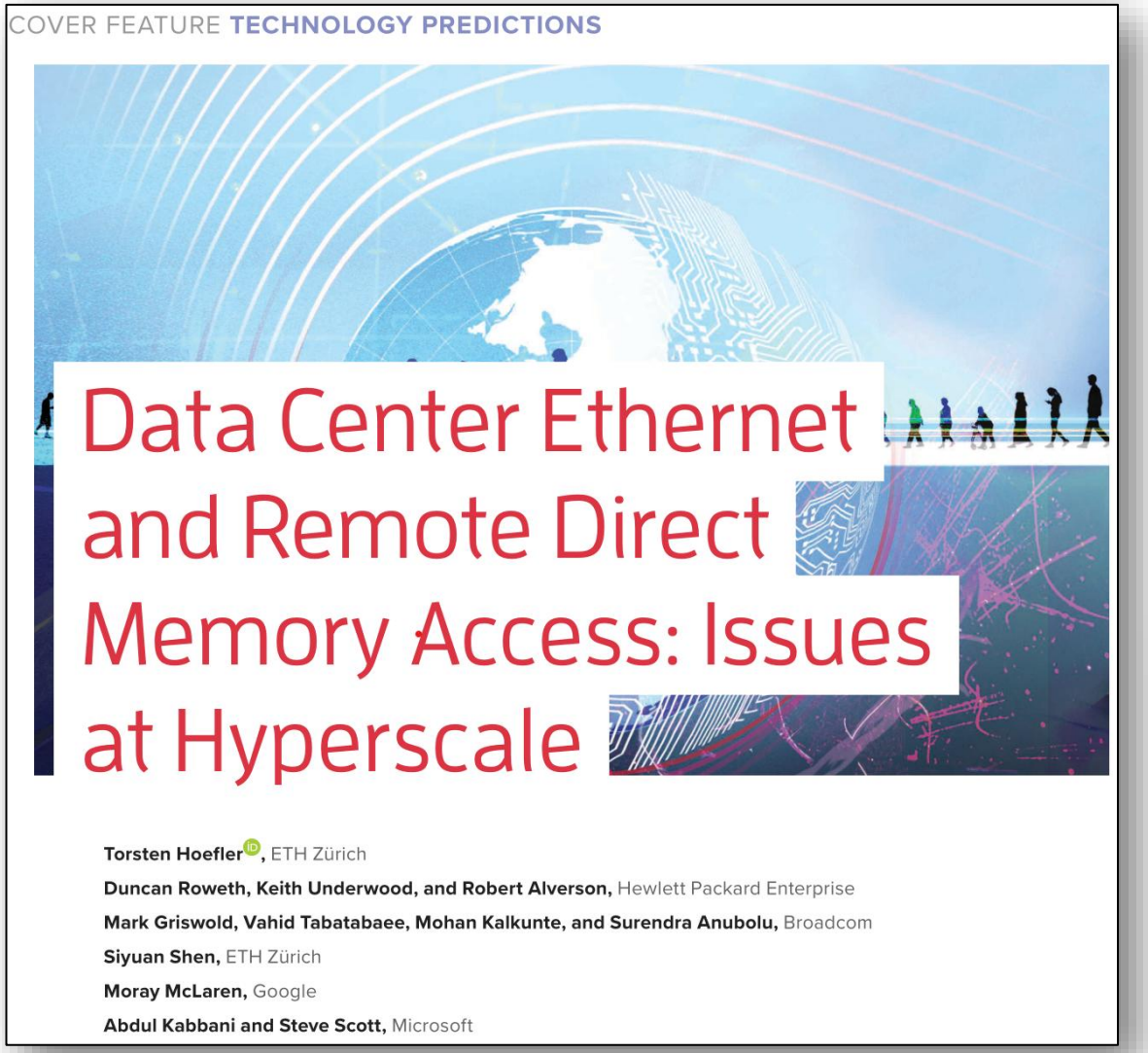
Ultra Ethernet Consortium

Principles: high local bandwidth, reliability, cost

Networks Converge

The Datacenter will be a Supercomputer

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

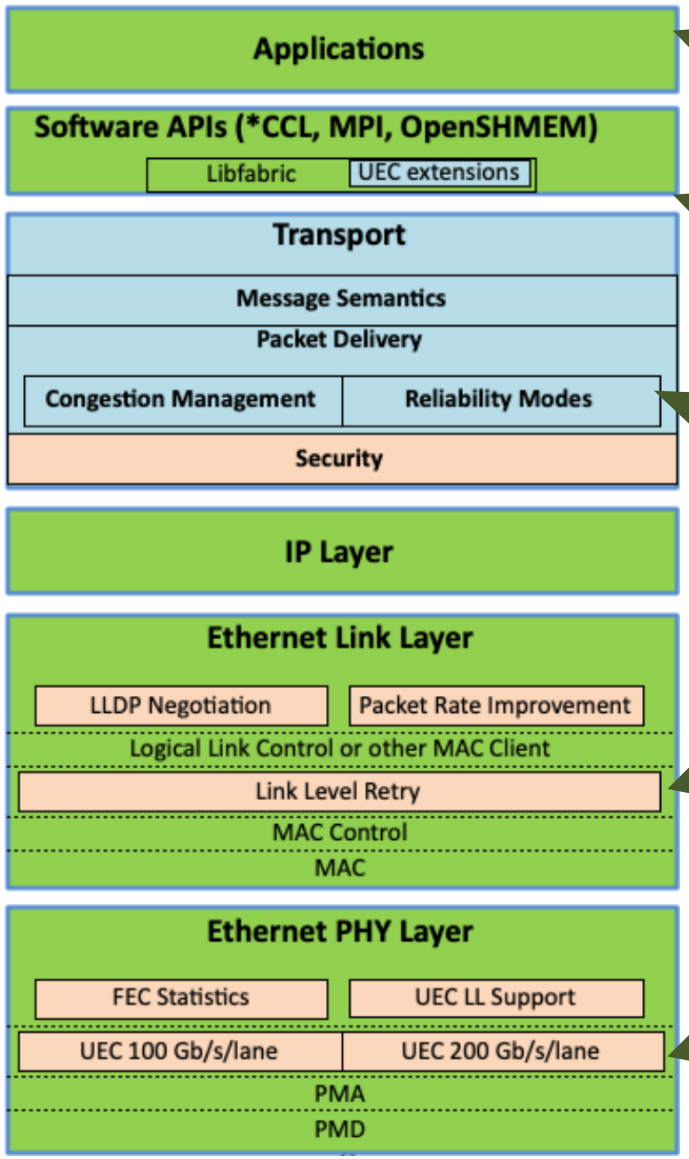
Ultra Ethernet Consortium

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

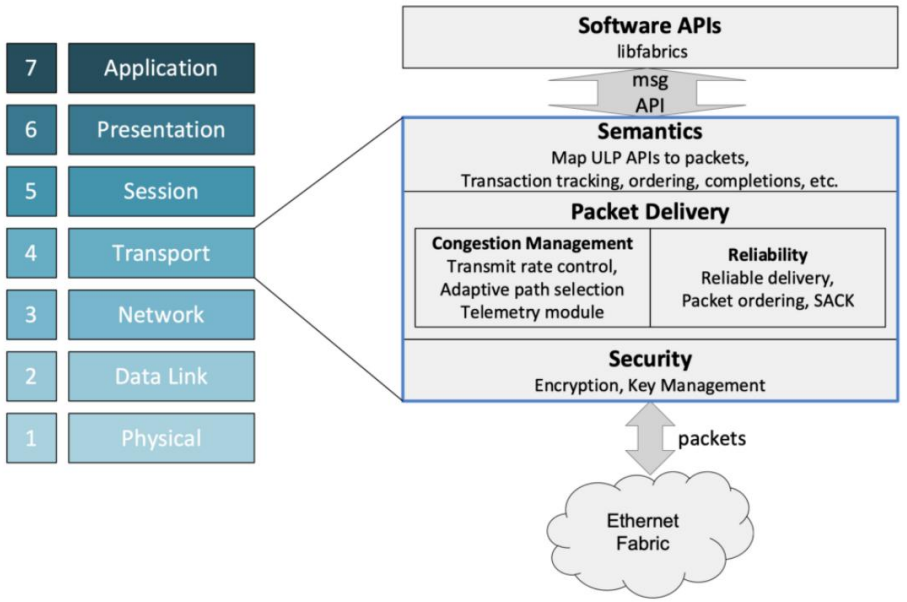
Libfabric + UE extensions

Here is where most of the innovation lies!

Some Ethernet extensions to benefit AI/HPC workloads

Close collaboration with IEEE 802.3 standards body

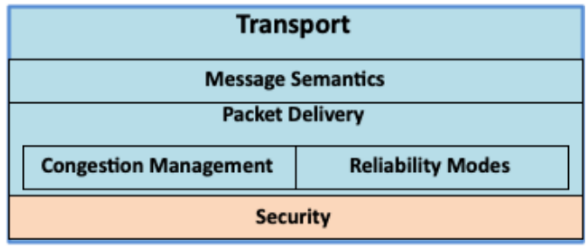
UE enables cheap high-performance **hardware implementations** of an **optimized transport** over (legacy) **Ethernet networks** while **enabling vendor innovation**



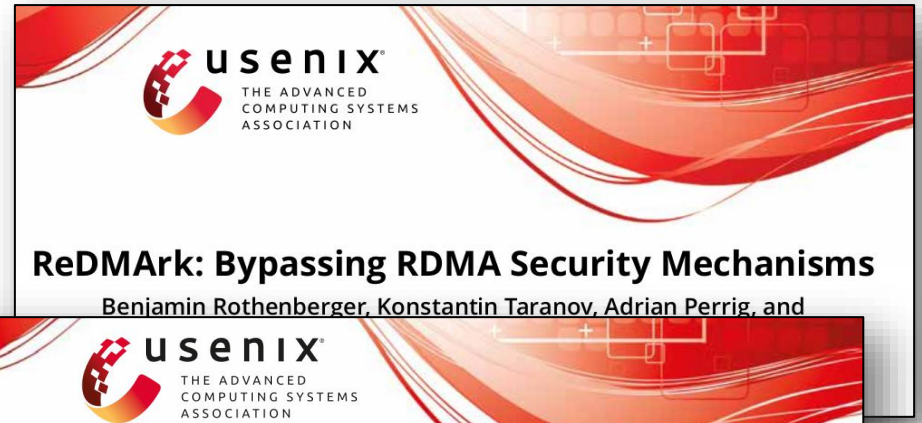
https://ultraethernet.org/uec-progresses-towards-v1-0-set-of-specifications/

Ultra Ethernet's key features compared RoCE and others

Ultra Ethernet Consortium



- **Lossy (best effort) & lossless operation**
  - Solves all PFC/blocking issues!
- **Flexible ordering in packet and message delivery**
  - Reliable Ordered Delivery (ROD), Reliable Unordered Delivery (RUD/RUDI), Unreliable Unordered Delivery (UUD)
- **"State of the art" (2024), easily configured congestion control mechanisms**
  - Sender and Receiver-based mechanisms over lossy networks
  - Supports trimming extensions
- **Multi-path packet spraying**
  - Adaptive routing with ordering using existing switches (ECMP), zero copy
- **Switch offload (i.e., In-Network Collectives)**
  - Cheap & effective
- **Security as a first-class citizen co-designed with the transport**
  - Addressing issues in RoCE
- **Ethernet Link and Physical layer enhancements (optional)**
  - See previous slide

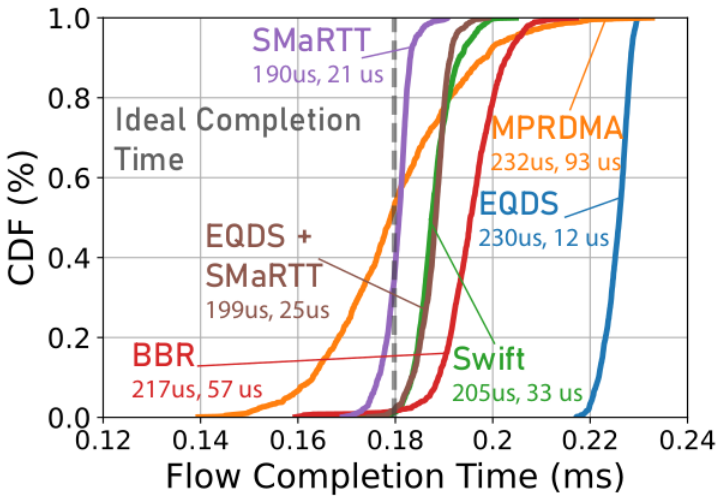


https://ultraethernet.org/uec-progresses-towards-v1-0-set-of-specifications/

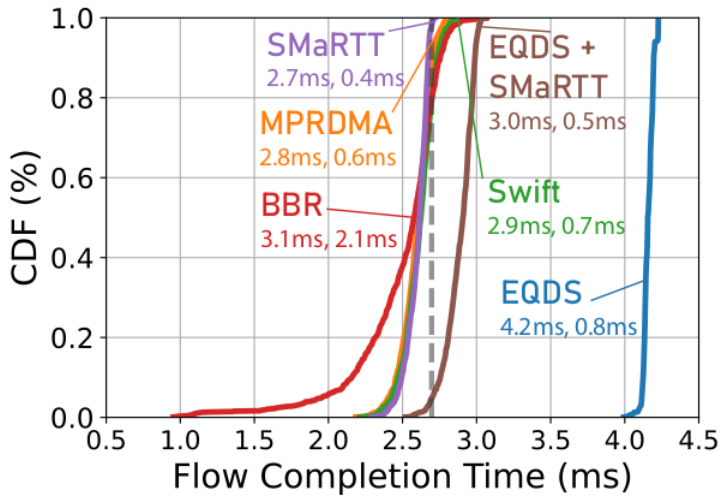
SMaRTT-REPS enables Modern Packet Spraying

Ultra Ethernet Consortium

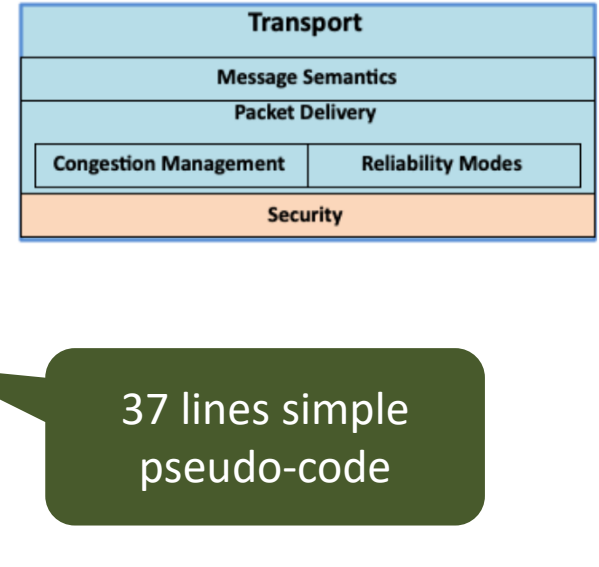
- **"State of the art" (2024), easily configured congestion control mechanisms**

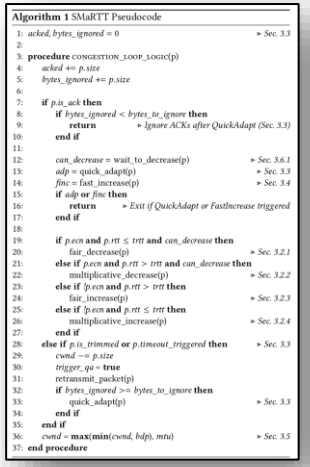


2 MiB Flows



32 MiB Flows



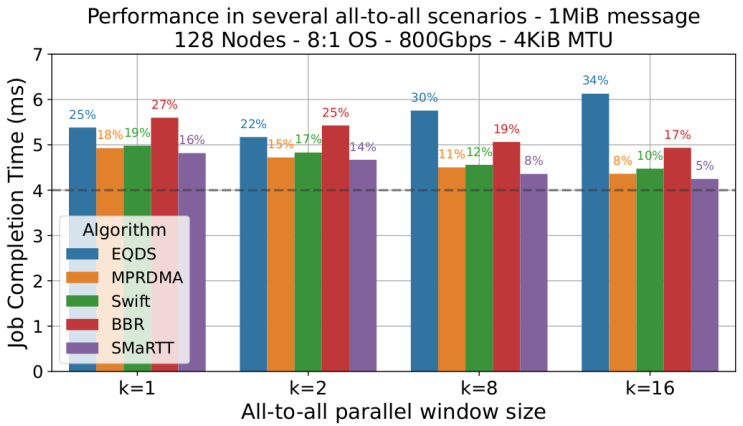


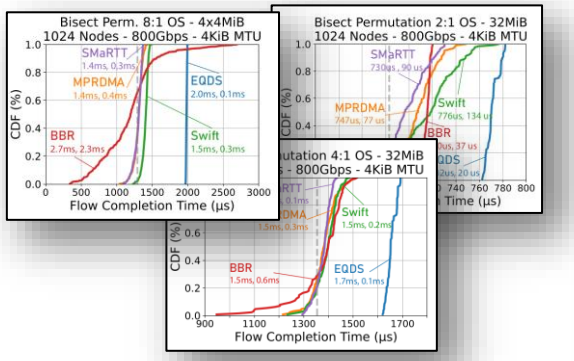
Algorithm 1 SMaRTT Pseudocode

```
acked, bytes_ignored = 0                                   ▸ Sec. 3.3

procedure CONGESTION_LOOP_LOGIC(p)
    acked += p.size
    bytes_ignored += p.size

    if p.is_ack then
        if bytes_ignored < bytes_to_ignore then
            return                  ▸ Ignore ACKs after QuickAdapt (Sec. 3.3)
        end if

        can_decrease = wait_to_decrease(p)                 ▸ Sec. 3.6.1
        adp = quick_adapt(p)                               ▸ Sec. 3.3
        finc = fast_increase(p)                            ▸ Sec. 3.4
        if adp or finc then
            return          ▸ Exit if QuickAdapt or FastIncrease triggered
        end if

        if p.ecn and p.rtt ≤ trtt and can_decrease then
            fair_decrease(p)                               ▸ Sec. 3.2.1
        else if p.ecn and p.rtt > trtt and can_decrease then
            multiplicative_decrease(p)                     ▸ Sec. 3.2.2
        else if !p.ecn and p.rtt > trtt then
            fair_increase(p)                               ▸ Sec. 3.2.3
        else if !p.ecn and p.rtt ≤ trtt then
            multiplicative_increase(p)                     ▸ Sec. 3.2.4
        end if
    else if p.is_trimmed or p.timeout_triggered then       ▸ Sec. 3.3
        cwnd −= p.size
        trigger_qa = true
        retransmit_packet(p)
        if bytes_ignored >= bytes_to_ignore then
            quick_adapt(p)                                 ▸ Sec. 3.3
        end if
    end if
    cwnd = max(min(cwnd, bdp), mtu)                        ▸ Sec. 3.5
end procedure
```

37 lines simple pseudo-code

Permutation traffic on 8:1 oversubscribed fat tree



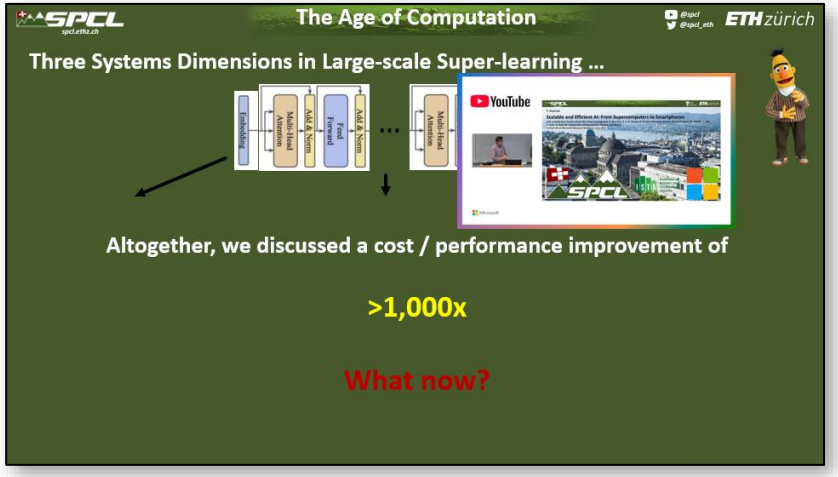


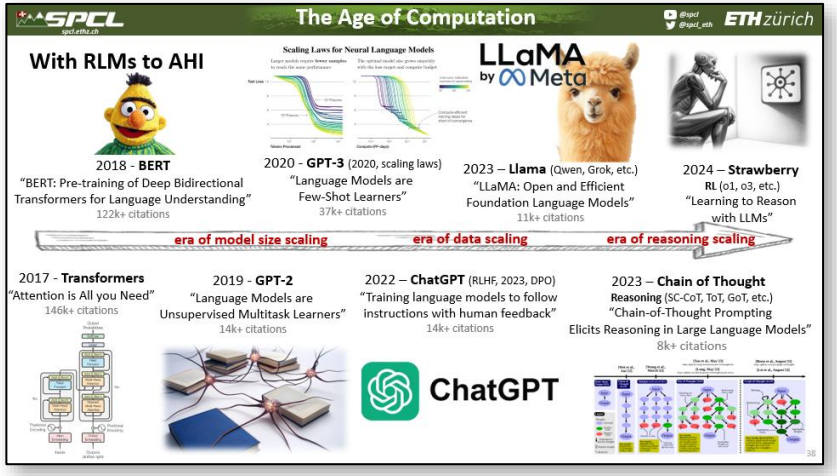
SMaRTT-REPS: Sender-based Marked Rapidly-adapting Trimmed & Timed Transport with Recycled Entropies

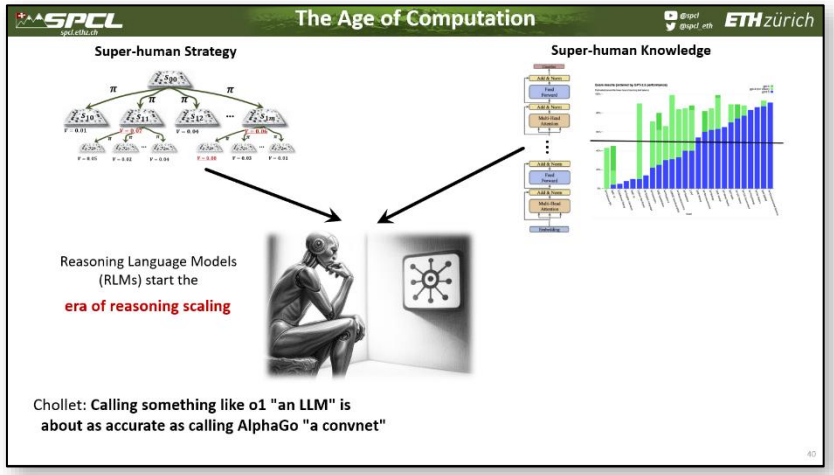
Tommaso Bonato — ETH Zürich, Microsoft
Abdul Kabbani — Microsoft
Daniele De Sensi — Sapienza University of Rome
Rong Pan — AMD
Yanfang Le — AMD
Costin Raiciu — Broadcom Inc.
Mark Handley — Broadcom Inc.
Timo Schneider — ETH Zürich
Nils Blach — ETH Zürich
Ahmad Ghalayini — Microsoft
Daniel Alves — Microsoft
Michael Papamichael — Microsoft
Adrian Caulfield — Microsoft
Torsten Hoefler — ETH Zürich, Microsoft

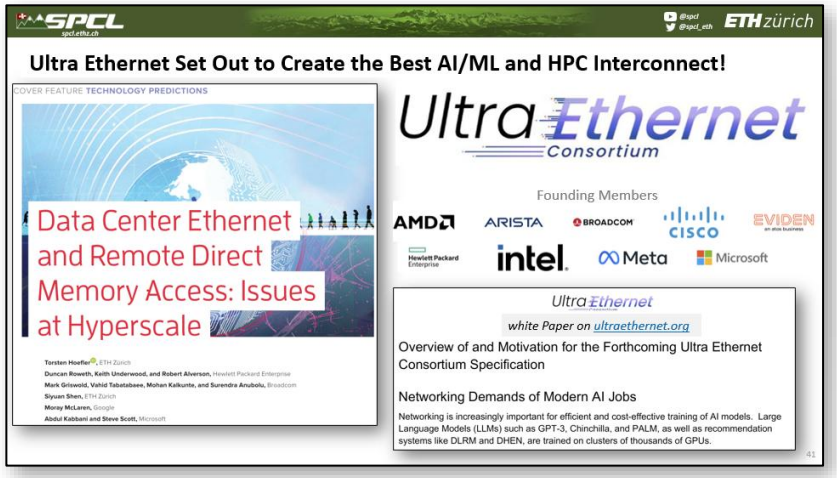
Bonato et al.: SMaRTT-REPS: Sender-based Marked Rapidly-adapting Trimmed & Timed Transport with Recycled Entropies, arXiv 2404.01630

Key Points and Conclusions









More of SPCL's research:

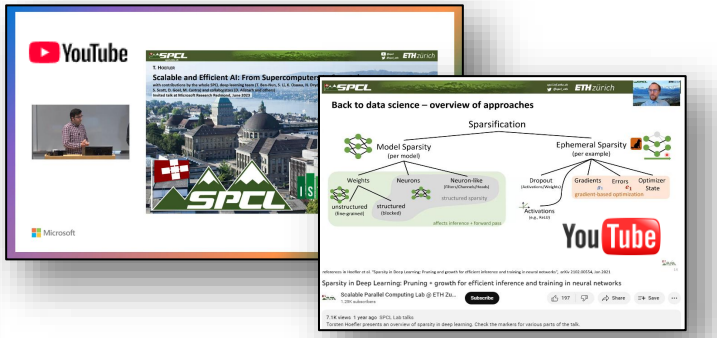
youtube.com/@spcl — 210+ Talks

twitter.com/spcl_eth — 1.4K+ Followers

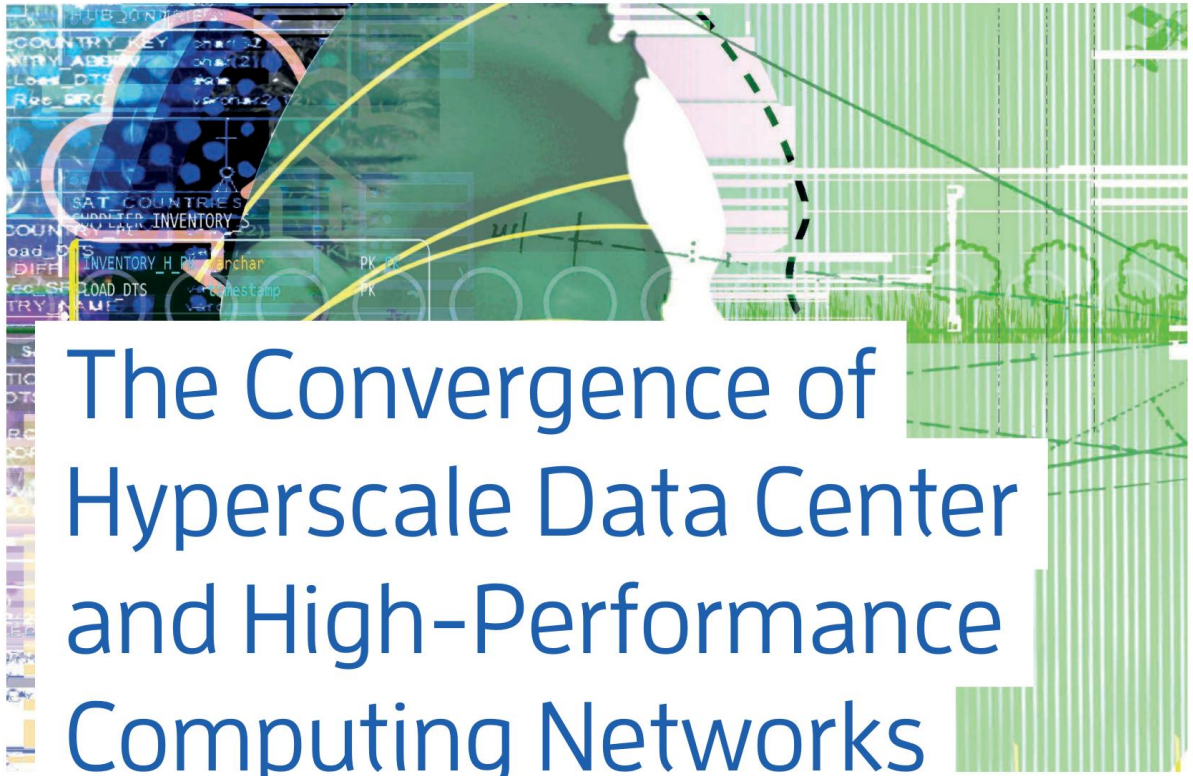
github.com/spcl — 2K+ Stars

… or spcl.ethz.ch

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



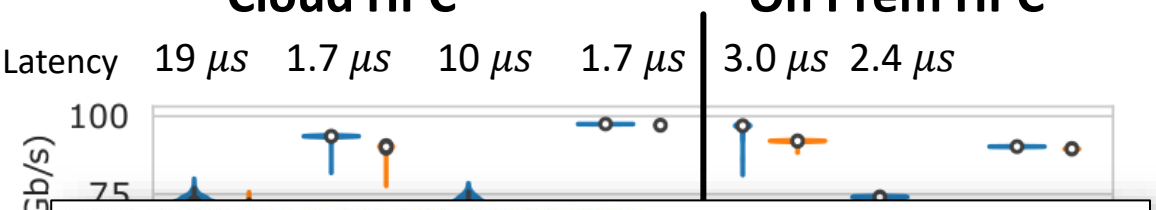
COVER FEATURE TECHNOLOGY PREDICTIONS



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

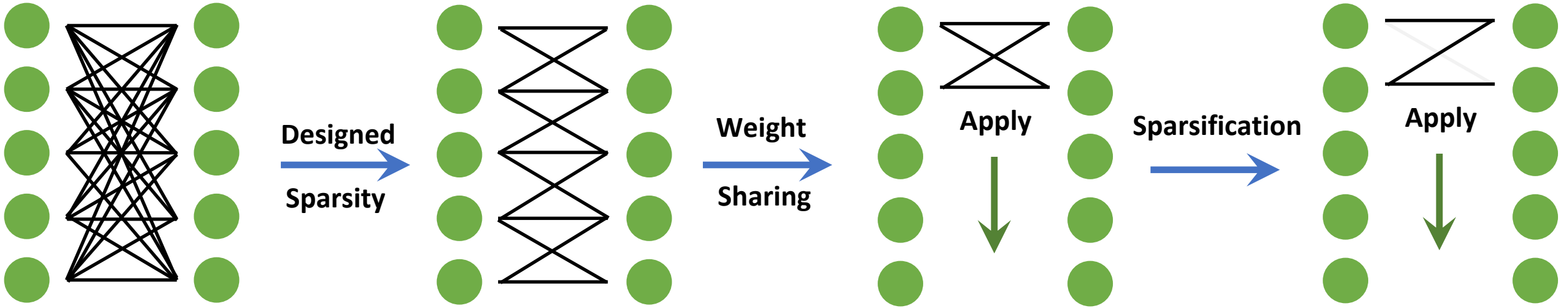
Torsten Hoefler, ETH Zürich
Duncan Roweth, Keith Underwood, and Robert Alverson, Hewlett Packard Enterprise
Mark Griswold, Vahid Tabatabaee, Mohan Kalkunte, and Surendra Anubolu, Broadcom
Siyuan Shen, ETH Zürich
Moray McLaren, Google
Abdul Kabbani and Steve Scott, Microsoft

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

An example to relate to – CNNs from a sparsity viewpoint

"With all things being equal, the simplest explanation tends to be the right one"

- William of Ockham, ~1300



Fully Connected

Locally Connected

Convolutional

Sparse Convolutional

universal approximation
hard to train

inductive locality bias
reduce training complexity

inductive translational
equivariance bias
image recognition/object detection

sparsification
reduce representational
complexity (MDL, Occam)

references in Hoefler et al. "Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks", arXiv 2102.00554, Jan 2021

Performance and storage overheads of sparsity

dense
[0,2,0,0,3,4,0,0,0,0,0,5]

bitmap
[010011000001|2345]

runlength / delta
[1|2,2|3,0|4,5|5]

compressed sparse row / column
[1] [1|2,2|3,0|4,5|5]

coordinate offset
[1|2, 5|3, 6|4, 12|5]

0%	10%	70%	90%	99.9%	99.99999%
dense	low sparsity	medium sparsity	moderate sparsity	high sparsity	extreme

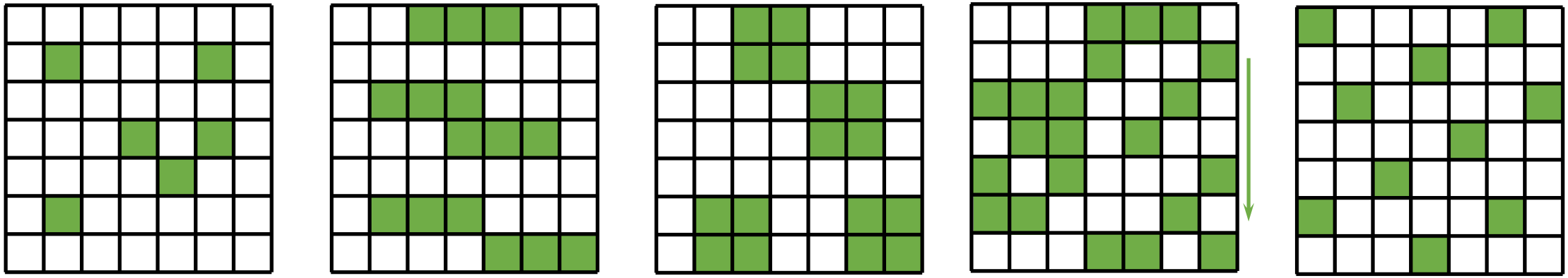
not (really) worth it

today's SoA
(equivalent accuracy)

today's SoA
(accuracy loss)

scientific computing
(tomorrow's DL?)

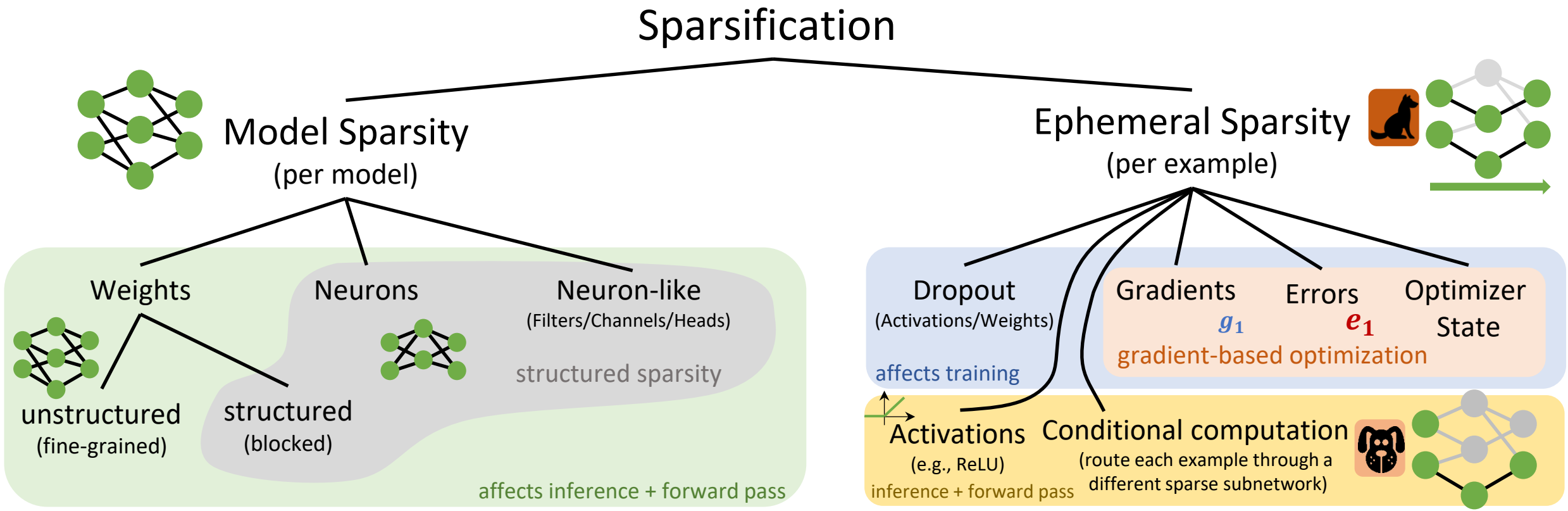
Structure matters at least as much as the sparsity level for overheads! Not a simple problem ☹



unstructured | 1d blocked | 2d blocked | block-balanced | strided

references in Hoefler et al. "Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks", arXiv 2102.00554, Jan 2021

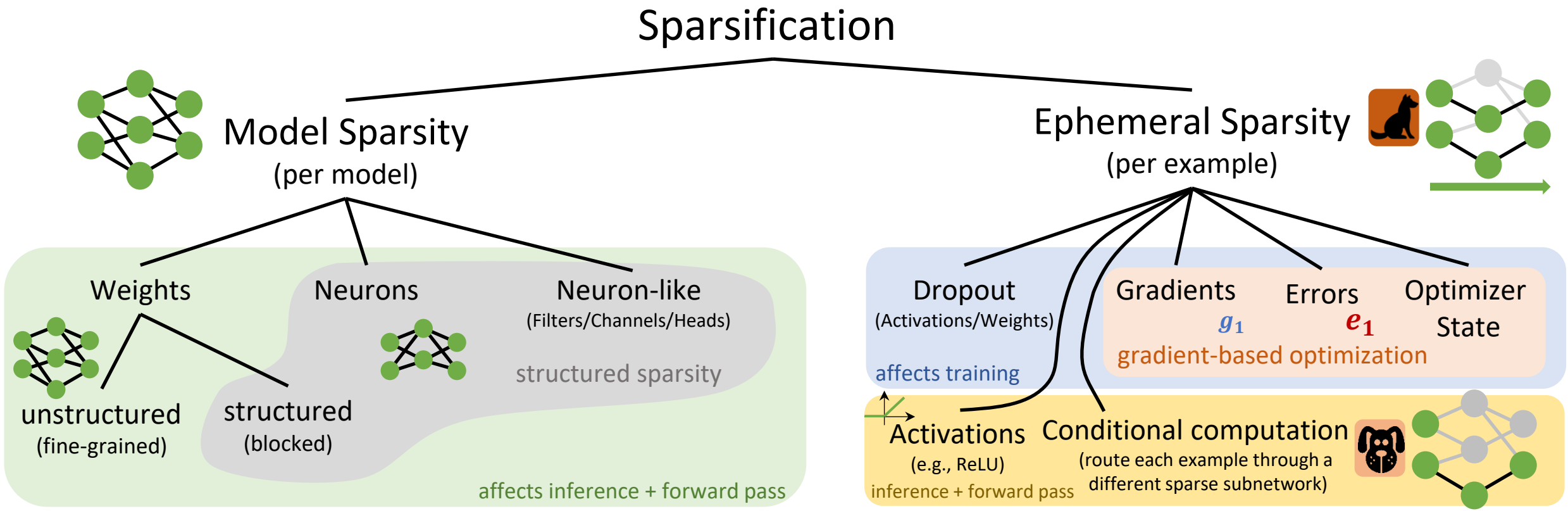
Back to data science – overview of approaches



Quite complex, isn't it? It'll get better ☺

references in Hoefler et al. "Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks", arXiv 2102.00554, Jan 2021

Back to data science – overview of approaches



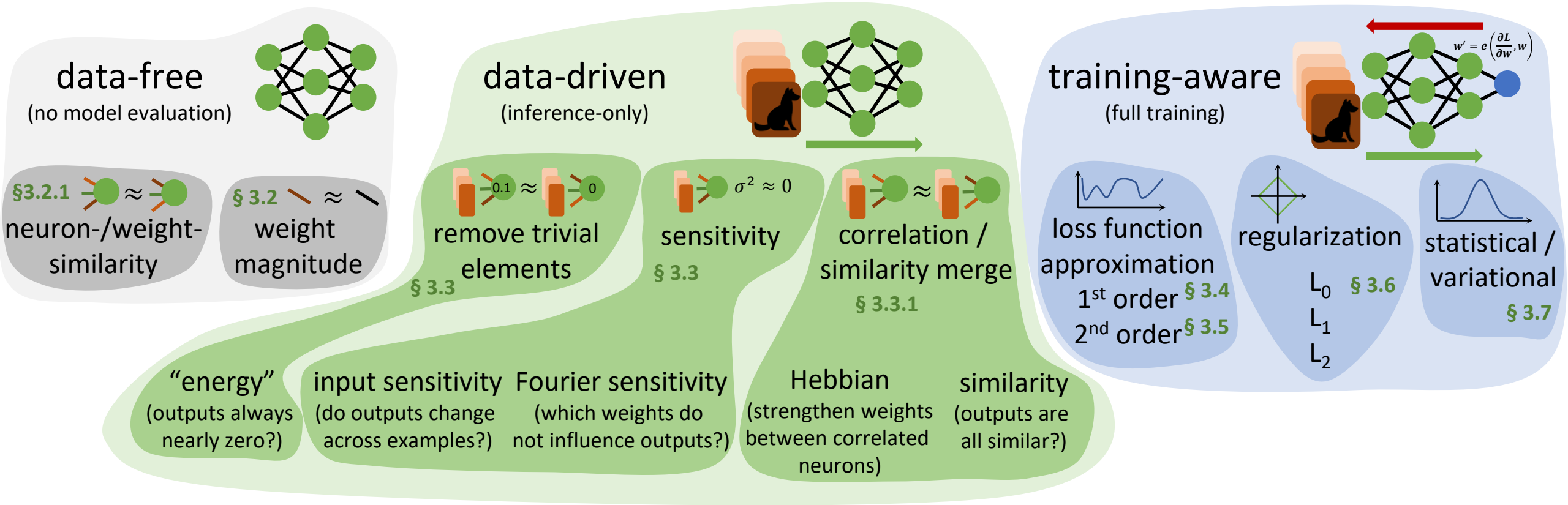
Quite complex, isn't it? It'll get better ☺

references in Hoefler et al. "Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks", arXiv 2102.00554, Jan 2021

What, when, now the how to sparsify / remove elements!

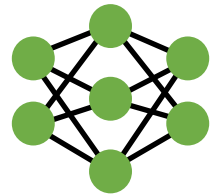
- **Simplest scheme: leave k out – train $\binom{n}{k}$ models to convergence ☹**
  - Various selection schemes by some importance metric

    Whirlwind overview of various metrics and selection techniques – then focus on some

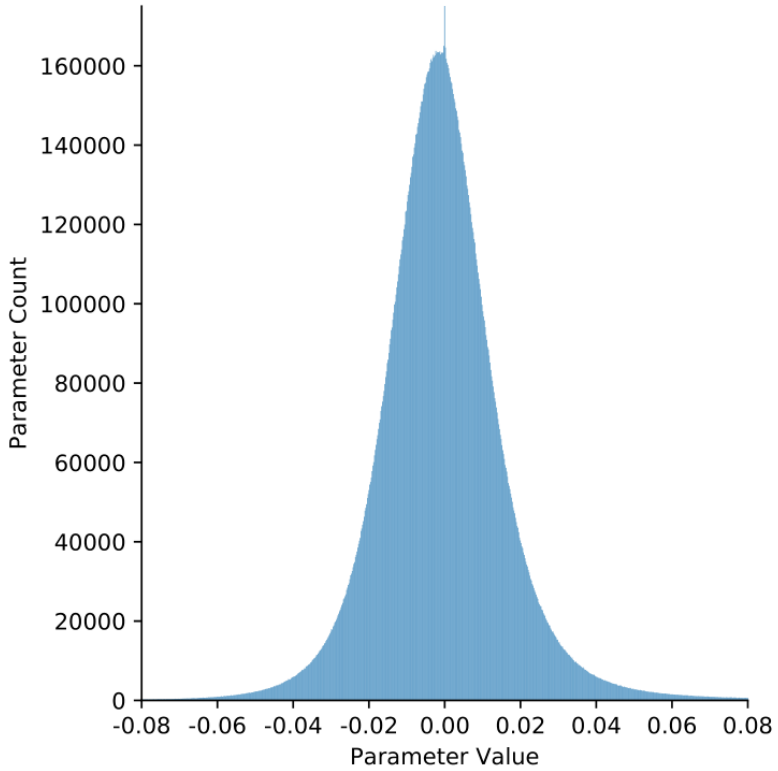


references in Hoefler et al. "Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks", arXiv 2102.00554, Jan 2021

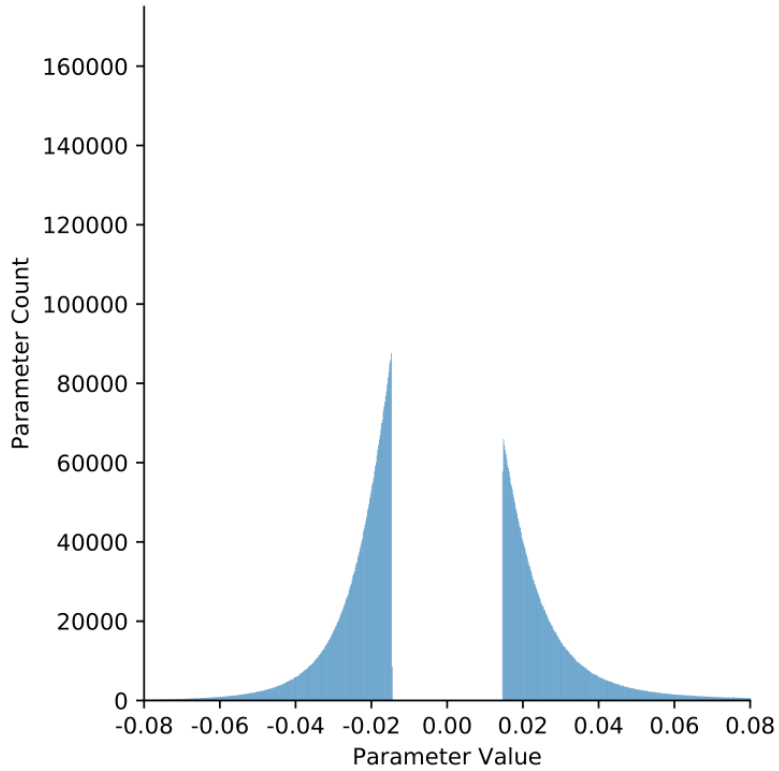
[Data free] Magnitude-based pruning



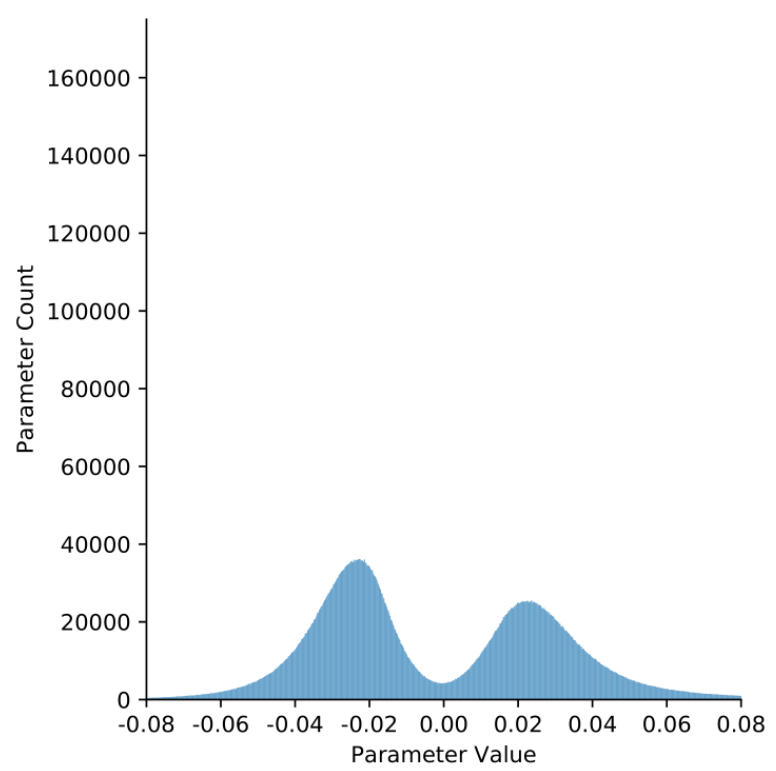
- **Remove weights with smallest absolute magnitude $|w|$**
  - Most popular and simplest - either by absolute value or select top-k – e.g., ResNet-50



(a) Dense Network (76.0%)



(b) 70% Pruned (36.1%)



(c) After 3-epoch Retraining (71.4%)

references in Hoefler et al. “Sparsity in Deep Learning: Pruning and growth for efficient inference and training in neural networks”, arXiv 2102.00554, Jan 2021