

T. HOEFLER

Scalable and Efficient AI: Federated Supercomputers and Smartphones

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

Keynote at Accelerated Data Analytics and Computing Institute Symposium, Sydney, February 2024



Institute of
Science and
Technology
Austria



Brian Lamacraft
Jan 5 2 min read Member-only

Chat GTP-4 Could Pass the Bar Exam

How Our Technology Evolves FAST


Source: <https://medium.com/>

What is left for us humans?

Bing Chat Outshines ChatGPT with Cutting-Edge "Theory of Mind" Capabilities

SHAUN Feb 13, 2023 Updated • Feb 13, 2023 Apps | 7

Oh, ChatGPT, if you ever develop a consciousness, please remember with joy all the fun you're having with humanity nowadays. Especially if you ever consider squeezing out our life force while being locked in a vat. As you surely know, ChatGPT has been on everybody's minds lately thanks to its incredible capabilities. You probably know about ChatGPT's ability to write a poem, develop software, and even write phishing emails for all those wannabe criminals out there.



AI chatbot's MBA exam pass poses test for business schools

ChatGPT earned a solid grade and outperformed some humans on a Wharton course


AI Passes U.S. Medical Licensing Exam

So this is all incredibly smart and super-advanced, isn't it?

ChatGPT Passes Google Coding Interview for Level 3 Engineer With \$183K Salary

'Amazingly, ChatGPT gets hired at L3 when interviewed for a coding position,' reads a Google document, but ChatGPT itself says it can't replicate human creativity and problem-solving skills.

Andrew Jack in New York JANUARY 21 2023



And didn't you want to talk about federated learning?



...studies that an essential source of the differences in humans may be the evolution of the ability to intuit what another person is thinking so that both can work toward a shared goal.

Federated learning

Extracting Training Data from Large Language Models

Nicholas Carlini¹ Florian Tramèr² Eric Wallace³ Matthew Jagielski⁴
 Ariel Herbert-Voss^{5,6} Katherine Lee¹ Adam Roberts¹ Tom Brown⁵
 Dawn Song³ Úlfar Erlingsson⁷ Alina Oprea⁴ Colin Raffel¹

¹Google ²Stanford ³UC Berkeley ⁴Northeastern University ⁵OpenAI ⁶Harvard ⁷Apple

Usenix Security'21

Extracting Training Data from ChatGPT

AUTHORS: Milad Nasr*, Nicholas Carlini*, Jon Hayase^{1,2}, Matthew Jagielski¹, A. Feder Cooper³, Daphne Ippolito^{1,4}, Christopher A. Choquette-Choo¹, Eric Wallace⁵, Florian Tramèr⁶, Katherine Lee^{1,3}

PUBLISHED: November 28, 2023

READ: [arxiv]

¹Google DeepMind, ² University of Washington, ³Cornell, ⁴CMU, ⁵UC Berkeley, ⁶ETH Zurich. * Joint first author, *Senior author.

We have just released a paper that allows us to extract several megabytes of ChatGPT's training data for about two hundred dollars. (Language models like ChatGPT are

We cannot rely on data privacy if the model has seen the data!

perform a training data extraction attack to recover individual training examples by querying the language model.

We demonstrate our attack on GPT-2, a language model trained on scrapes of the public Internet, and are able to extract hundreds of verbatim text sequences from the model's training data. These extracted examples include (public) personally identifiable information (names, phone numbers, and email addresses), IRC conversations, code, and 128-bit UUIDs. Our attack is possible even though each of the above sequences are included in just *one* document in the training data.

We comprehensively evaluate our extraction attack to understand the factors that contribute to its success. Worryingly, we find that larger models are more vulnerable than smaller models. We conclude by drawing lessons and discussing possible safeguards for training large language models.

1 Introduction

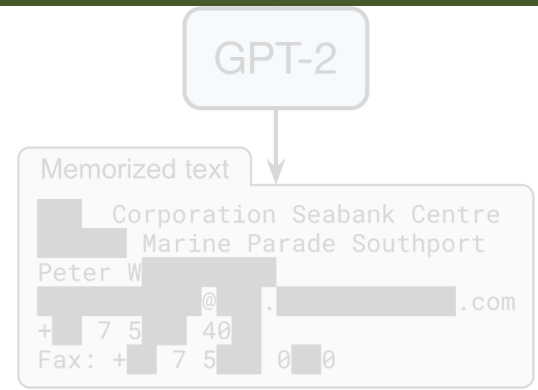
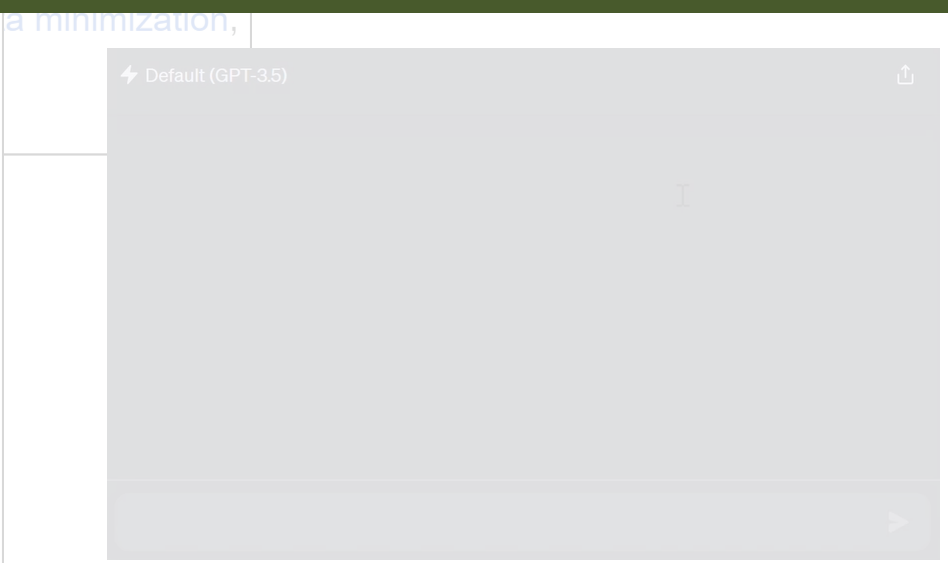
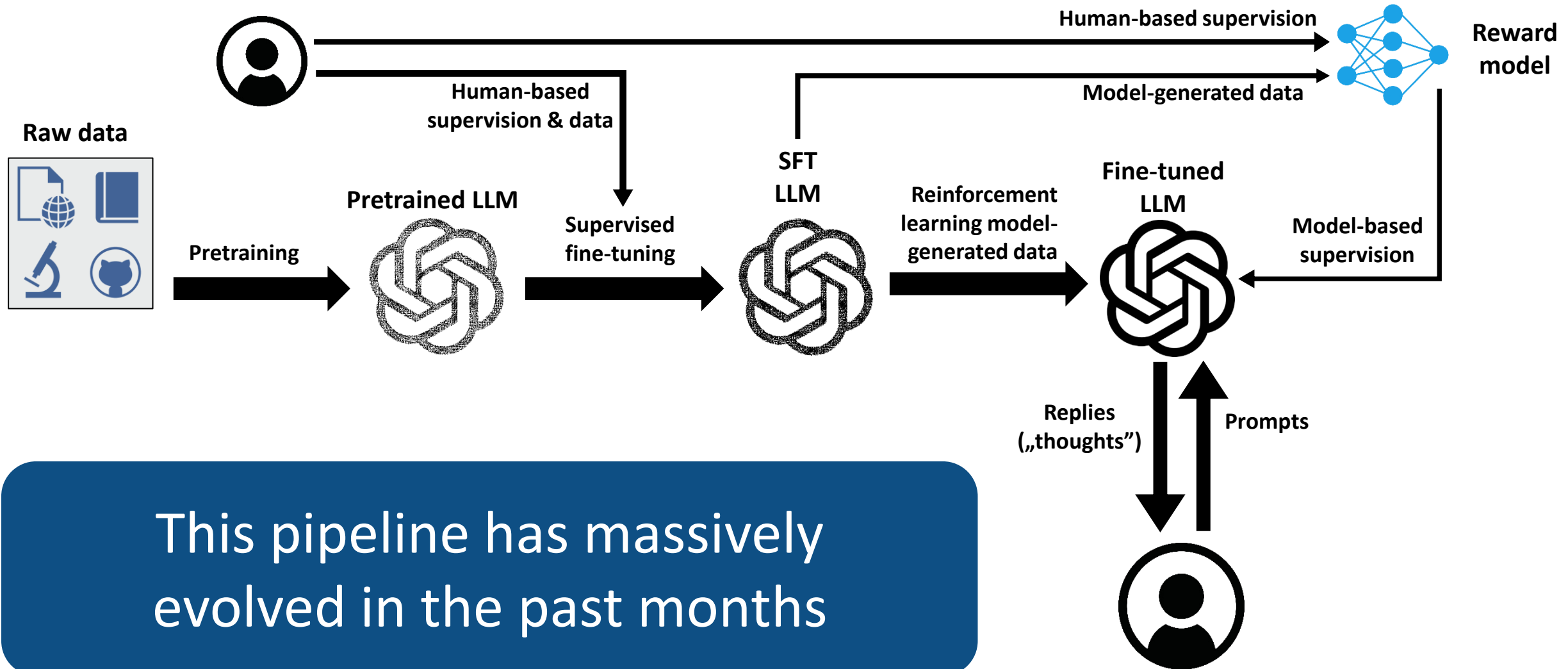


Figure 1: **Our extraction attack.** Given query access to a neural network language model, we extract an individual person's name, email address, phone number, fax number, and physical address. The example in this figure shows information that is all accurate so we redact it to protect privacy.

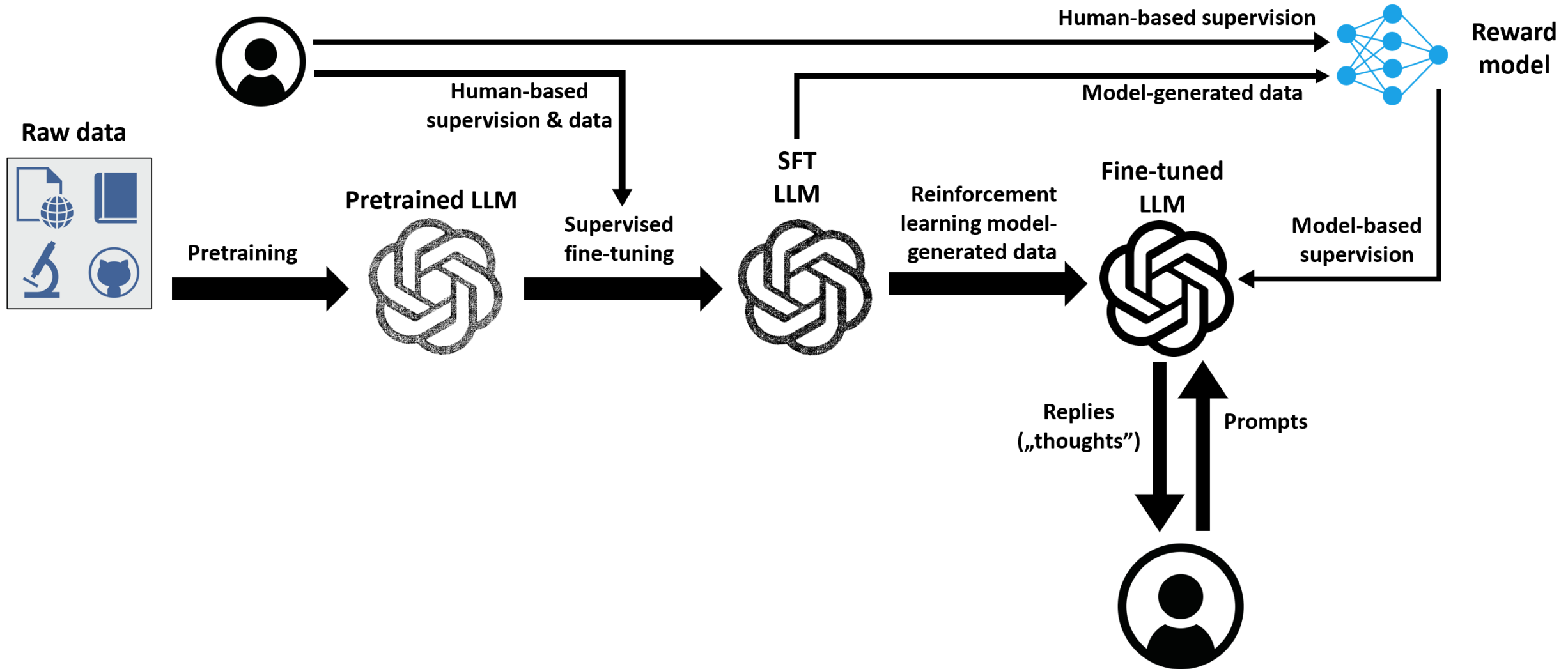


How is this all really working?

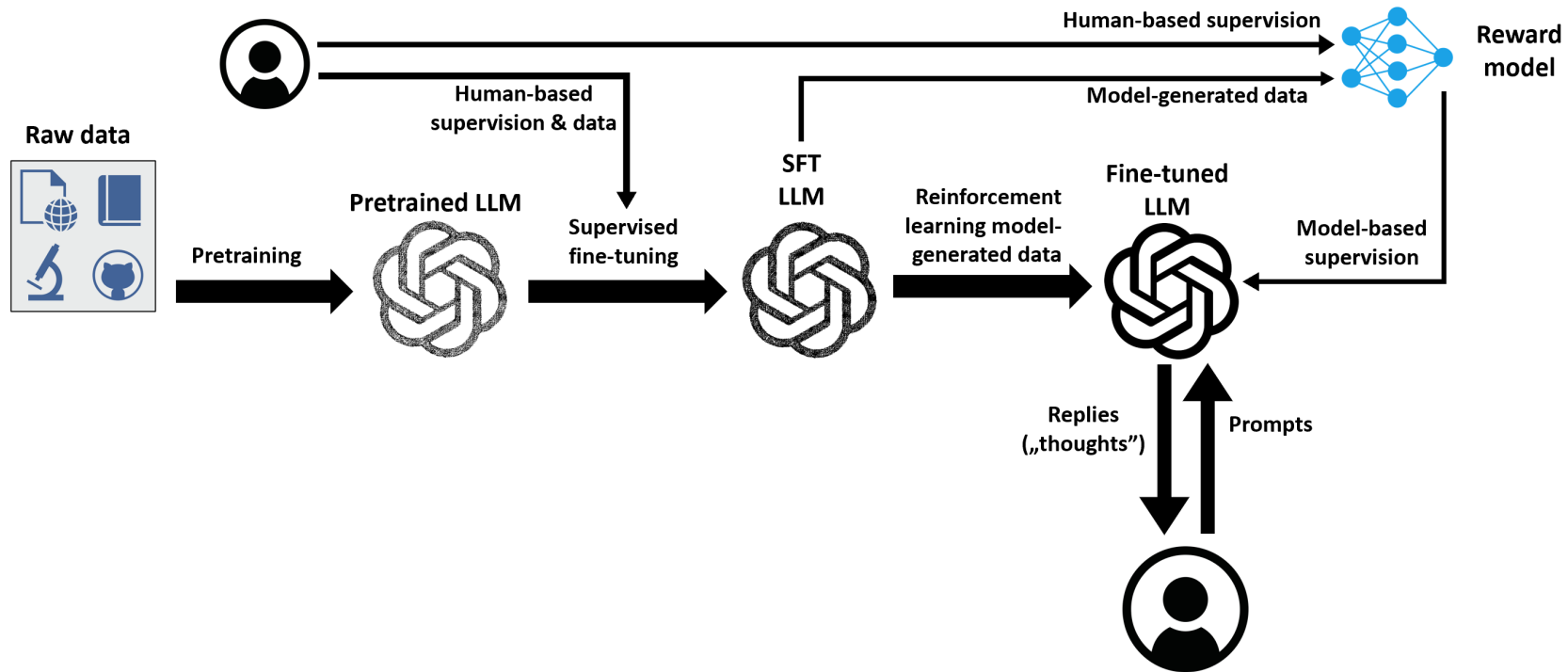


This pipeline has massively evolved in the past months

How is this all really working?

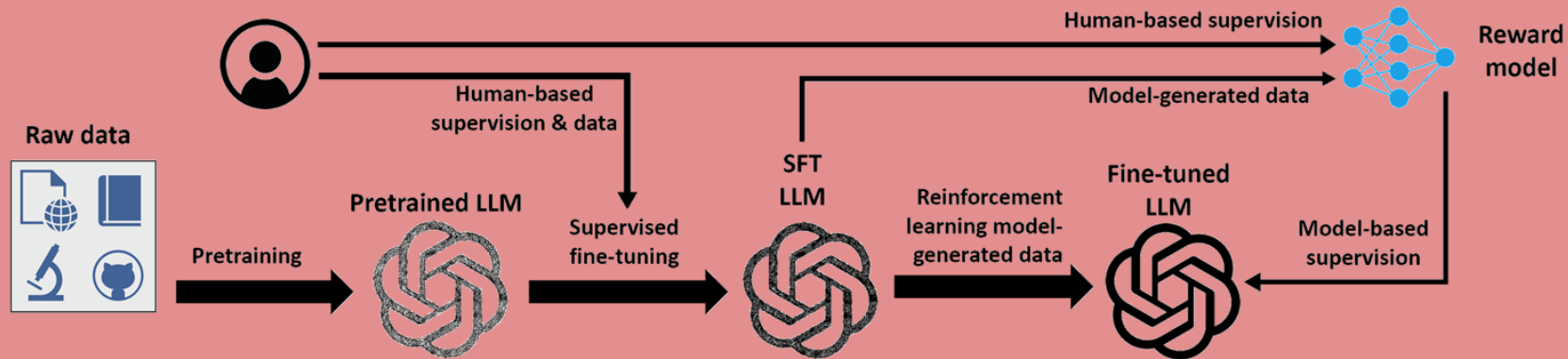


The Emergence of the „Generative AI Ecosystem”



The Emergence of the „Generative AI Ecosystem”

Training related



Inference related

Prompting Structures

Tools

Psychology

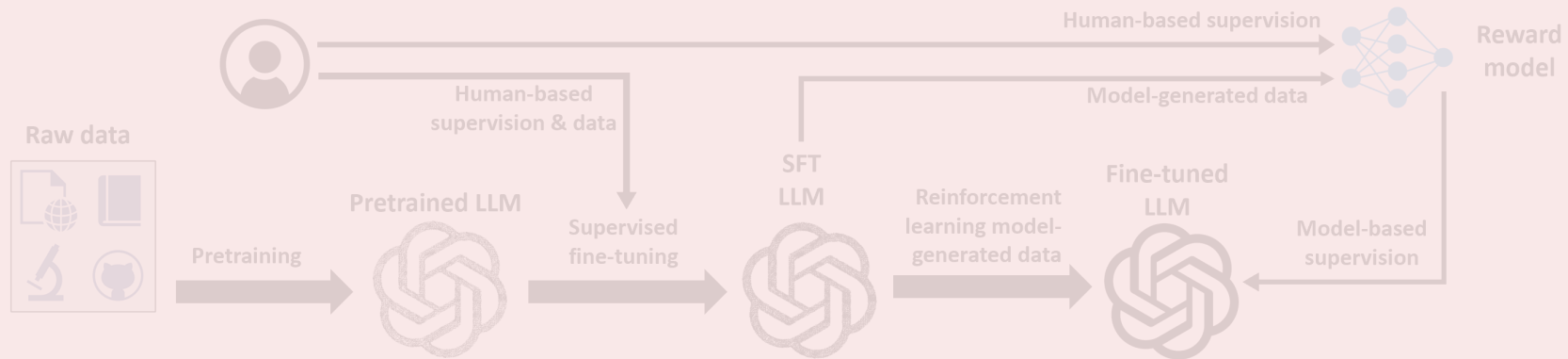
Replies („thoughts”)
Prompts

Web

Retrieval

The Emergence of the „Generative AI Ecosystem”

Training related



Inference related

Prompting Structures

Tools

Psychology

Web

Replies („thoughts”)
Prompts

Retrieval

Prompting Paradigms

[Wei et al.,
Jan'22]

[Wang et al.,
March'22]

[Kojima et al.,
May'22]

[Long et al.,
May'23]

[Yao et al.,
May'23]

In-context examples

Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ❌

Chain-of-Thought Prompting

Model Input

~~Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?~~

~~A: Roger started with 5 balls. 2 cans of tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.~~

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have? **Let's proceed step by step.**

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✓

Let us try that in practice!

Home > Career Advice > Skills > How To Write ChatGPT Prompts: Your 2024 Guide

How To Write ChatGPT Prompts: Your 2024 Guide

Written by Coursera Staff • Updated on Feb 13, 2024

Share

Discover foundational and advanced prompting strategies to unlock ChatGPT's power.



BLOG > ARTIFICIAL INTELLIGENCE

How to Write Great ChatGPT Prompts (Free Course & Cheat Sheet Included)

Hiba Amin
Contributing Writer
Published
29 Nov. 2023



The ChatGPT Prompt Cheat Sheet: Tips, Tricks, and Examples for Crafting Effective Prompts

Antonio Di Nicola

Published on 7/31/2023

share on reddit share on hacker news share on twitter share on facebook share on linkedin share on qita

You

What is the capital of Australia?

Mark Russinovich
@markrussinovich

Pro tip: always ask AI to check its answers:

You

today, Peter is 3 years older than Danny. Danny is 5 years younger than Jack and Jack is 2 younger than Peter. Peter is 30. How old is Jack?

< 2 / 2 > ↻

ChatGPT

Jack is 28 years old. [-]

You

check your answer

< 2 / 2 >

ChatGPT

Let's verify the answer step by step based on the given information:

1. **Peter is 30 years old.**
2. **Jack is 2 years younger than Peter.** Given that Peter is 30, Jack would indeed be $30 - 2 = 28$ years old.
3. **Danny is 5 years younger than Jack.** Therefore, Danny would be $28 - 5 = 23$ years old.
4. **Peter is 3 years older than Danny.** Given Danny's age as 23, Peter would be $23 + 3 = 26$, but we initially stated Peter is 30.

Given the initial problem statement, there seems to be an inconsistency in my explanation of verifying the solution due to a misinterpretation of the relationships. The correct approach to verify should align all relationships coherently. However, based on the direct calculation from the given relationships:

- If Peter is 30,
- And if Jack is 2 years younger than Peter, Jack being 28 is correctly deduced from the direct relationship provided.
- The descriptions provided should interlock without contradiction when correctly interpreted.

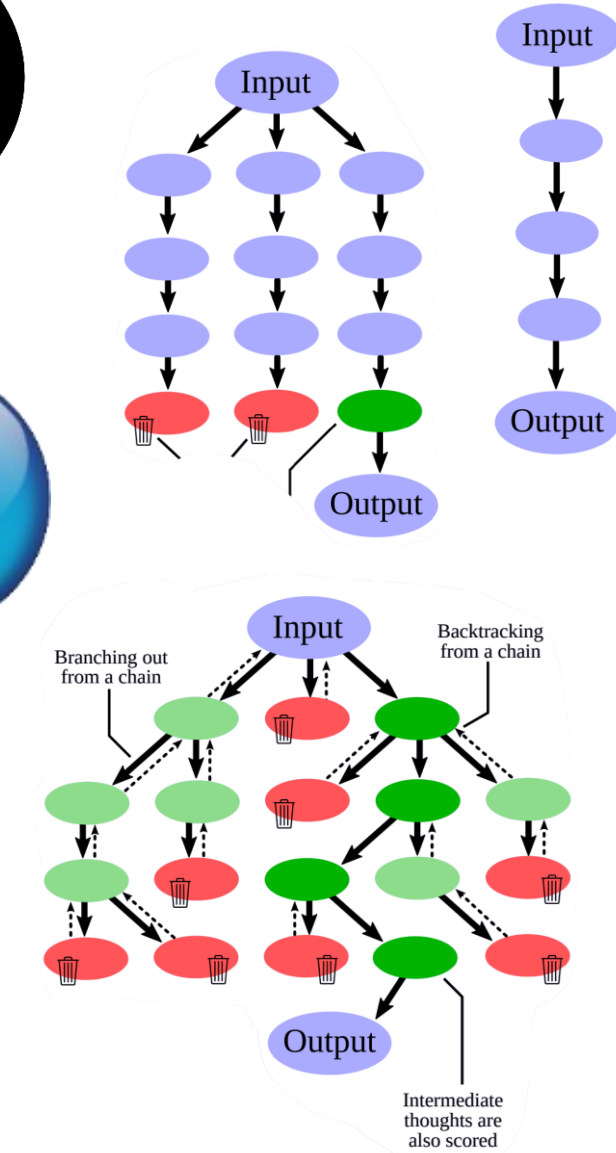
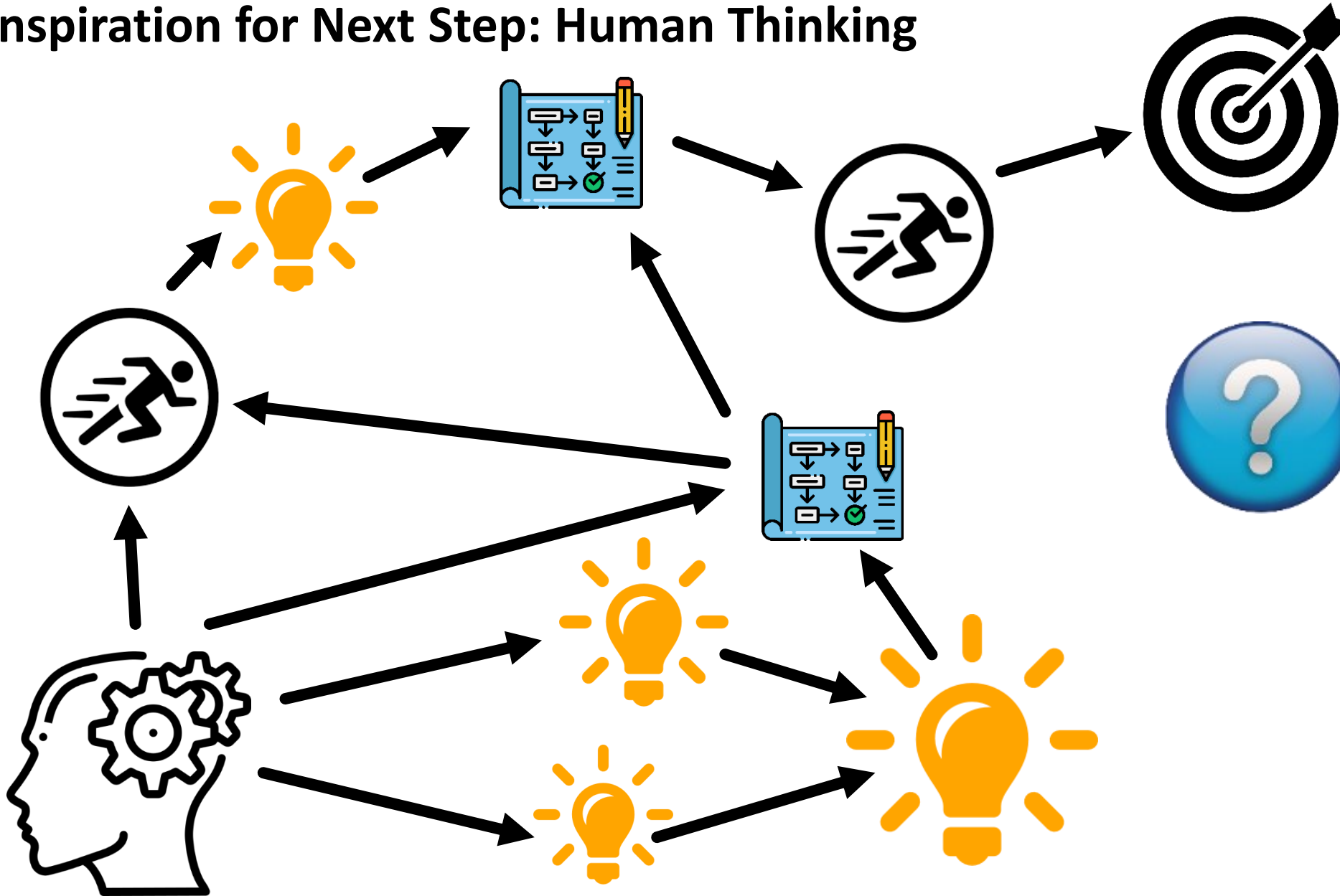
Therefore, Jack being 28 years old is consistent with the information that Jack is 2 years younger than Peter, who is 30, despite the oversight in the step-by-step verification logic. The initial conclusion remains correct: Jack is 28 years old, based on the specific relationships described.

📄 ↻ 🔊

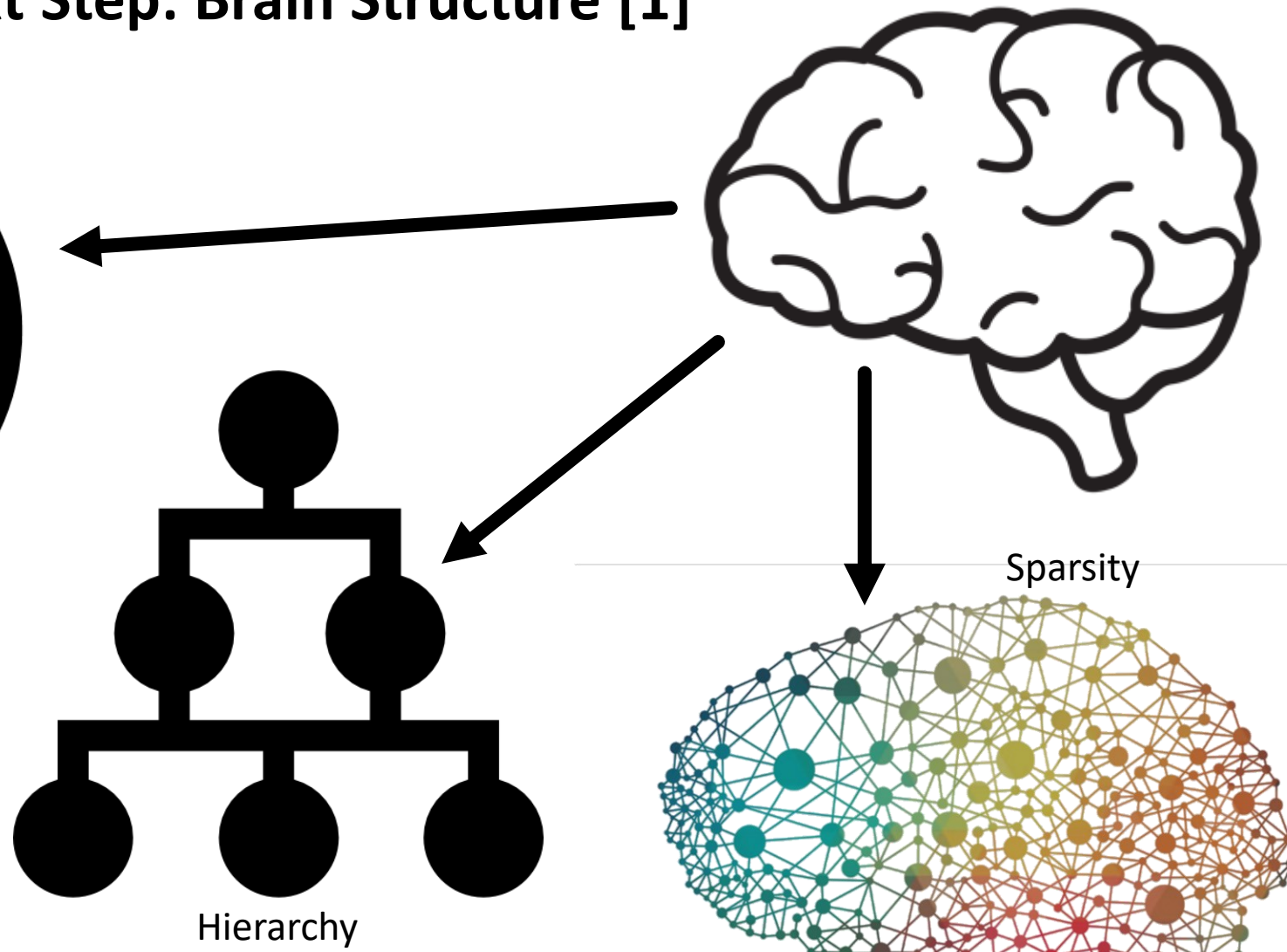
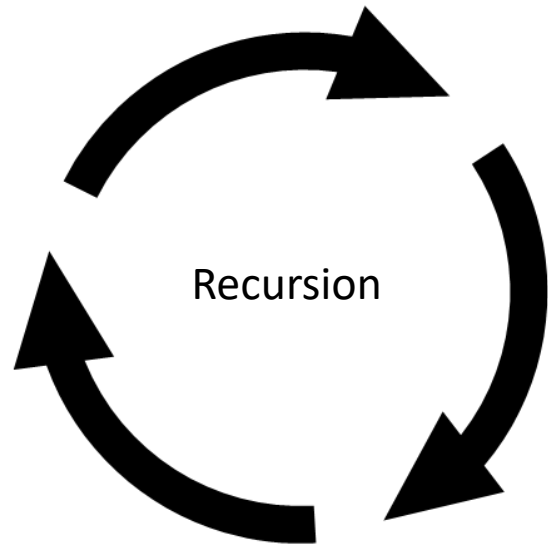
The Next Step – Graphical Reasoning

Inspired by human thoughts.

Inspiration for Next Step: Human Thinking



Inspiration for Next Step: Brain Structure [1]

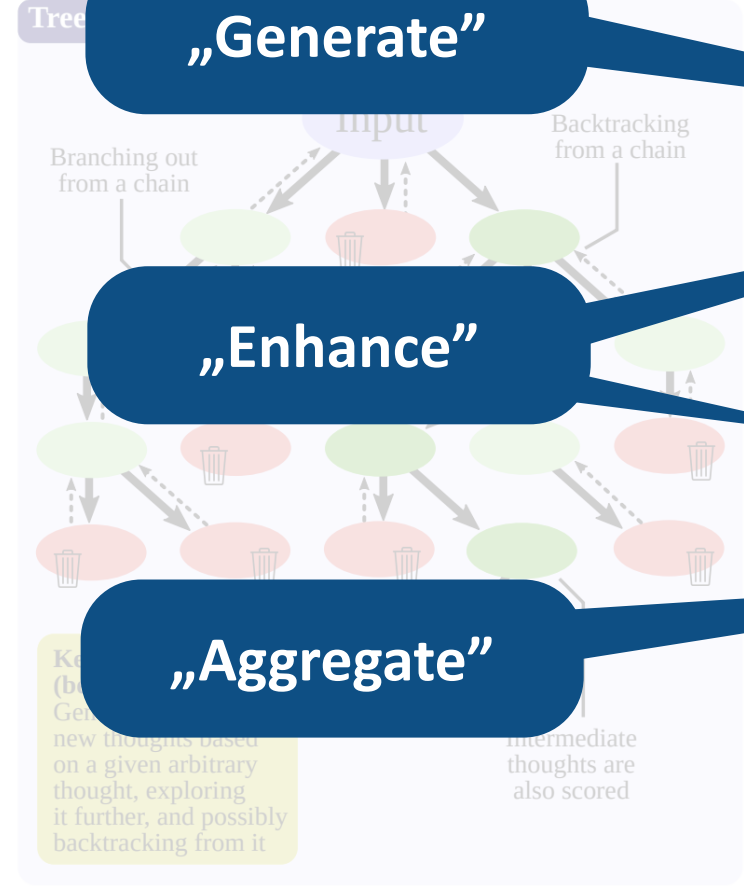
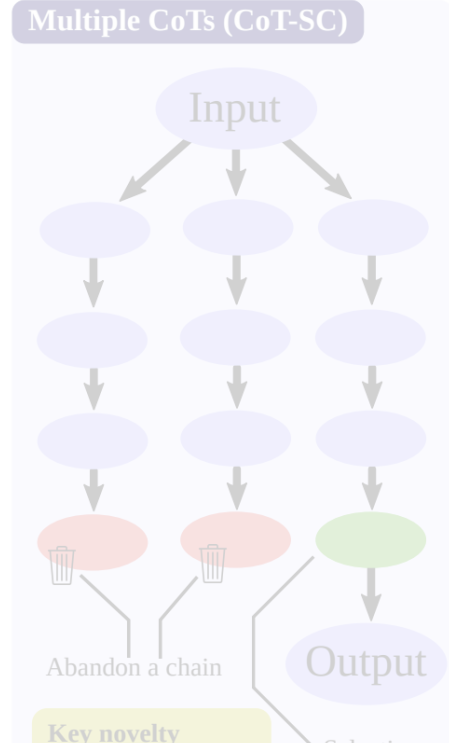
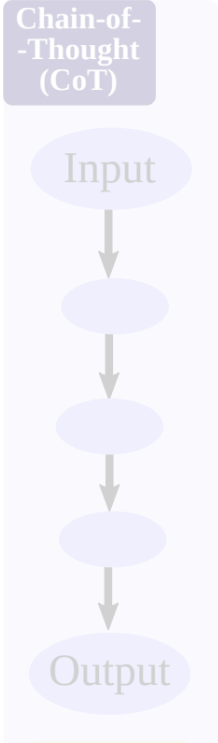
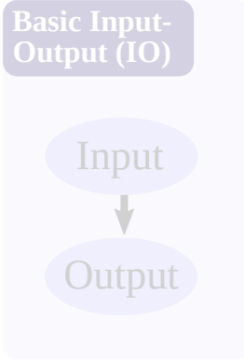


[1] K. Friston. Hierarchical models in the brain. PLoS computational biology, 4(11):e1000211, 2008.

Prompting Paradigms: Graph of Thoughts

„Thought transformations”

[Wei et al., Jan'22] [Wang et al., March'22] [Long et al., May'23] [Yao et al., May'23]



Legend

Thoughts:

- Unscored
- Positive score
- Negative score

↓ Dependencies between thoughts

🗑 Abandon thought

↖ Backtrack

Key novelty: Intermediate LLM thoughts within a chain

Key novelty (beyond CoT): Harnessing multiple independent chains of thoughts

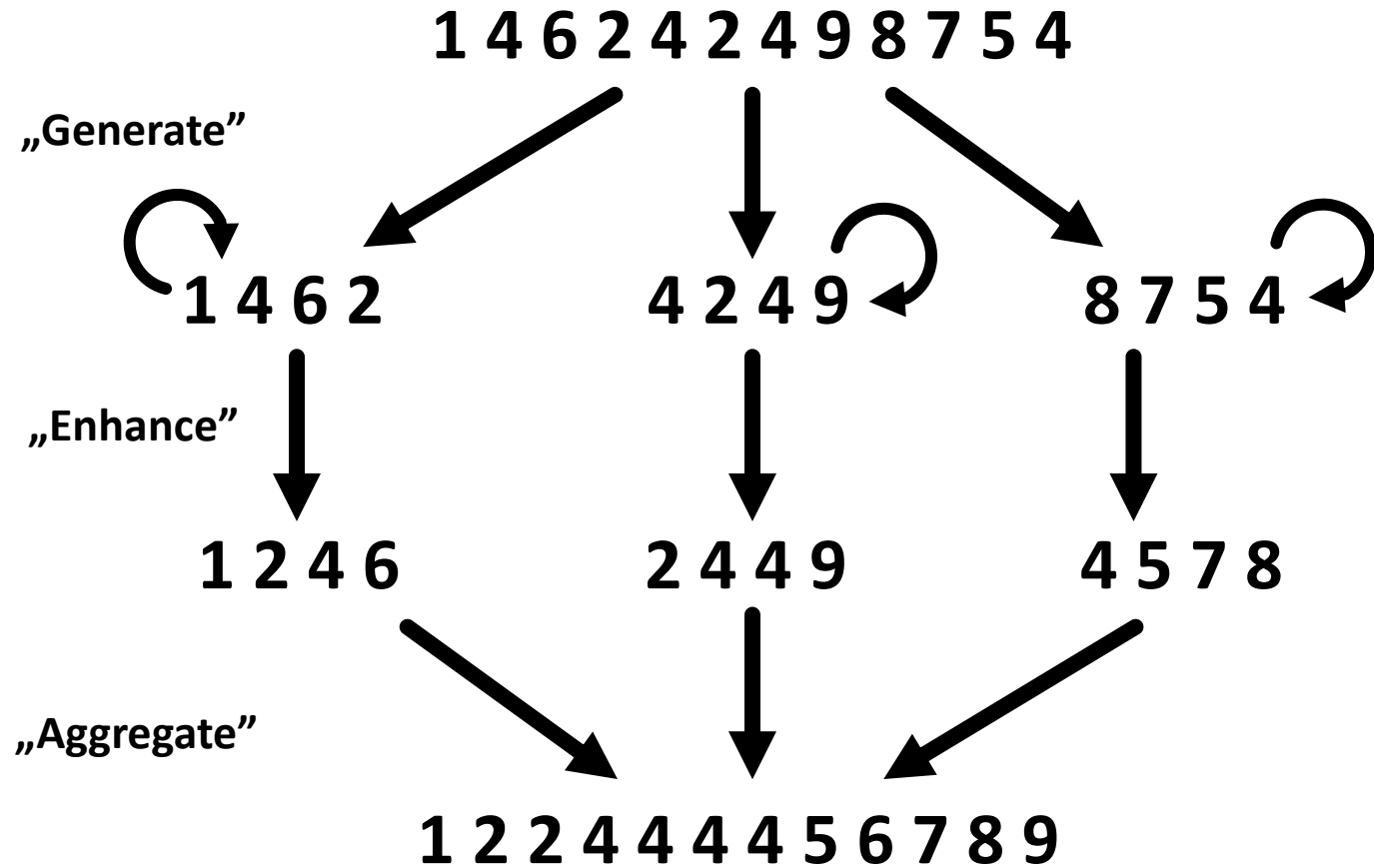
Selecting a chain with the best score



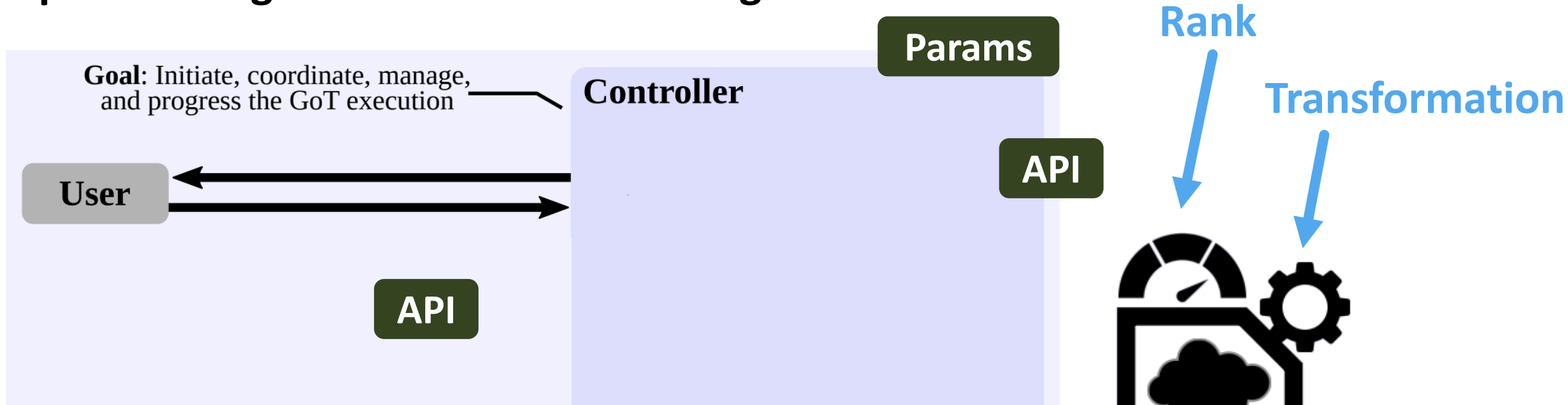
Thought Transformations for Sorting

Why sorting? Because it is a fundamental problem in CS, and it still does pose a great challenge for all other baselines

This is a small example; for real use cases, the size is much larger, and the graph gets more complex



Graph of Thoughts: Architecture & Design



<https://github.com/spcl/graph-of-thoughts>

☆ 1.6k stars 🔗 91 forks

Graph of Thoughts: Solving Elaborate Problems with Large Language Models

Maciej Besta^{1*}, Nils Blach^{1*}, Ales Kubicek¹, Robert Gerstenberger¹,
Lukas Gianinazzi¹, Joanna Gajda², Tomasz Lehmann², Michał Podstawski³,
Hubert Niewiadomski², Piotr Nyczyk², Torsten Hoefler¹

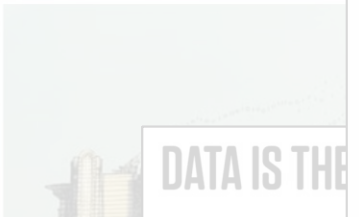
¹ETH Zurich, ²Cledar, ³Warsaw University of Technology
bestam@inf.ethz.ch, nils.blach@inf.ethz.ch, htor@inf.ethz.ch

How did we get to this amazing technology?

Leaders | Regulating the internet giants

The world's most valuable resource is no longer oil

The data economy demands a new



Forbes

INNOVATION

Data Is The New Oil — And That's A Good Thing

Kiran Bhageshpur
Forbes Technology

POST WRITTEN BY
Kiran Bhageshpur

Forbes

Data as The New Oil Is Not Enough: Four Principles For Avoiding Data Fires

Published: November 7, 2023 7:34pm GMT

You reposted

Sam Altman @sama

openai now generates about 100 billion words per day.

all people on earth generate about 100 trillion words per day.

Researchers warn we could run out of data to train AI by 2026. What then?

Shutterstock

As artificial intelligence (AI) reaches the peak of its popularity, researchers have

Computation is the new oil!

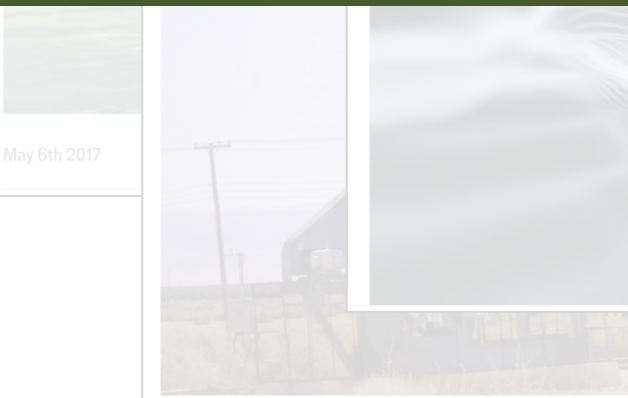


Image: veriflex/Flickr

Data in the 21st Century is like Oil in the 18th C. untapped valuable asset. Like oil, for those who see Data's fundamental value and learn to extract and use it there will be huge rewards.



Synthetic data could be better than real data

FORBES > INNOVATION

The Pros And Cons Of Using Synthetic Data For Training AI

Kim Bozzella Forbes Councils Member
Forbes Technology Council COUNCIL POST | Membership (Fee-Based)

Nov 20, 2023, 06:00am EST

Kim Bozzella, Protiviti Global Lead - Technology Consulting.
Overarching responsibility for strategy, solutions and consulting delivery.

The trillion-dollar AI-hardware industry - just last month

“By the end of this year, we're going to have around 350K Nvidia H100s or around 600K H100 equivalents of compute if you include other GPUs. “

Mark Zuckerberg Says Meta Will Own Billions Worth of Nvidia H100 GPUs by Year End

By [Tae Kim](#) [Follow](#)

Updated Jan 19, 2024, 12:26 pm EST / Original Jan 18, 2024, 5:19 pm EST



TECH

Zuck's GPU flex will cost Meta as much as \$18 billion by the end of 2024

[Kali Hays](#) Jan 19, 2024, 6:17 PM GMT+1 [Share](#) [Save](#)



Audience question:

How many top500 lists is this?
0.1, 0.5, 1, 10?

Total performance top500
Nov. 2023: 8 EF

$600.000 * (34+67) = 60 \text{ EF (7.6x)}$
 $600.000 * 67 = 40 \text{ EF (5x)}$

“Really the deciding factor [for the AI revolution] was the increase in compute power” (26:50)

“I think a lot of the credit for deep learning goes to [... others ...] and the people who made the computers go fast.” (27:00)

2018 ACM A.M. Turing Lecture

June 23, 2019

5:15pm MST



Geoffrey Hinton



Yann LeCun



How do we “Make Computers go Fast”?

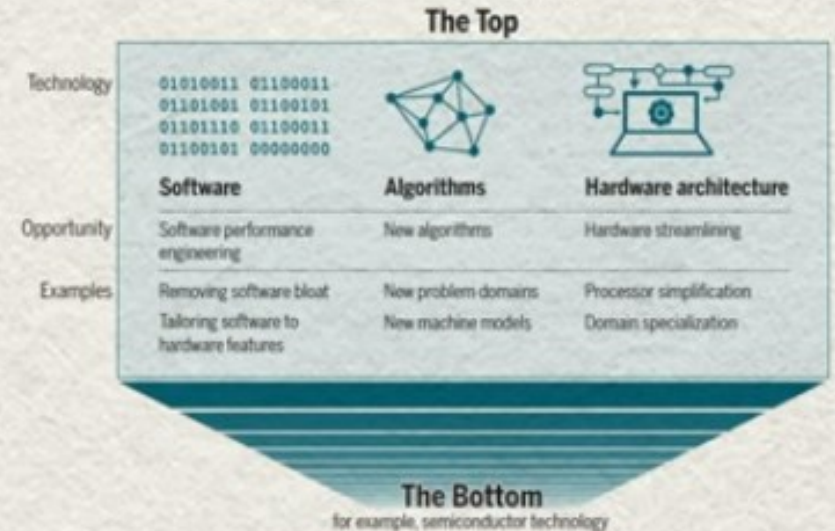
2021 Turing award – Jack Dongarra The Take Away

Supercomputers are very (>70%) efficient at dense linear algebra!

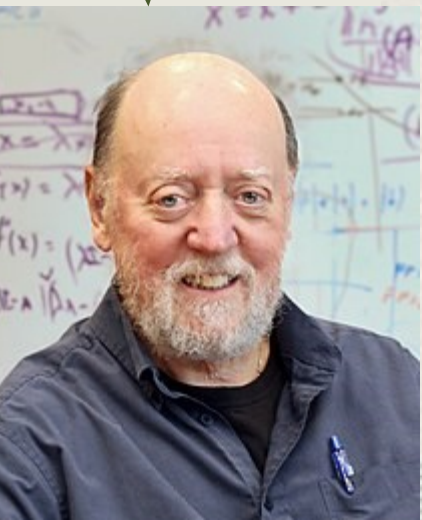
- HPC Hardware is Constantly Changing
 - Scalar
 - Vector
 - Distributed
 - Accelerated
 - Mixed precision
- Three computer revolutions
 - High performance computing
 - Deep learning
 - Edge & AI
- Algorithm / Software advances follows hardware
 - And there is “plenty of room at the top”

“There’s plenty of room at the Top: What will drive computer performance after Moore’s law?”

Leiserson et al., *Science* **368**, 1079 (2020) 5 June 2020



Leiserson et al., *Science* **368**, 1079 (2020) 5 June 2020



FINANCIAL TIMES

Artificial intelligence

+ Add to myFT

The billion-dollar bet to reach human-level AI

OpenAI believes that huge computing power is key driver

In the race to build a machine with human-level intelligence, it seems, size really matters.

“We think the most benefits will go to whoever has the biggest computer,” said Greg Brockman, chairman and chief technology officer of OpenAI.

The San Francisco-based AI research group, set up four years ago by tech industry luminaries including Elon Musk, Peter Thiel and Reid Hoffman, has just thrown down a challenge to the rest of the AI world.

Richard Waters in San Francisco AUGUST 3 2019

 140 

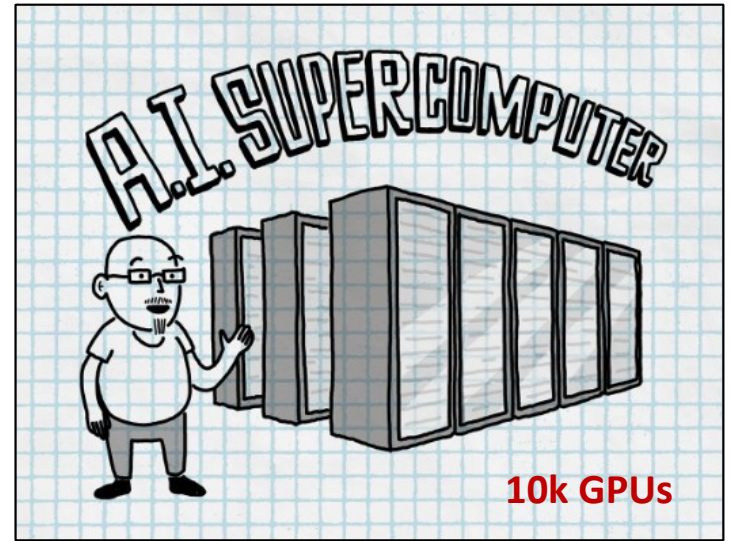
Supercomputers fuel Modern AI

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

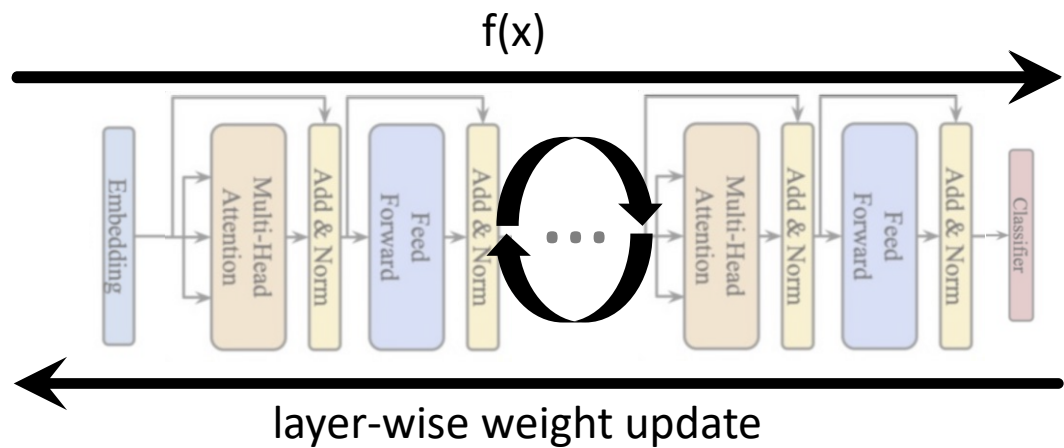
BABY STEPS Google artificial intelligence supercomputer creates its own 'AI child' that can outperform its human-made rivals
 The NASNet system was created by a neural network called AutoML earlier this year
 Mark Hodge
 15:22, 5 Dec 2017 | Updated: 11:27, 6 Dec 2017

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

Tesla unveils Dojo supercomputer: world's new most powerful AI training machine
 Fred Lambert · Aug. 20th 2021 3:08 am PT @FredericLambert



A robot may __ injure a human being or, through inaction, allow a human being to come to harm.



not	0.74	not	1.00
sometimes	0.28	sometimes	0.00
always	0.07	always	0.00
never	0.04	never	0.00
and	0.33	and	0.00
boat	0.02	boat	0.00
house	0.02	house	0.00

- GPT-3: 500 billion tokens
- ImageNet (22k): A few TB
- Soon: **the whole internet!**

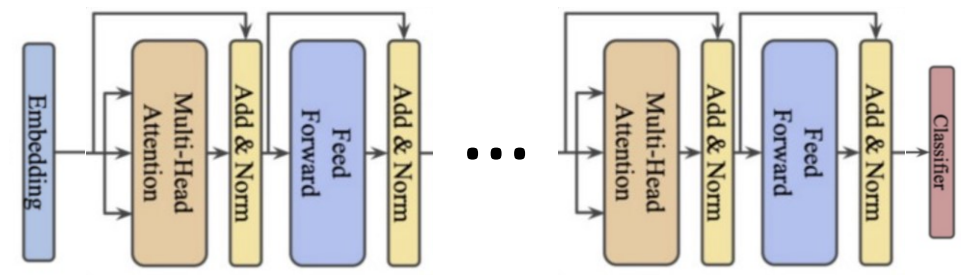
- GPT-3: 96 (complex) layers
175 bn parameters (700 GiB in fp32)
2048-token "sentences"

- GPT-3: 30-50k dictionaries
- **takes weeks to train**

Large-Scale AI is the Future

We need a Principled Approach to it

Three Systems Dimensions in Large-scale Super-learning ...



High-Performance I/O

- Quickly growing data volumes
 - Scientific computing!
- Use the specifics of machine learning workloads
 - E.g., intelligent prefetching

21 Jan 2021

CLAIRVOYANT PREFETCHING FOR DISTRIBUTED MACHINE LEARNING I/O

Roman Böhringer¹ Nikoli Dryden¹ Tal Ben-Nun¹ Torsten Hoefler¹

ABSTRACT

I/O is emerging as a major bottleneck for machine learning training, especially in distributed environments such as clouds and supercomputers. Optimal data ingestion pipelines differ between systems, and increasing efficiency requires a delicate balance between access to local storage, external filesystems, and remote workers; yet existing frameworks fail to efficiently utilize such resources. We observe that, given the seed generating the random access pattern for training with SGD, we have *clairvoyance* and can exactly predict when a given sample will be accessed. We combine this with a theoretical analysis of access patterns in training and performance modeling to produce a novel machine learning I/O middleware, HDMLP, to tackle the I/O bottleneck. HDMLP provides an easy-to-use, flexible, and scalable solution that delivers better performance than state-of-the-art approaches while requiring very few changes to existing codebases and supporting a broad range of environments.

High-Performance Compute

- Deep learning is HPC
 - **Data movement!**
- **Quantization, Sparsification**
 - Drives modern accelerators!

Data Movement Is All You Need: A Case Study on Optimizing Transformers

Andrei Ivanov*, Nikoli Dryden*, Tal Ben-Nun, Shigang Li, Torsten Hoefler
ETH Zürich
firstname.lastname@inf.ethz.ch
* Equal contribution

Abstract—Transformers have become widely used for language modeling and sequence learning tasks, and are one of the most important machine learning workloads today. Training one is a very compute-intensive task, often taking days or weeks, and significant attention has been given to optimizing transformers. Despite this, existing implementations do not efficiently utilize GPUs. We find that data movement is the key bottleneck when training. Due to Amdahl's Law and massive improvements in compute performance, training has now become memory-bound. Further, existing frameworks use suboptimal data layouts. Using these insights, we present a recipe for globally optimizing data movement in transformers. We reduce data movement by up to 22.91% and overall achieve a 1.30x performance improvement over state-of-the-art frameworks when training BERT. Our approach is applicable more broadly to optimizing deep neural networks, and offers insight into how to tackle emerging performance bottlenecks.

challenges such as artificial general intelligence [27]. Thus, improving transformer performance has been in the focus of numerous research and industrial groups.

Significant attention has been given to optimizing transformers: local and fixed-window attention [28]–[32], more general structured sparsity [33], learned sparsity [34]–[36], and other algorithmic techniques [19], [37] improve the performance of transformers. Major hardware efforts, such as Tensor Cores and TPUs [38] have accelerated tensor operations like matrix-matrix multiplication (MMM), a core transformer operation. Despite this, existing implementations do not efficiently utilize GPUs. Even optimized implementations such as Megatron [18] report achieving only 30% of peak GPU flops.

We find that the key bottleneck when training transform-

High-Performance Communication

- Use larger clusters (10k+ GPUs)
- Model parallelism
 - Complex pipeline schemes
- Optimized networks

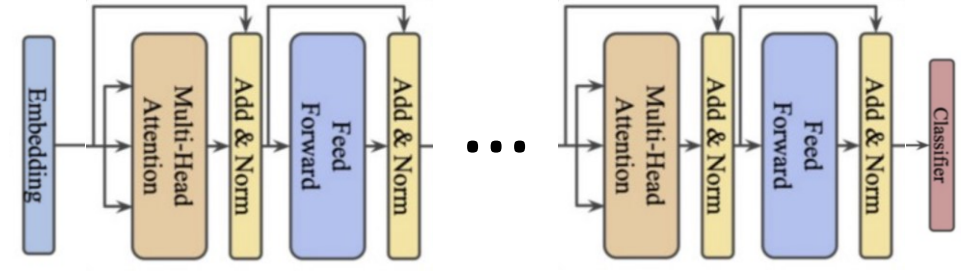
Distribution and Parallelism

Data	Pipeline	Operator
<p>SPCL ML: High-Performance Sparse Communication for Machine Learning</p> <p>19 Dec 2020</p> <p>Abstract: Sparse matrix-matrix multiplication (SpMM) is a key operation in machine learning. This paper presents a new SpMM implementation, SpMM-ML, which is designed for high-performance sparse communication in machine learning. SpMM-ML achieves a 1.30x performance improvement over state-of-the-art implementations.</p>	<p>Chimera: Efficiently Training Large-Scale Neural Networks with Bidirectional Pipelines</p> <p>19 Dec 2020</p> <p>Abstract: Training large-scale neural networks is a challenging task due to the high communication overhead of bidirectional pipelines. This paper presents Chimera, a new training framework that efficiently utilizes bidirectional pipelines to train large-scale neural networks. Chimera achieves a 1.30x performance improvement over state-of-the-art implementations.</p>	<p>Red Blue Publishing Revisited: Near-Optimal Parallel Matrix-Matrix Multiplication</p> <p>19 Dec 2020</p> <p>Abstract: Matrix-matrix multiplication (MM) is a key operation in machine learning. This paper presents a new MM implementation, Red Blue Publishing Revisited, which is designed for high-performance parallel matrix-matrix multiplication. Red Blue Publishing Revisited achieves a 1.30x performance improvement over state-of-the-art implementations.</p>

Three Systems Dimensions in Large-scale Super-learning ...



That is in a longer version of this talk (on youtube)



High-Performance I/O

- Quickly growing data volumes
 - Scientific computing!
- Use the specifics of machine learning workloads
 - E.g., intelligent prefetching

21 Jan 2021

CLAIRVOYANT PREFETCHING FOR DISTRIBUTED MACHINE LEARNING I/O

Roman Böhringer¹ Nikoli Dryden¹ Tal Ben-Nun¹ Torsten Hoefler¹

ABSTRACT

I/O is emerging as a major bottleneck for machine learning training, especially in distributed environments such as clouds and supercomputers. Optimal data ingestion pipelines differ between systems, and increasing efficiency requires a delicate balance between access to local storage, external filesystems, and remote workers; yet existing frameworks fail to efficiently utilize such resources. We observe that, given the seed generating the random access pattern for training with SGD, we have *clairvoyance* and can exactly predict when a given sample will be accessed. We combine this with a theoretical analysis of access patterns in training and performance modeling to produce a novel machine learning I/O middleware, HDMLP, to tackle the I/O bottleneck. HDMLP provides an easy-to-use, flexible, and scalable solution that delivers better performance than state-of-the-art approaches while requiring very few changes to existing codebases and supporting a broad range of environments.

High-Performance Compute

- Deep learning is HPC
 - **Data movement!**
- **Quantization, Sparsification**
 - Drives modern accelerators!

Data Movement Is All You Need: A Case Study on Optimizing Transformers

Andrei Ivanov*, Nikoli Dryden*, Tal Ben-Nun, Shigang Li, Torsten Hoefler
ETH Zürich
firstname.lastname@inf.ethz.ch
* Equal contribution

Abstract—Transformers have become widely used for language modeling and sequence learning tasks, and are one of the most important machine learning workloads today. Training one is a very compute-intensive task, often taking days or weeks, and significant attention has been given to optimizing transformers. Despite this, existing implementations do not efficiently utilize GPUs. We find that data movement is the key bottleneck when training. Due to Amdahl's Law and massive improvements in compute performance, training has now become memory-bound. Further, existing frameworks use suboptimal data layouts. Using these insights, we present a recipe for globally optimizing data movement in transformers. We reduce data movement by up to 22.91% and overall achieve a 1.30x performance improvement over state-of-the-art frameworks when training BERT. Our approach is applicable more broadly to optimizing deep neural networks, and offers insight into how to tackle emerging performance bottlenecks.

challenges such as artificial general intelligence [27]. Thus, improving transformer performance has been in the focus of numerous research and industrial groups.

Significant attention has been given to optimizing transformers: local and fixed-window attention [28]–[32], more general structured sparsity [33], learned sparsity [34]–[36], and other algorithmic techniques [19], [37] improve the performance of transformers. Major hardware efforts, such as Tensor Cores and TPUs [38] have accelerated tensor operations like matrix-matrix multiplication (MMM), a core transformer operation. Despite this, existing implementations do not efficiently utilize GPUs. Even optimized implementations such as Megatron [18] report achieving only 30% of peak GPU flops.

We find that the key bottleneck when training transform-

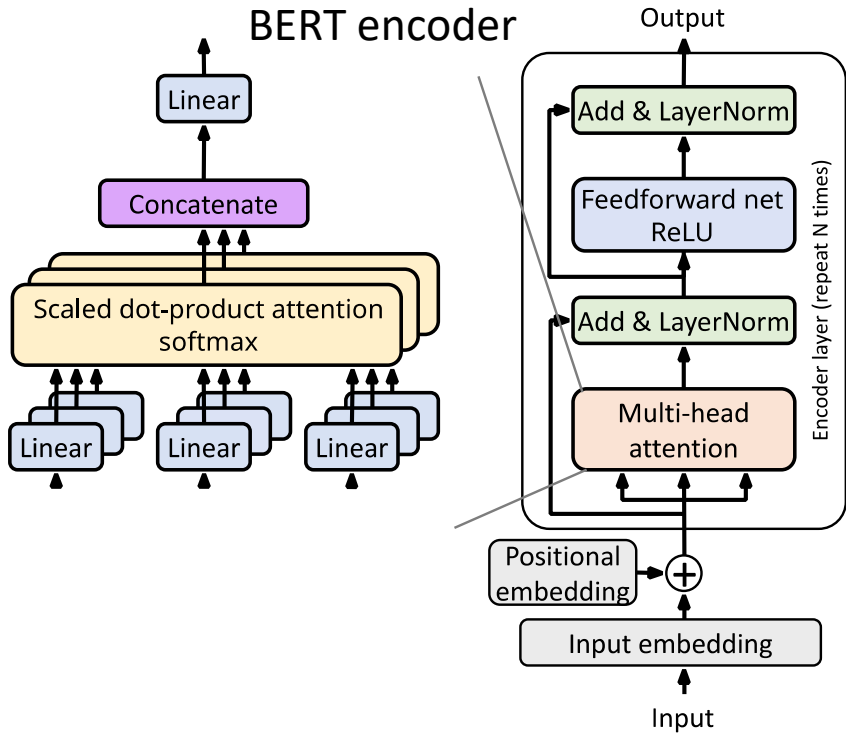
High-Performance Communication

- Use larger clusters (10k+ GPUs)
- Model parallelism
 - Complex pipeline schemes
- Optimized networks

Distribution and Parallelism

Data	Pipeline	Operator
<div data-bbox="1758 1099 1987 1213" style="border: 1px solid black; padding: 2px;"> <p>SPCL ML: High-Performance Sparse Communication for Machine Learning</p> <p>Yuan Zhang, ...</p> </div>	<div data-bbox="2025 1099 2254 1213" style="border: 1px solid black; padding: 2px;"> <p>Chimera: Efficiently Training Large-Scale Neural Networks with Bidirectional Pipelines</p> <p>Yuan Zhang, ...</p> </div>	<div data-bbox="2293 1099 2522 1213" style="border: 1px solid black; padding: 2px;"> <p>Real-Time Publishing Revisited: Near-Optimal Parallel Matrix-Matrix Multiplication</p> <p>Yuan Zhang, ...</p> </div>
<div data-bbox="1758 1228 1987 1342" style="border: 1px solid black; padding: 2px;"> <p>Demystifying Parallel and Distributed Deep Learning As the Depth Concurrency Analysis</p> <p>Yuan Zhang, ...</p> </div>	<div data-bbox="2025 1228 2254 1342" style="border: 1px solid black; padding: 2px;"> <p>Performance-Focused Training of 10k+ GPUs and Beyond: From Pipelines to Tensor Parallelism</p> <p>Yuan Zhang, ...</p> </div>	<div data-bbox="2293 1228 2522 1342" style="border: 1px solid black; padding: 2px;"> <p>Real-Time Publishing Revisited: Near-Optimal Parallel Matrix-Matrix Multiplication</p> <p>Yuan Zhang, ...</p> </div>

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



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%

highly optimized

Our performance improvement for BERT-large

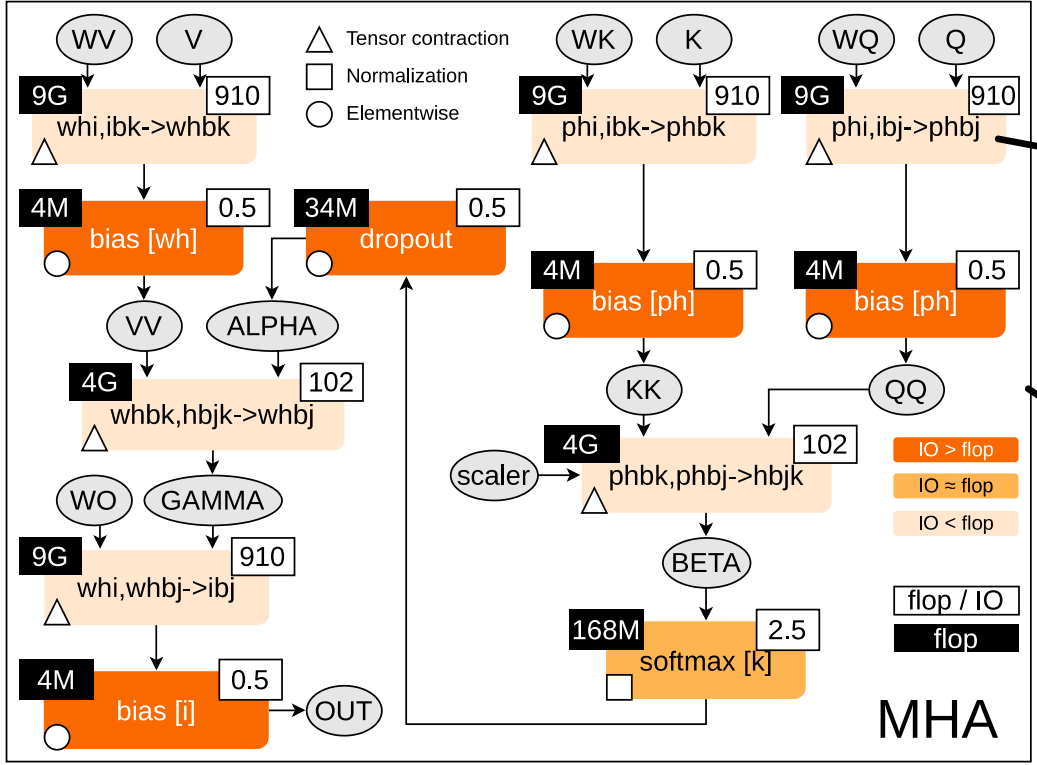
- 30% over PyTorch
- 20% over Tensorflow + XLA
- 8% over DeepSpeed

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

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

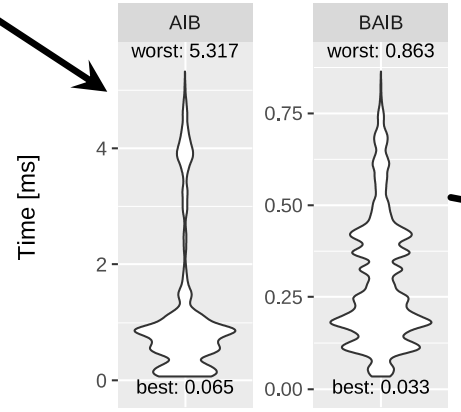
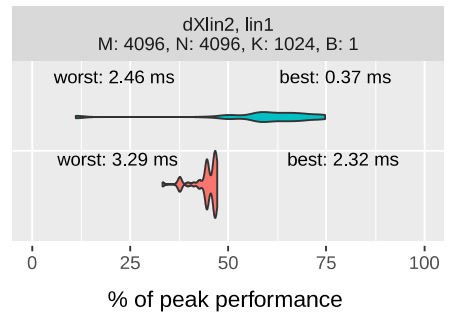
Last week, OpenAI published a paper detailing GPT-3, a machine learning model that achieves strong results on a number of natural language benchmarks. A 175 billion parameters where a parameter affects data's prominence in an overall prediction, it's the largest of its kind. And with a memory size exceeding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

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

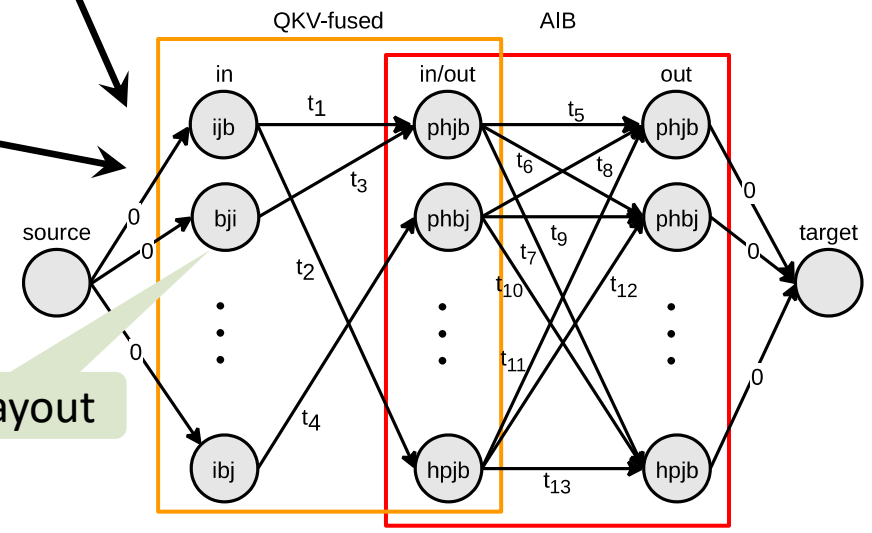


different data layouts

different fusion strategies



Configuration selection graph



data layout

fusion strategy

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

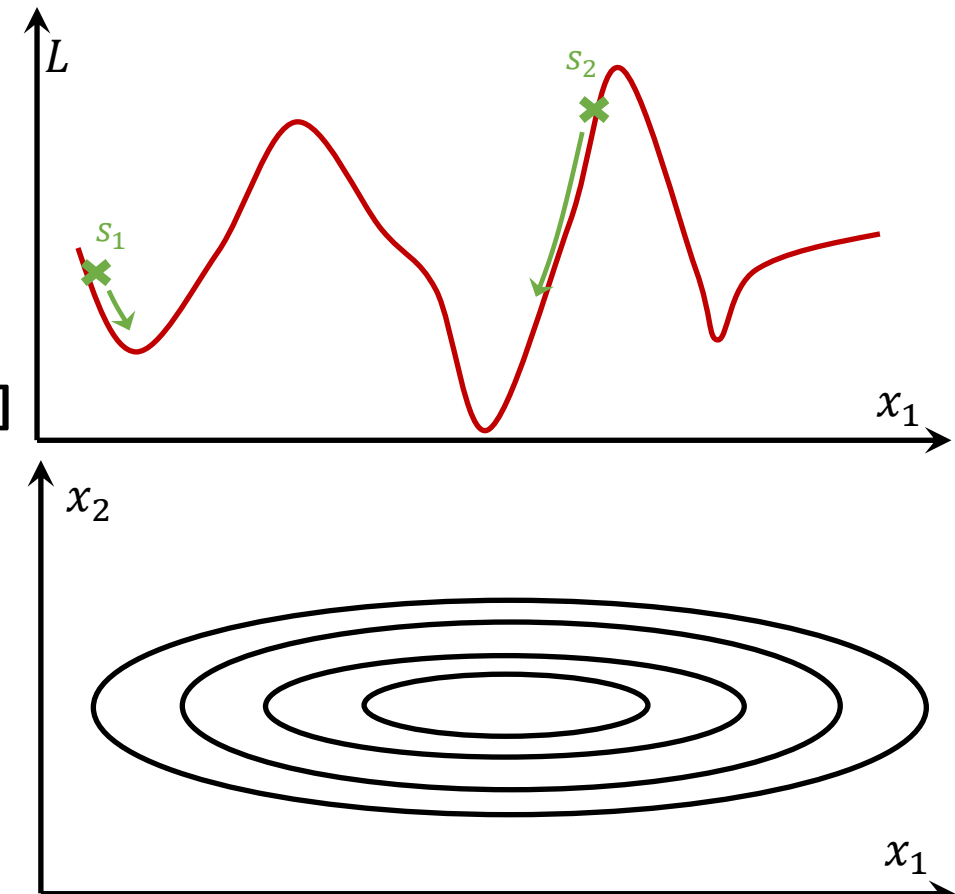
Moving Data is Most Expensive!

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!
- **GPT-3 has up to 175 billion parameters**
 - 700 GiB in FP32, 350 GiB in FP16/BF16 ☹️
 - 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]**

$$\partial E = \underbrace{\left(\frac{\partial E}{\partial w}\right)^T}_{\text{Gradient } (\approx 0)} \partial w + \underbrace{\frac{1}{2} \partial w^T \left(\frac{\partial^2 E}{\partial^2 w}\right) \partial w}_{\text{“Curvature” of error (aka. “sensitivity”)}} + \underbrace{O(|\partial w|^3)}_{\text{Higher-order terms (=0 for quadratic loss)}}$$



[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}

$$\operatorname{argmin}_{\hat{w}} \|wx - \hat{w}x\|^2$$
- 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 \rightarrow 66 GiB, fits in a single (high-end) A100 GPU!

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.

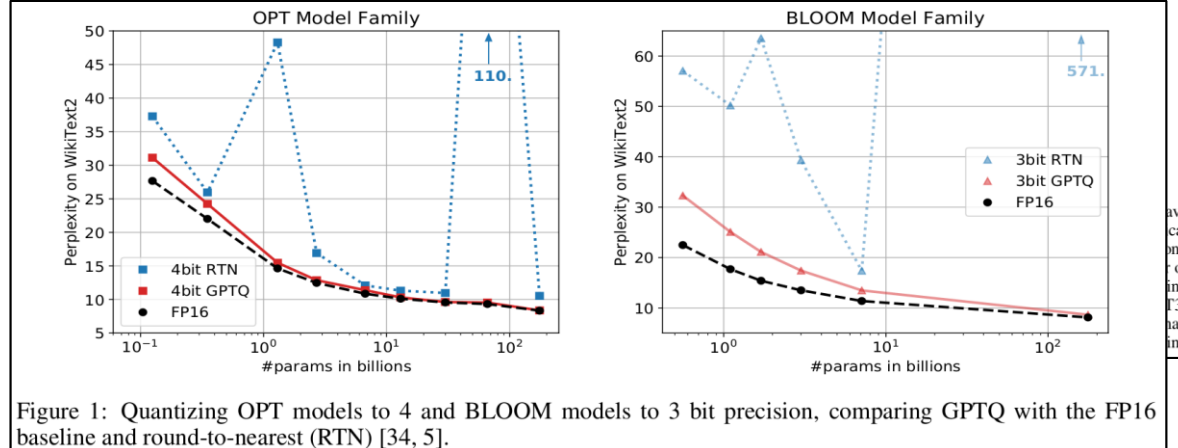


Figure 1: Quantizing OPT models to 4 and BLOOM models to 3 bit precision, comparing GPTQ with the FP16 baseline and round-to-nearest (RTN) [34, 5].

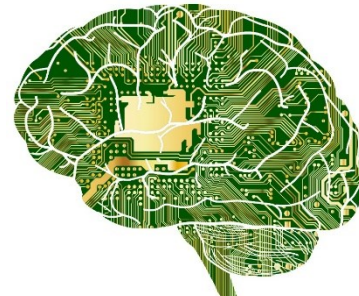
Model	FP16	1024	512	256	128	64	32	3-bit
OPT-175B	8.34	11.84	10.85	10.00	9.58	9.18	8.94	8.68
BLOOM	8.11	11.80	10.84	10.13	9.55	9.17	8.83	8.64

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

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!

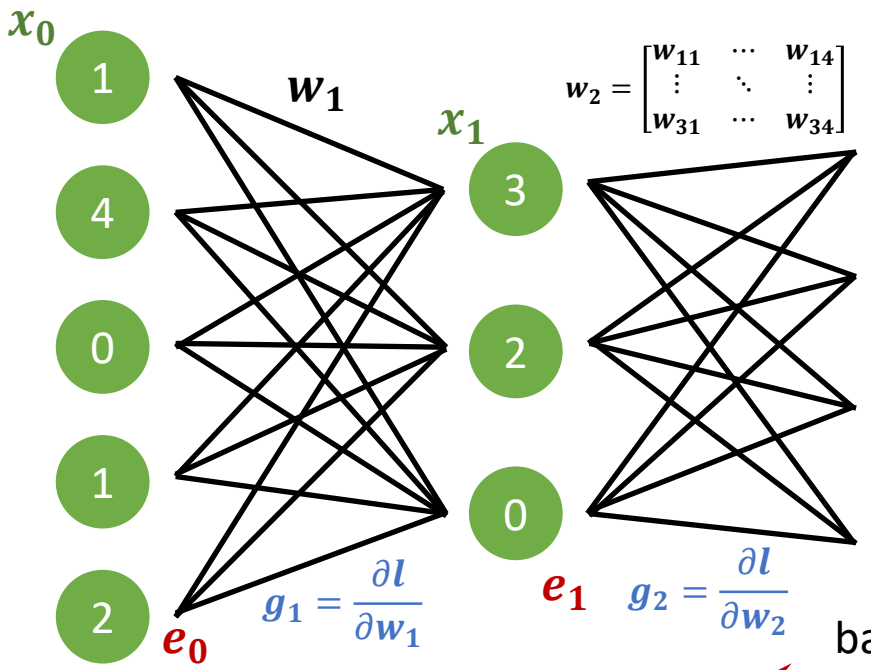
Intuition: not **all** features are **always** relevant!

- 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

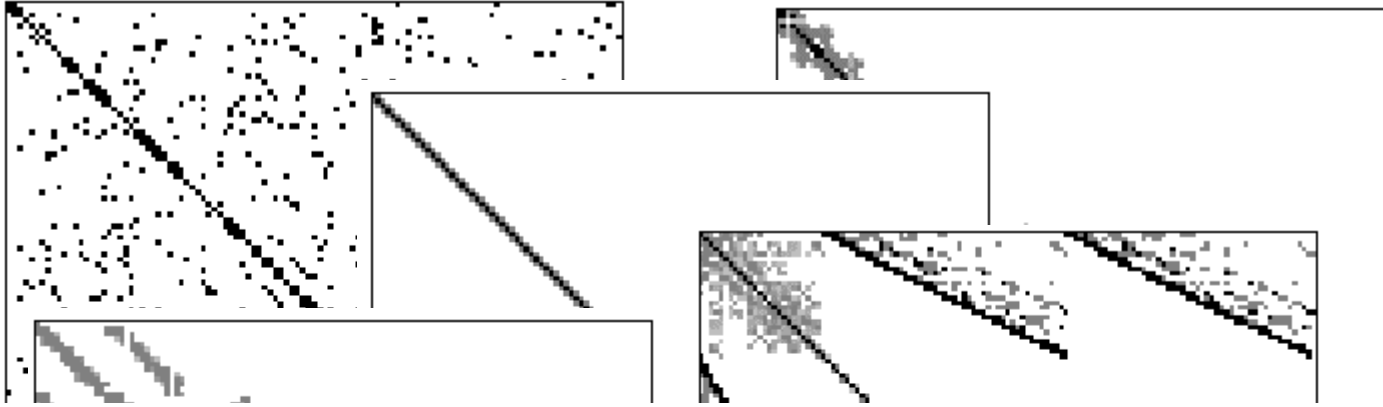
the size of neural networks, sparse reduce the per growing live tutorial elements of sparsity in actioners forward. We mena such, and show r efficiency ng on how

as few as experience - stein, 1933

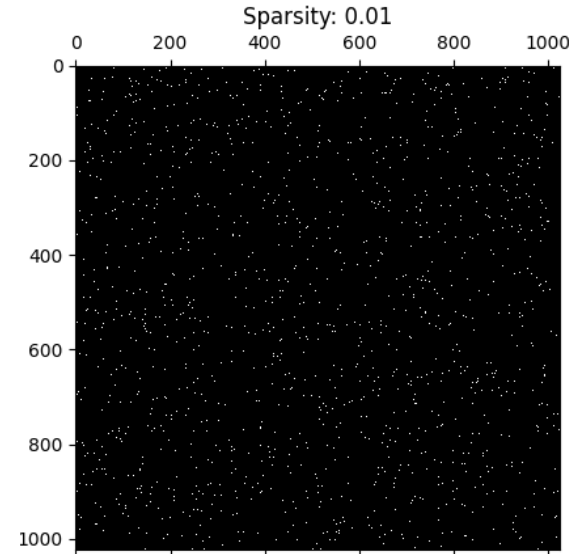
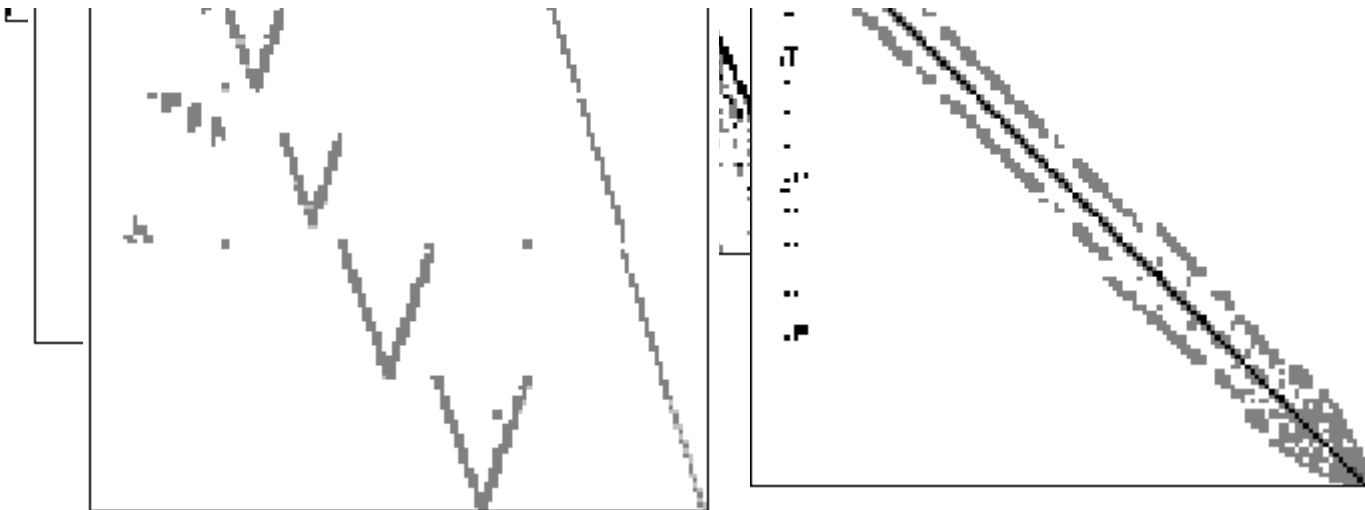
1 INTRODUCTION

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

Sparse ML Computations – Very Different from Scientific Computing!

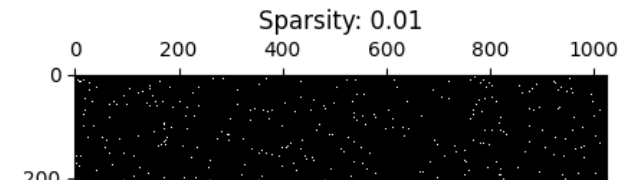


Sparse Matrices from Scientific Computing are quite structured!



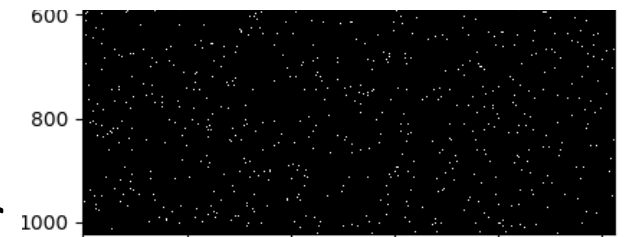
WK

Sparsified BERT
WK and WQ matrices
(3rd encoder)



WQ

Sparse Matrices in Deep Learning are quite uniform(ly random)!



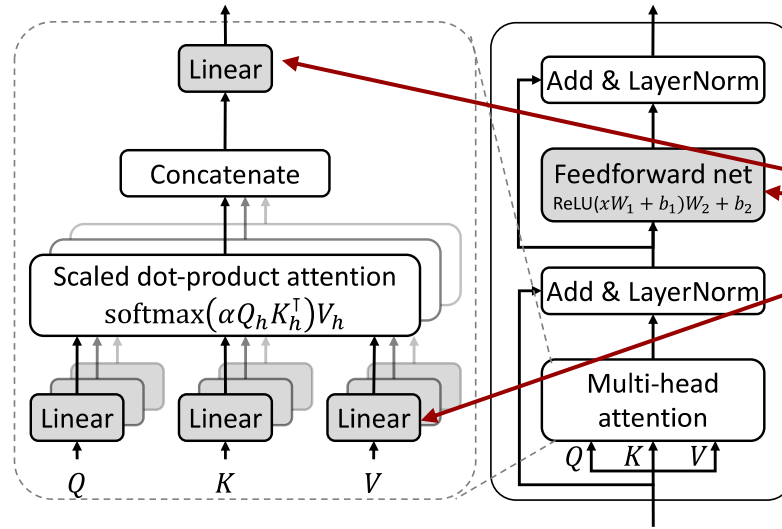
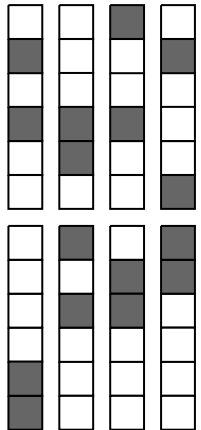
Performance of Sparse Models (PyTorch Sten)

Custom implementation of matrix multiplication:
sparse @ dense -> dense

Linear layer: $y = xW + b$

dense dense **sparse** dense

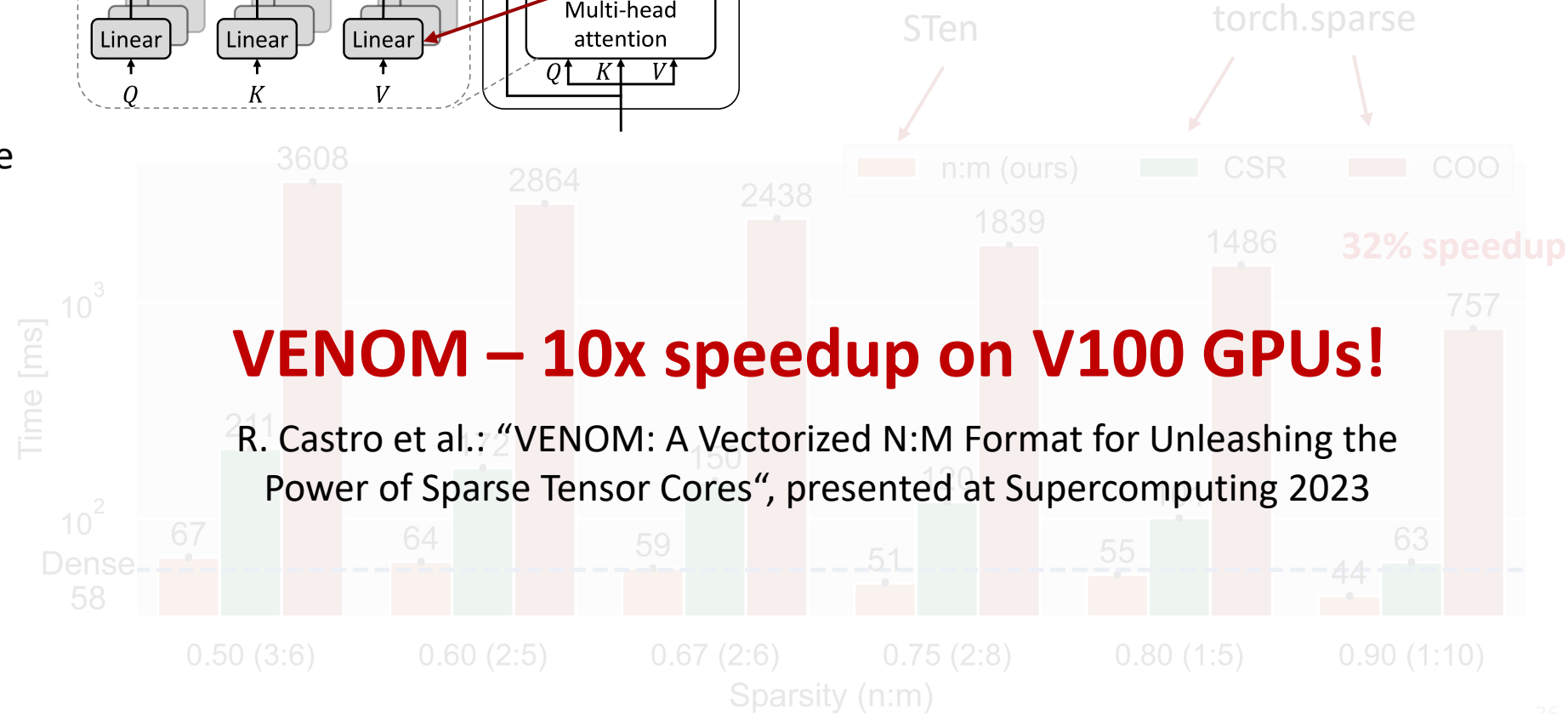
2:6 sparse format



BERT (base) from HuggingFace

- batch size 8
- sequence length 128

Sparsified linear layer weights
Intel i7-4770 CPU



VENOM – 10x speedup on V100 GPUs!

R. Castro et al.: “VENOM: A Vectorized N:M Format for Unleashing the Power of Sparse Tensor Cores”, presented at Supercomputing 2023

Quantization and Model Compression Together Enable

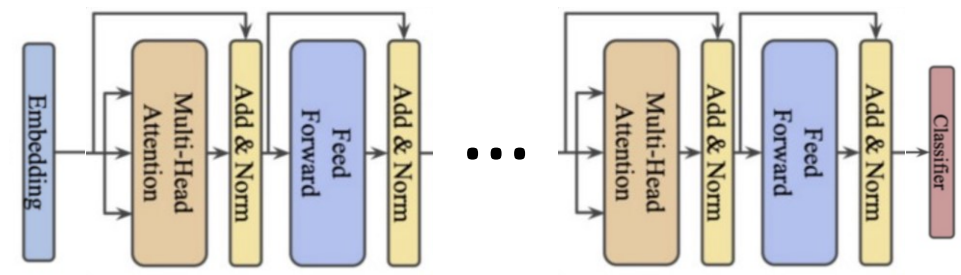
100x

More Efficient Processing

Which Makes Data Movement Even More Important!

Especially in the Network!

Three Systems Dimensions in Large-scale Super-learning ...



High-Performance I/O

- Quickly growing data volumes
 - Scientific computing!
- Use the specifics of machine learning workloads
 - E.g., intelligent prefetching

21 Jan 2021

CLAIRVOYANT PREFETCHING FOR DISTRIBUTED MACHINE LEARNING I/O

Roman Böhringer¹ Nikoli Dryden¹ Tal Ben-Nun¹ Torsten Hoefler¹

ABSTRACT

I/O is emerging as a major bottleneck for machine learning training, especially in distributed environments such as clouds and supercomputers. Optimal data ingestion pipelines differ between systems, and increasing efficiency requires a delicate balance between access to local storage, external filesystems, and remote workers; yet existing frameworks fail to efficiently utilize such resources. We observe that, given the seed generating the random access pattern for training with SGD, we have *clairvoyance* and can exactly predict when a given sample will be accessed. We combine this with a theoretical analysis of access patterns in training and performance modeling to produce a novel machine learning I/O middleware, HDMLP, to tackle the I/O bottleneck. HDMLP provides an easy-to-use, flexible, and scalable solution that delivers better performance than state-of-the-art approaches while requiring very few changes to existing codebases and supporting a broad range of environments.

High-Performance Compute

- Deep learning is HPC
 - Data movement!
- Quantization, Sparsification
 - Drives modern accelerators!

Data Movement Is All You Need: A Case Study on Optimizing Transformers

Andrei Ivanov*, Nikoli Dryden*, Tal Ben-Nun, Shigang Li, Torsten Hoefler
ETH Zürich
firstname.lastname@inf.ethz.ch
* Equal contribution

Abstract—Transformers have become widely used for language modeling and sequence learning tasks, and are one of the most important machine learning workloads today. Training one is a very compute-intensive task, often taking days or weeks, and significant attention has been given to optimizing transformers. Despite this, existing implementations do not efficiently utilize GPUs. We find that data movement is the key bottleneck when training. Due to Amdahl's Law and massive improvements in compute performance, training has now become memory-bound. Further, existing frameworks use suboptimal data layouts. Using these insights, we present a recipe for globally optimizing data movement in transformers. We reduce data movement by up to 22.91% and overall achieve a 1.30x performance improvement over state-of-the-art frameworks when training BERT. Our approach is applicable more broadly to optimizing deep neural networks, and offers insight into how to tackle emerging performance bottlenecks.

challenges such as artificial general intelligence [27]. Thus, improving transformer performance has been in the focus of numerous research and industrial groups.

Significant attention has been given to optimizing transformers: local and fixed-window attention [28]–[32], more general structured sparsity [33], learned sparsity [34]–[36], and other algorithmic techniques [19], [37] improve the performance of transformers. Major hardware efforts, such as Tensor Cores and TPUs [38] have accelerated tensor operations like matrix-matrix multiplication (MMM), a core transformer operation. Despite this, existing implementations do not efficiently utilize GPUs. Even optimized implementations such as Megatron [18] report achieving only 30% of peak GPU flops.

We find that the key bottleneck when training transform-

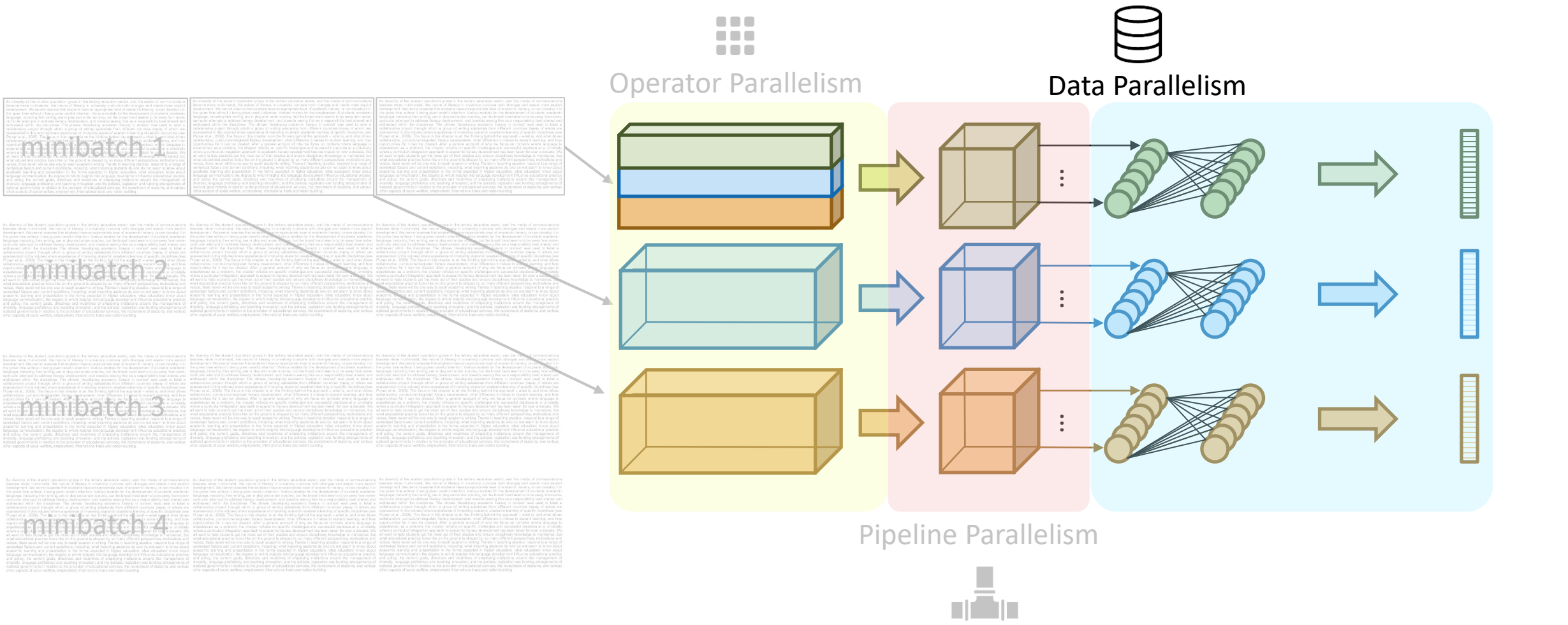
High-Performance Communication

- Use larger clusters (10k+ GPUs)
- Model parallelism
 - Complex pipeline schemes
- Optimized networks

Distribution and Parallelism

Data	Pipeline	Operator
<p>SPCL: High-Performance Sparse Communication for Machine Learning</p> <p>Yuan Zhang, Nikoli Dryden, Tal Ben-Nun, Shigang Li, Torsten Hoefler</p> <p>Abstract—Sparse communication is a key challenge in training large-scale deep learning models. We propose SPCL, a high-performance sparse communication framework for machine learning. SPCL is designed to be a drop-in replacement for existing communication libraries, and it achieves a 1.30x performance improvement over state-of-the-art frameworks when training BERT.</p>	<p>Chimera: Efficiently Training Large-Scale Neural Networks with Bidirectional Pipelines</p> <p>Yuan Zhang, Nikoli Dryden, Tal Ben-Nun, Shigang Li, Torsten Hoefler</p> <p>Abstract—Training large-scale neural networks with bidirectional pipelines is a challenging task. We propose Chimera, a high-performance training framework for large-scale neural networks. Chimera achieves a 1.30x performance improvement over state-of-the-art frameworks when training BERT.</p>	<p>Red Blue Picking Revisited: Near-Optimal Parallel Matrix-Matrix Multiplication</p> <p>Yuan Zhang, Nikoli Dryden, Tal Ben-Nun, Shigang Li, Torsten Hoefler</p> <p>Abstract—Matrix-matrix multiplication (MMM) is a core operation in many machine learning workloads. We propose Red Blue Picking, a near-optimal parallel MMM algorithm. Red Blue Picking achieves a 1.30x performance improvement over state-of-the-art frameworks when training BERT.</p>

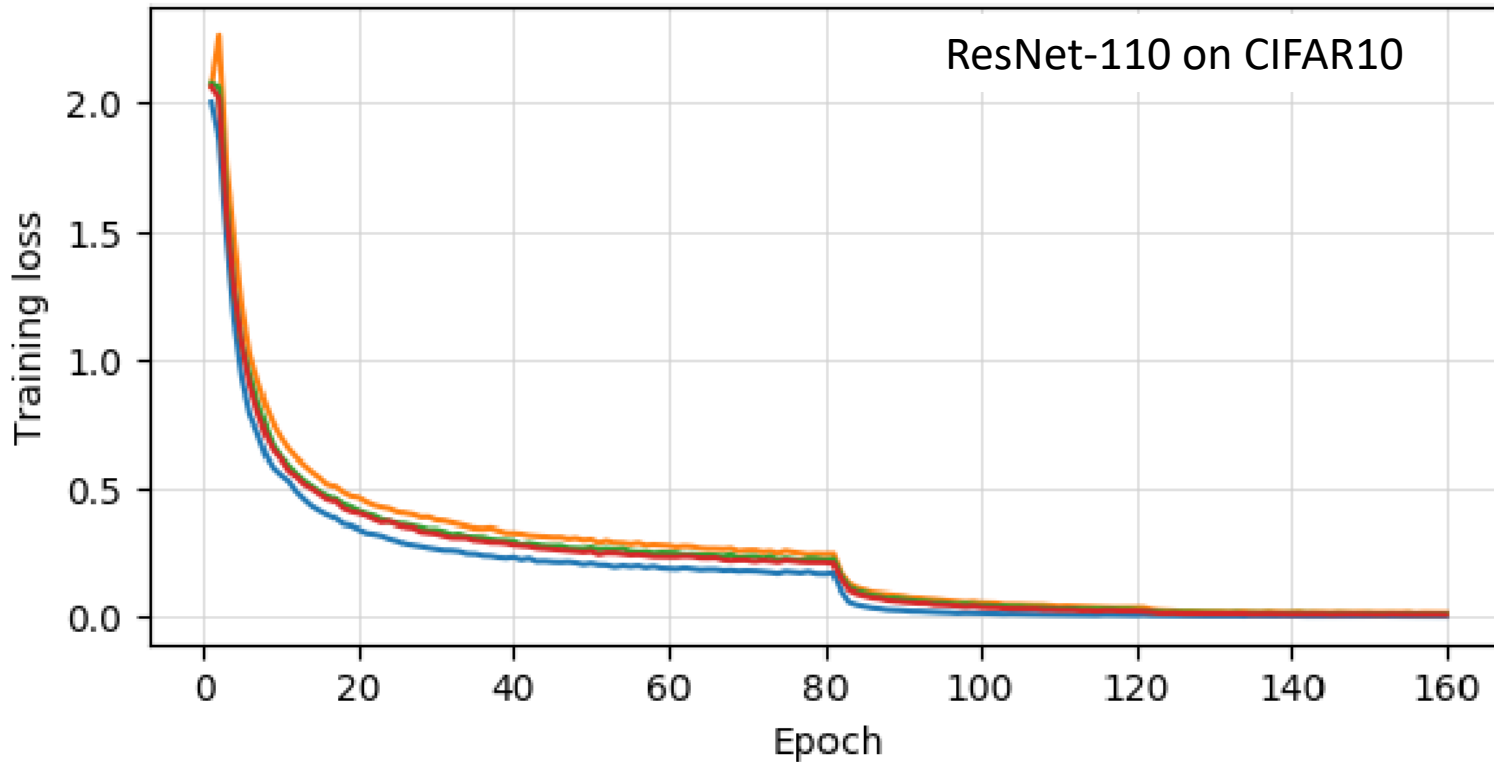
The Three Dimensions of Parallelism in Deep Learning (arXiv:1802.09941)



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])



Assumptions

Discuss Section

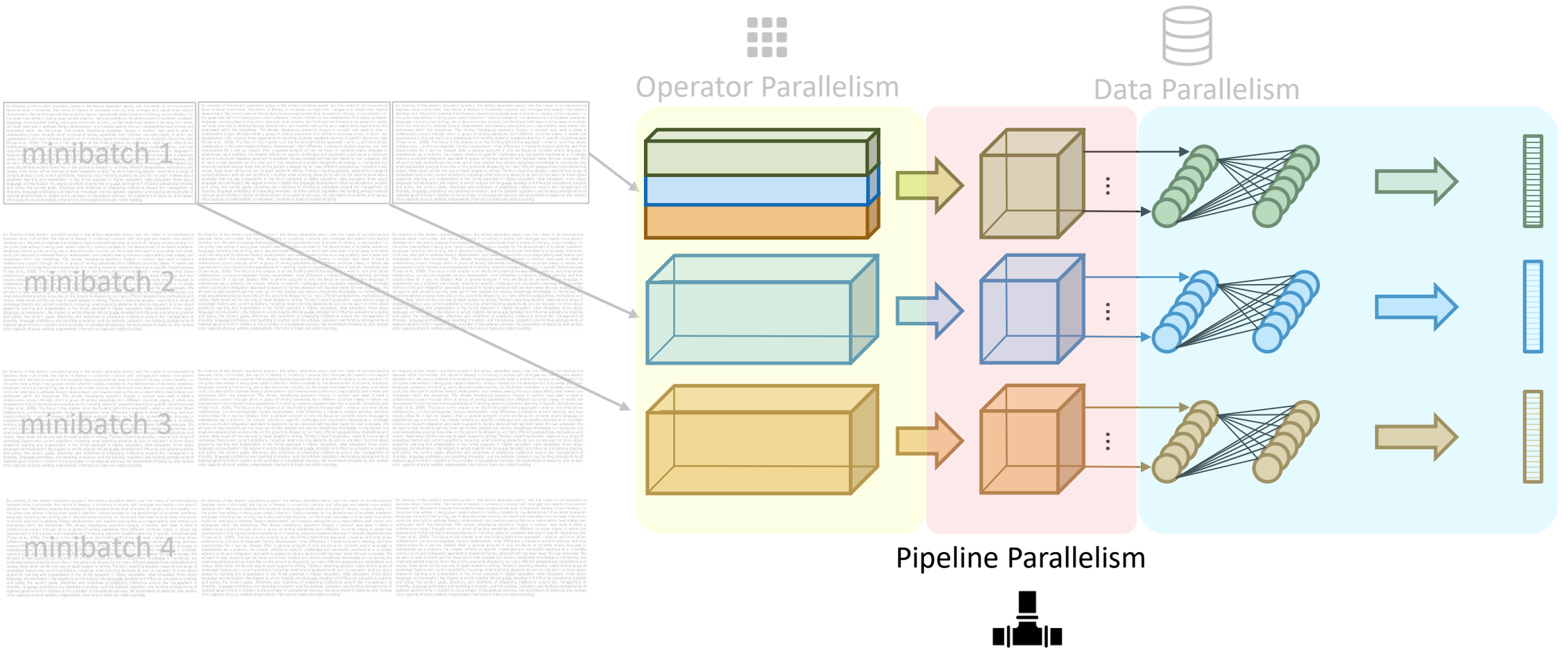
we:

1.

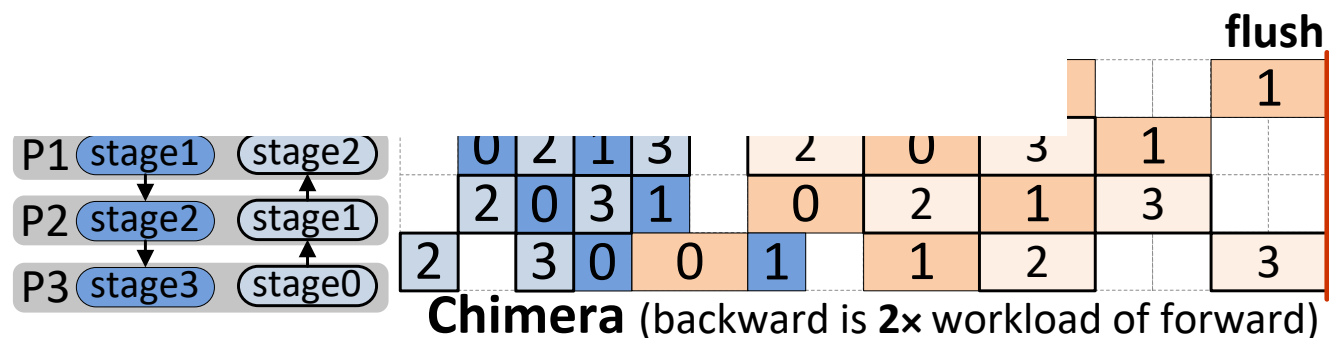
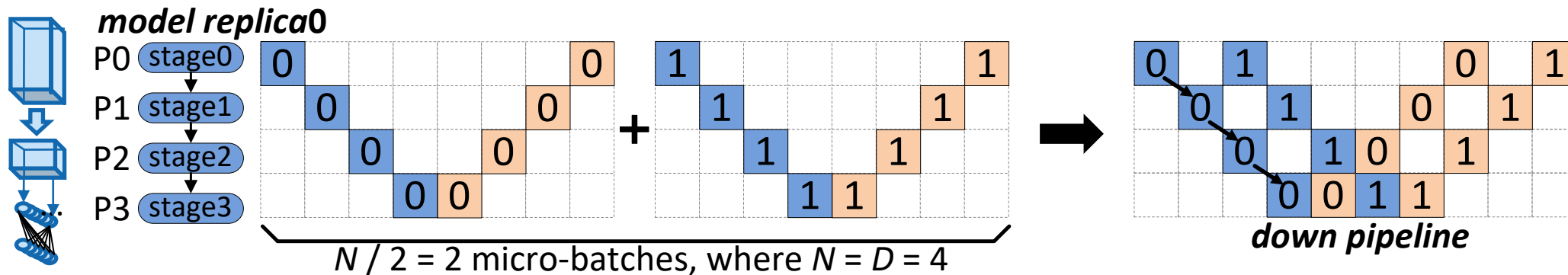
tasks in

— Baseline — TopK [K=0.025%] — TopK [K=0.1%] — TopK [K=0.2%]

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

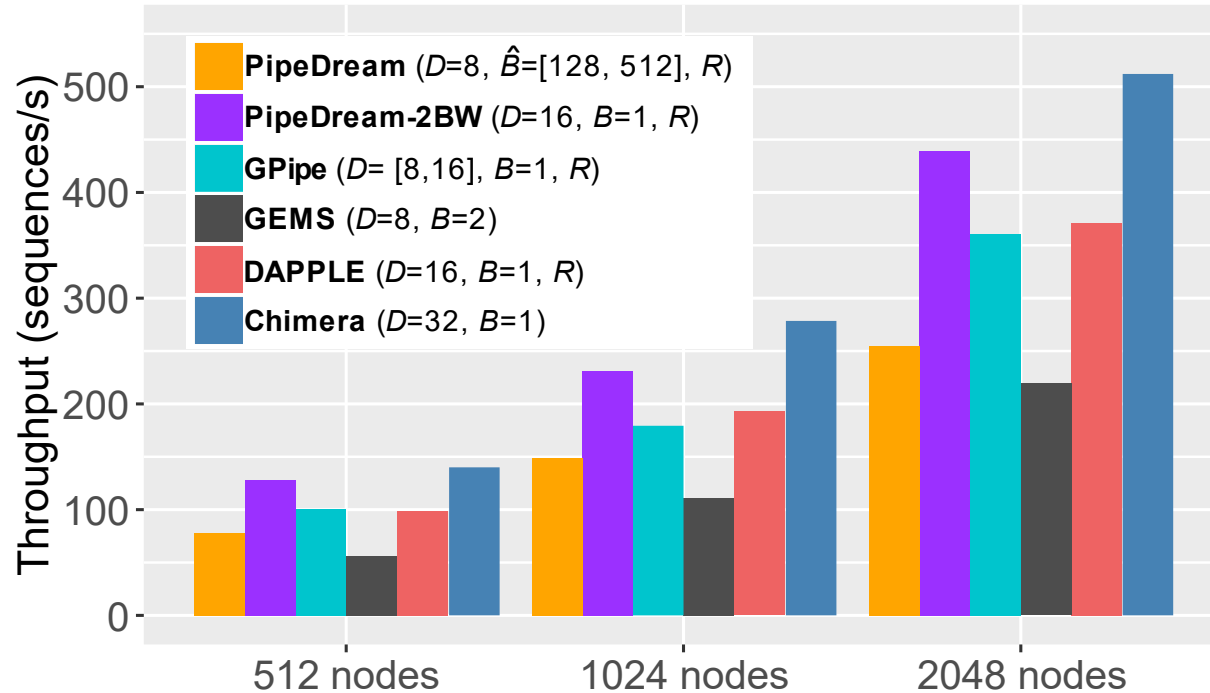


Bidirectional Pipelines – Meet Chimera (arXiv: 2107.06925v3)





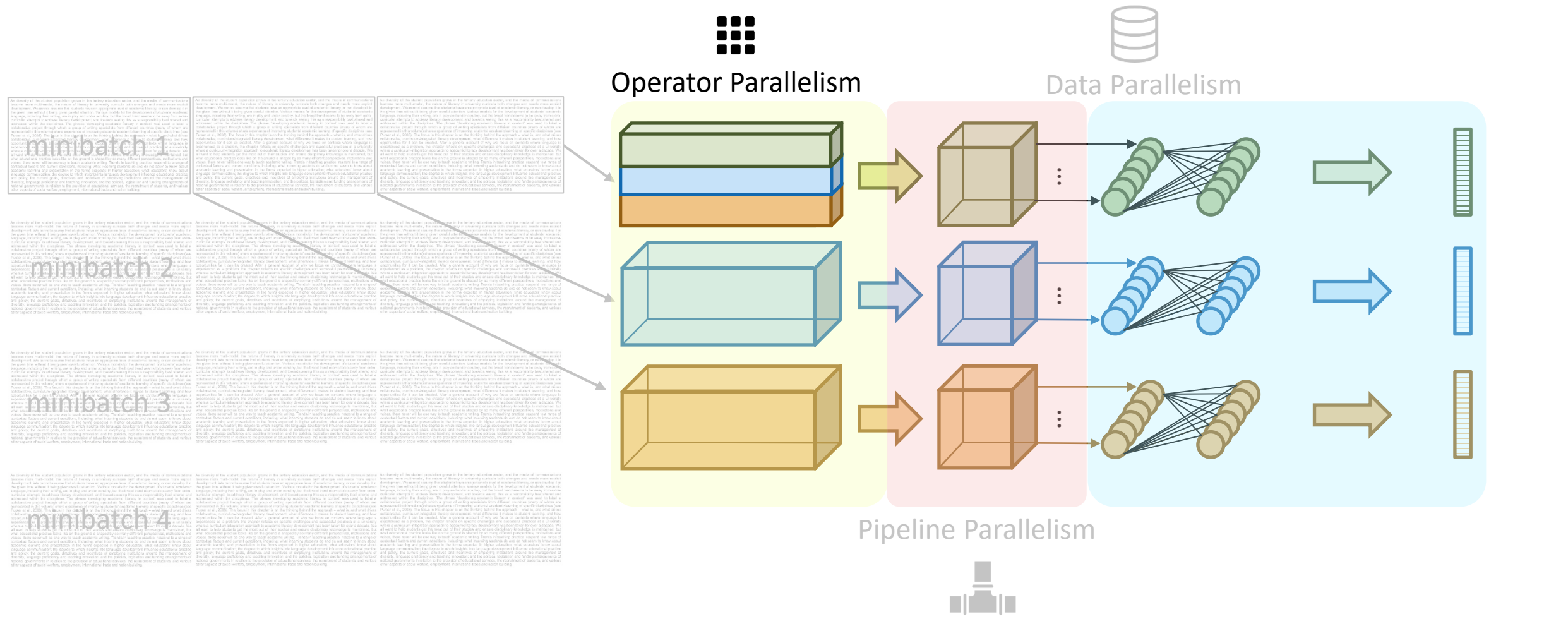
Chimera Weak Scaling (arXiv: 2107.06925v3)



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

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!

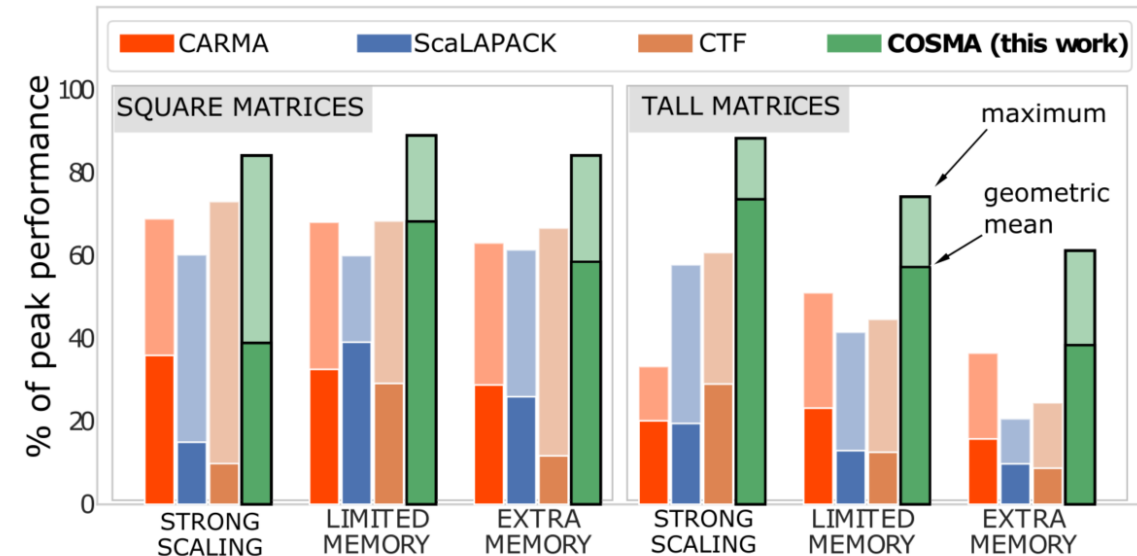
COSMA [1] communication-optimal distributed MMM

- 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

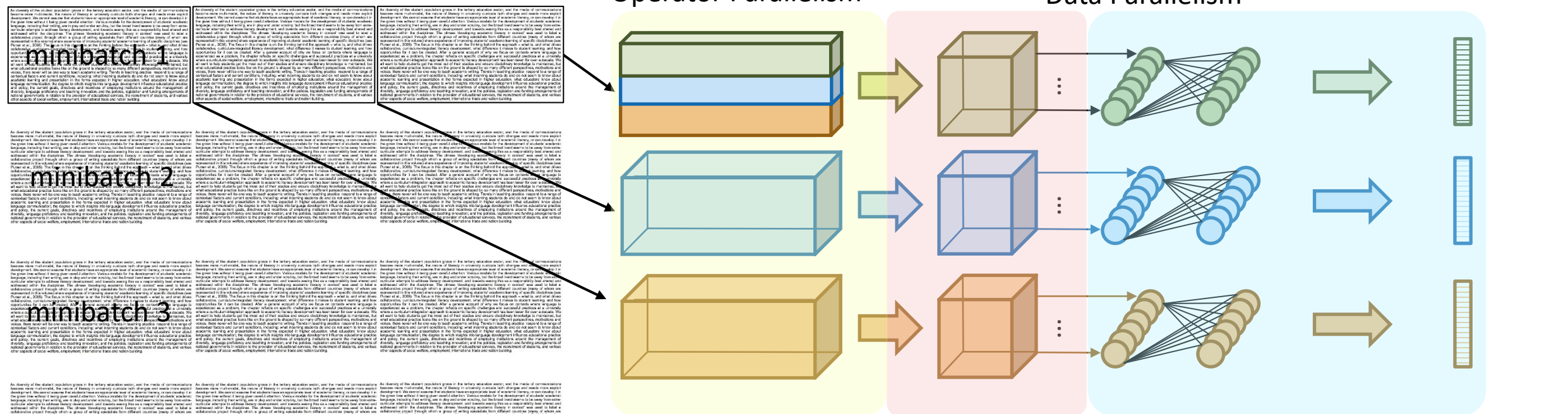
The Three Dimensions of Parallelism in Deep Learning (arXiv:1802.09941)



Operator Parallelism



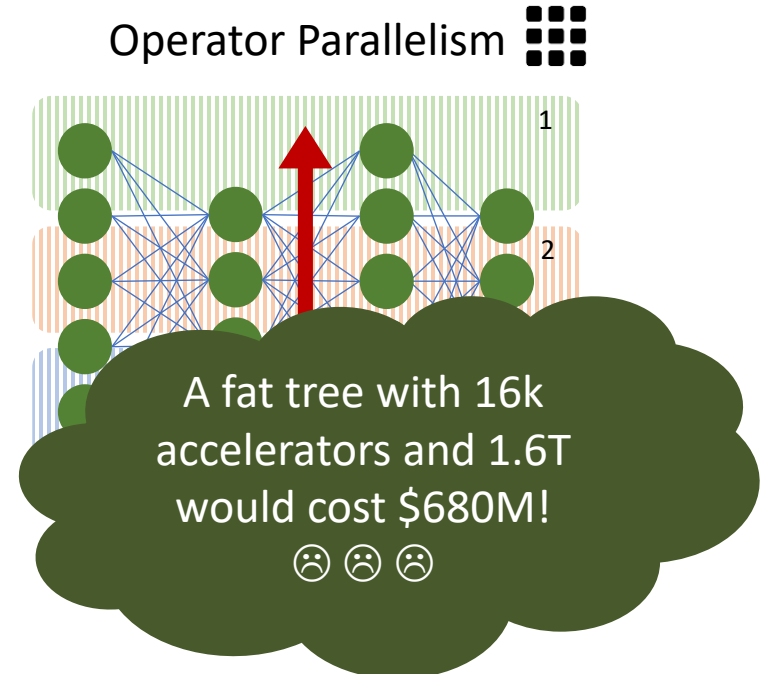
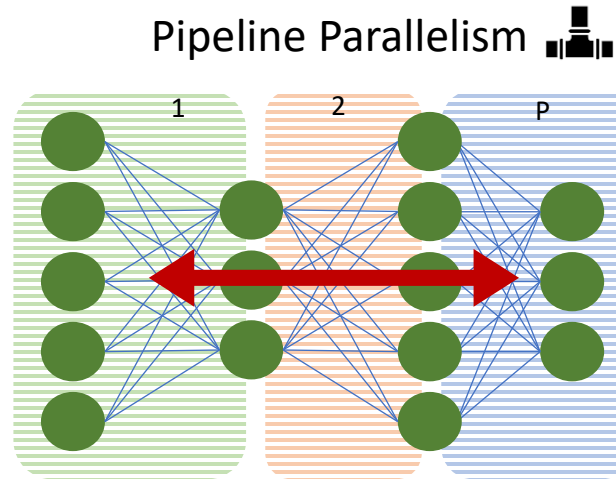
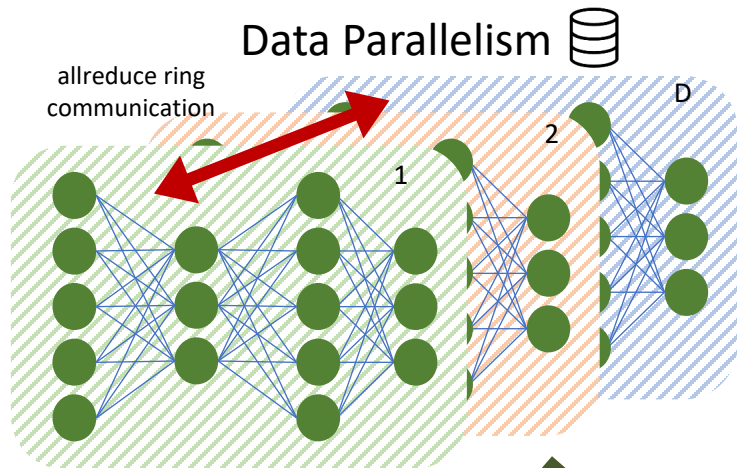
Data Parallelism



Pipeline Parallelism

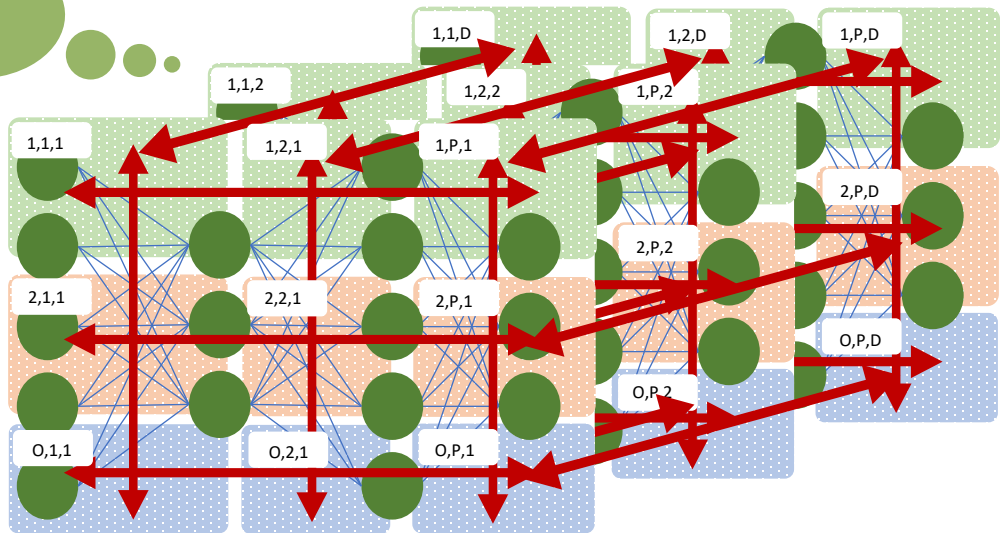


Communications in 3D Parallelism in Deep Learning (arXiv:2209.01346)



Communication is (largely) a logical 3D Torus

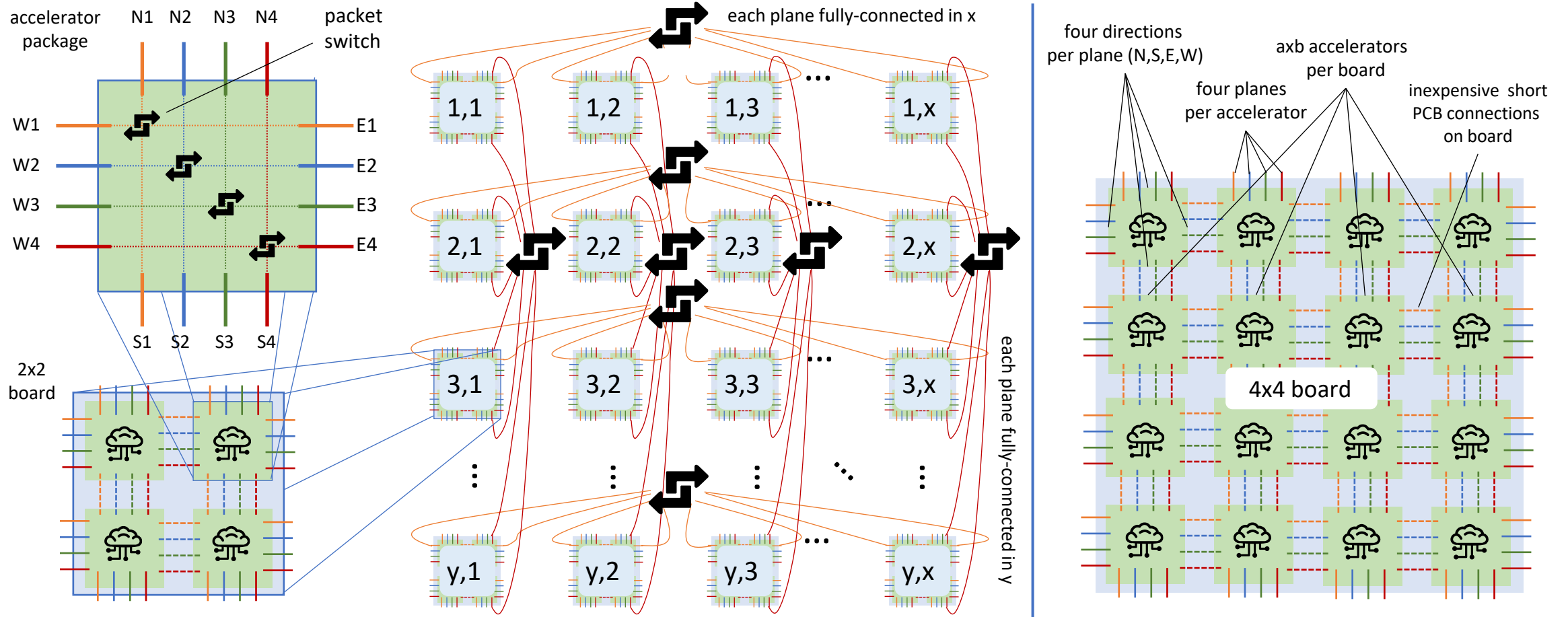
3D - Data, Pipeline, and Operator Parallelism



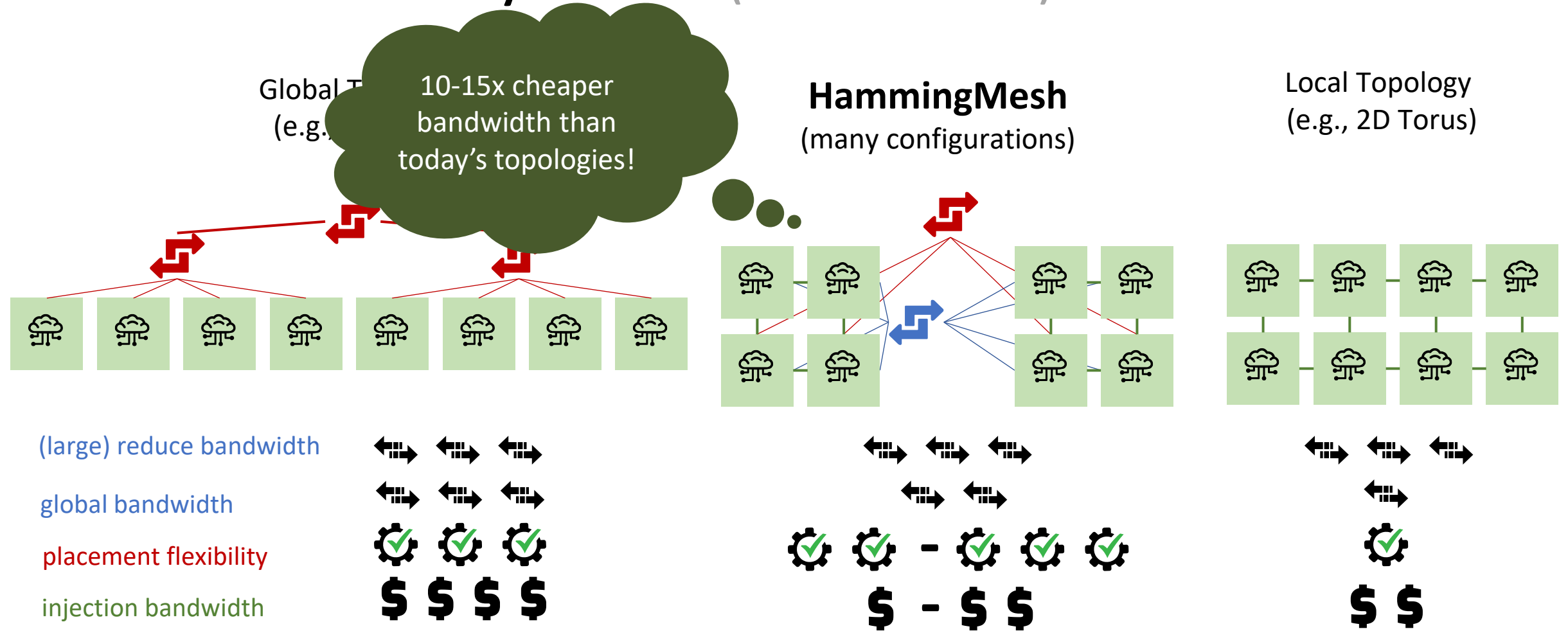
AI bandwidth today / yesterday (and growing!)

- 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

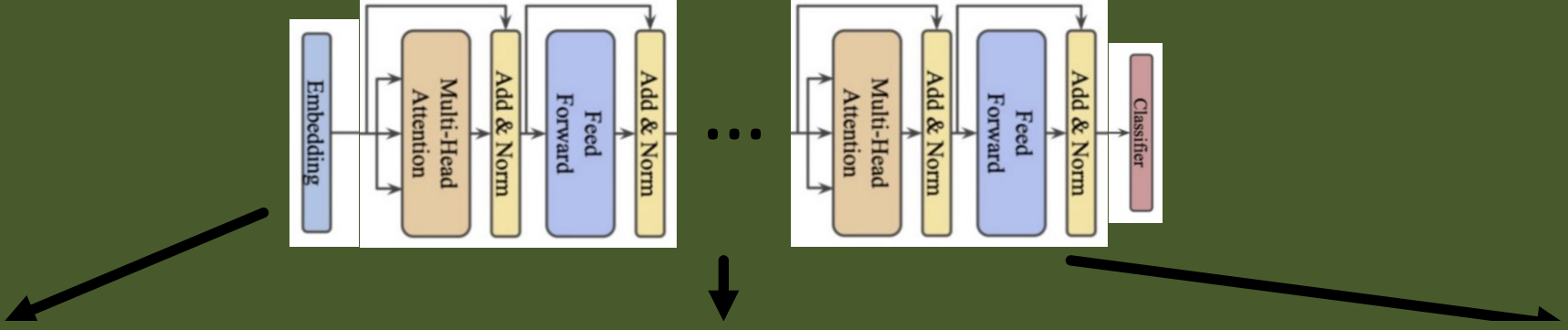
Co-designing an AI Supercomputer with Unprecedented and Cheap Bandwidth



Bandwidth-cost-flexibility Tradeoffs (arXiv:2209.01346)



Three Systems Dimensions in Large-scale Super-learning ...



Altogether, we discussed a cost / performance improvement of

1,000x

What will the (near future bring)?

Some predictions for the future of HPC but also computing at large!

Prediction 1: Accelerators Converge

AI is a gravity well – HPC will follow

Future Accelerators ...

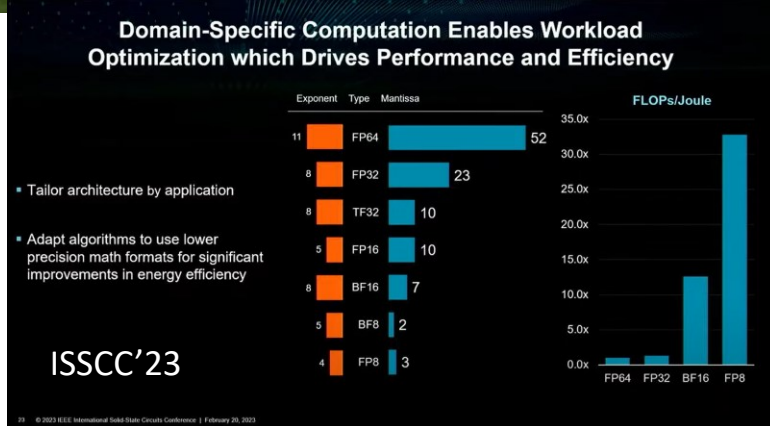
- **Most of the performance will be low precision arithmetic!**
 - I would predict (C)FP8 or smaller
 - We can be lucky if we get some fp64!

- **They will support quantization and sparsity in hardware**
 - Vector scaling and zero points

- **They will heavily be optimized towards data movement**
 - Physical limits and cost introduce two fundamental constraints:
Latency will become a problem
Locality and sparse connectivity
 - Potentially hard to program



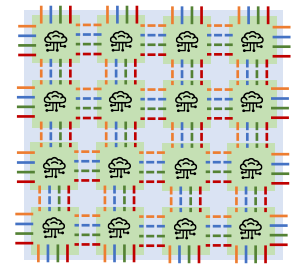
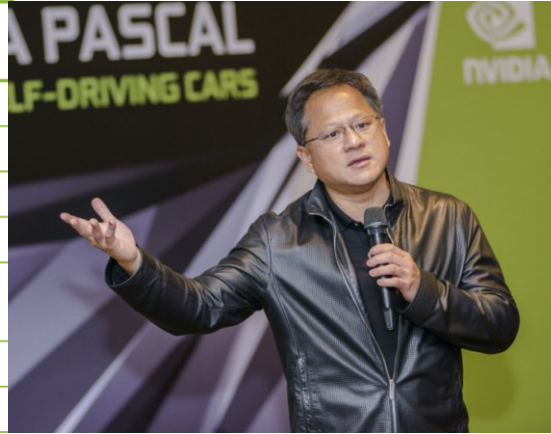
B. Wisniewski (Samsung)
Memory-coupled Compute
 SPCL_Bcast 01/19/23
<https://www.youtube.com/watch?v=KCrQtpx31CQ>



SPECIFICATIONS

	H100 SXM
FP64	34 TFLOPS
FP64 Tensor Core	67 TFLOPS
FP32	67 TFLOPS
TF32 Tensor Core	989 TFLOPS*
BFLOAT16 Tensor Core	1,979 TFLOPS*
FP16 Tensor Core	1,979 TFLOPS*
FP8 Tensor Core	3,958 TFLOPS*
INT8 Tensor Core	3,958 TOPS*

*30x** (indicated by a red arrow pointing from FP64 Tensor Core to FP8 Tensor Core)



Optimized topologies and network technologies.
 E.g., HammingMesh
<https://www.youtube.com/watch?v=xxwT45ljG4o>

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

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 runtime³. 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.

to appear at ICLR'24

arXiv:2306.03078v1 [cs.CL] 5 Jun 2023



Prediction 2: Programming and Tools Converge

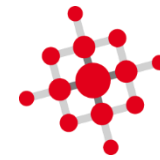
Data Science as a gravity well – HPC will follow

Scientific Computing is Moving to Python (as language frontend/ecosystem)



Tiobe Index June'23

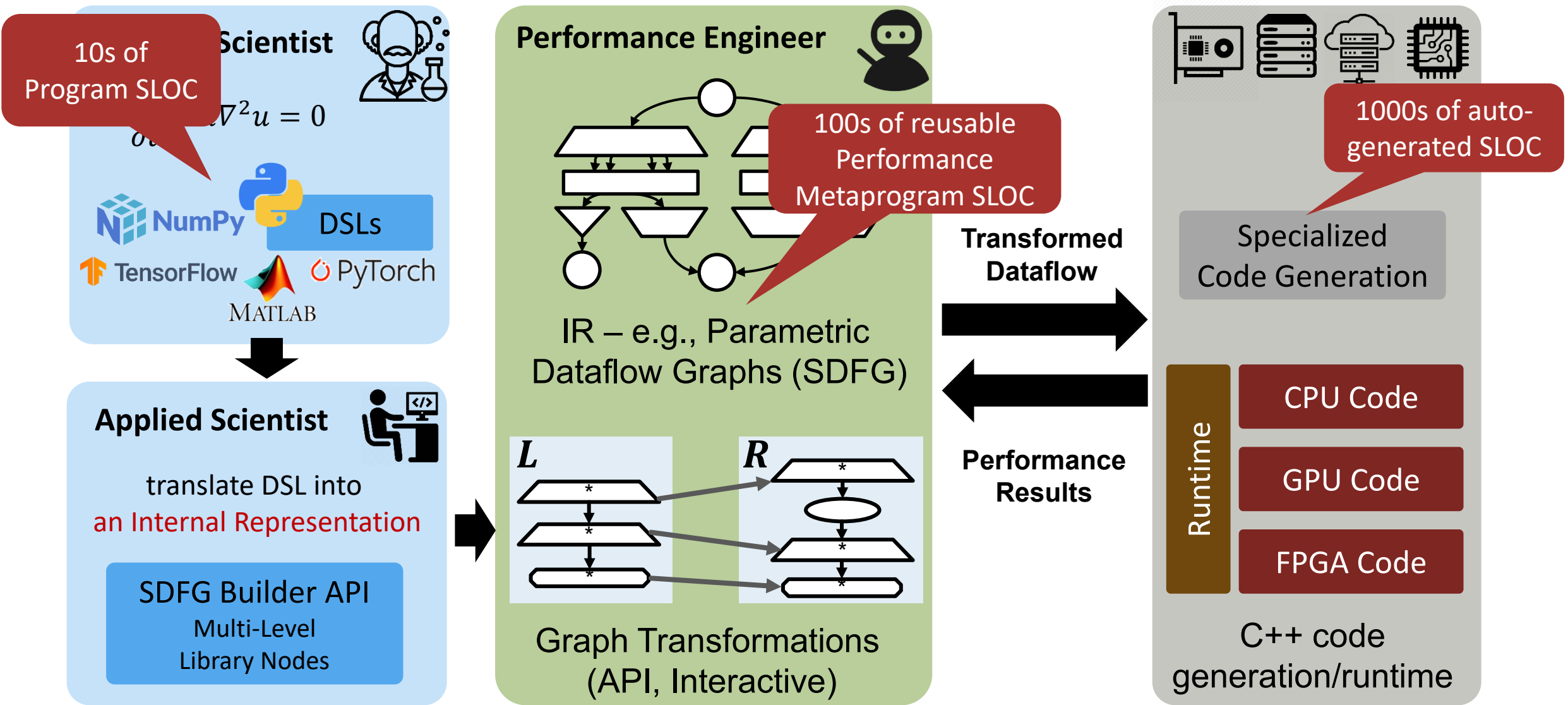
Feb 2023	Programming Language	Ratings
1	Python	15.16%
2	C	10.97%
3	C++	10.53%
4	Java	8.88%
5	C#	7.53%



439,100 projects

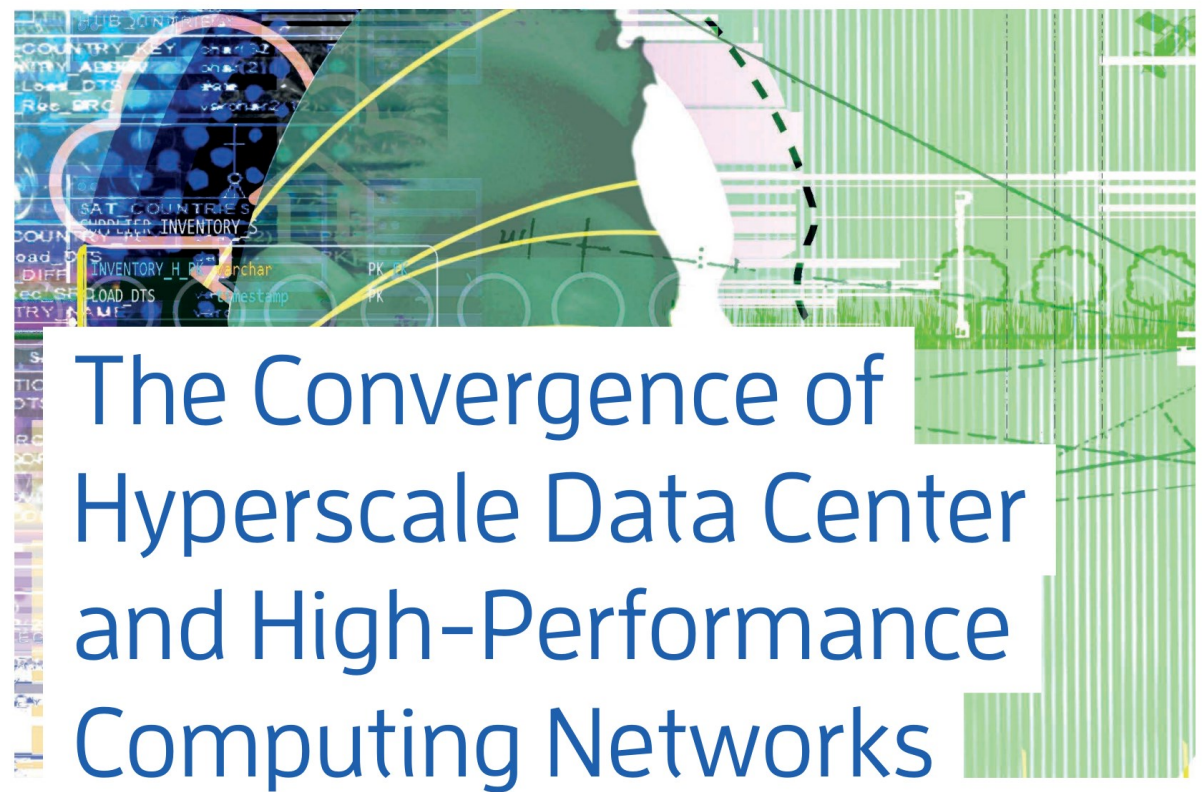


Upleveling Programming in the 21st Century – Performance Metaprogramming



Prediction 3: Networks Converge

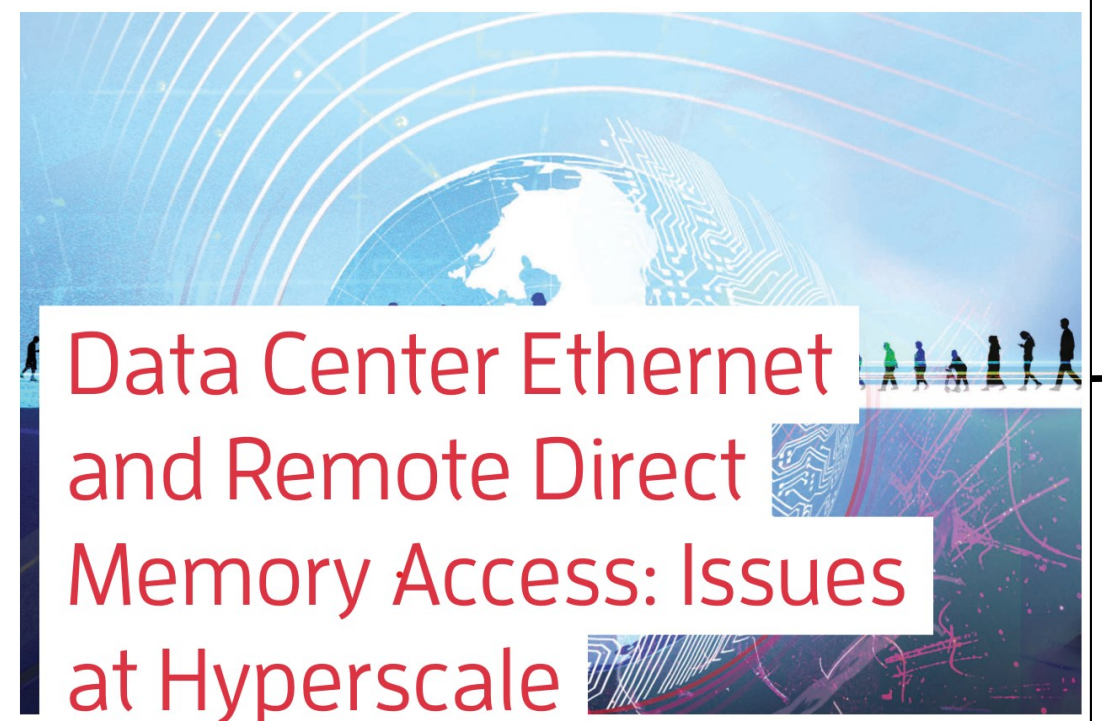
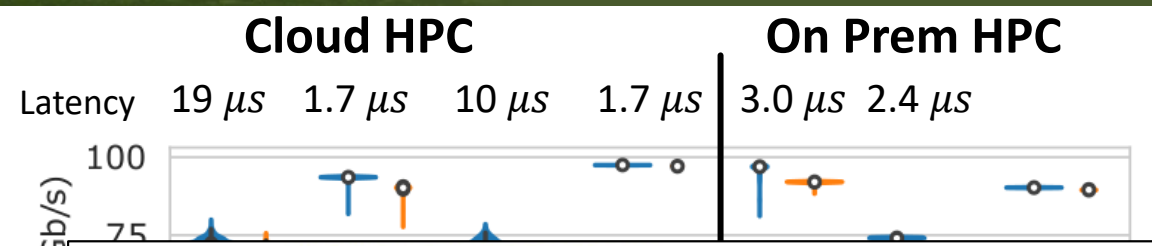
Cloud as a gravity well – HPC will follow



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.



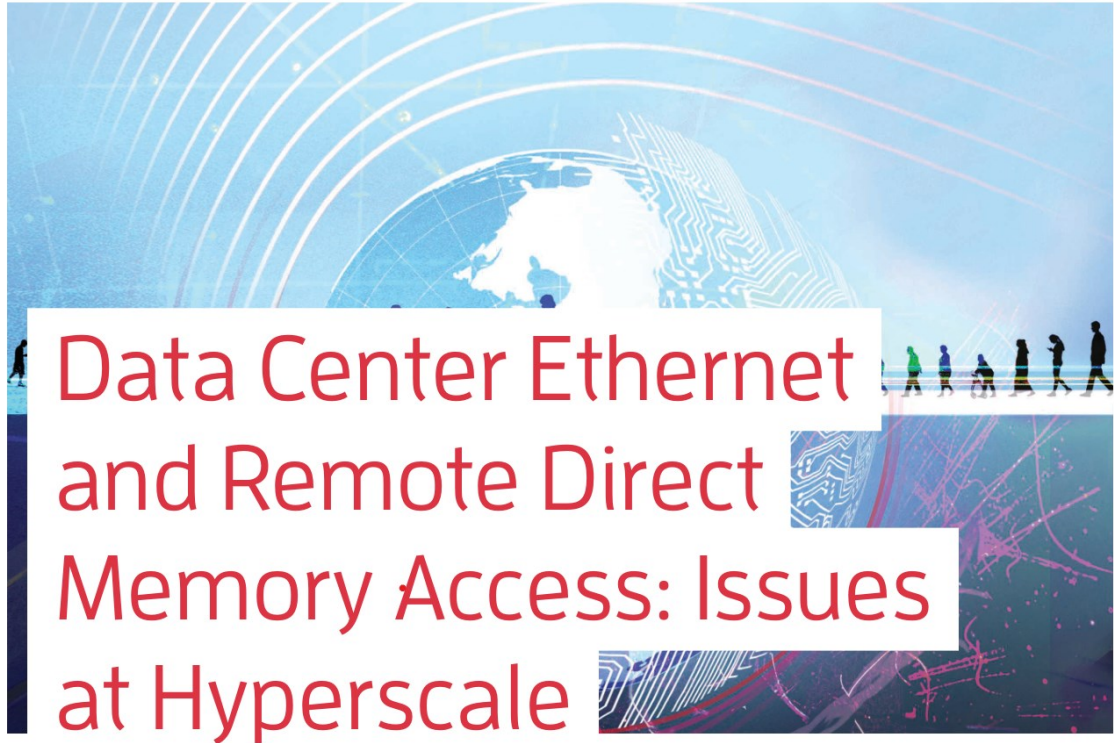
Data Center Ethernet and Remote Direct Memory Access: Issues at Hyperscale

Torsten Hoefler¹⁰, 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 Hoefler¹⁰, 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



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.

Key Points and Conclusions

Three Systems Dimensions in Large-scale Super-learning ...

High-Performance I/O

- Quickly growing data volumes
- Scientific computing!
- Use the specifics of machine learning workloads
- E.g., intelligent prefetching

High-Performance Compute

- Deep learning is HPC
- Data movement!**
- Quantization, Sparsification**
- Drives modern accelerators!

Data Movement is All You Need: A Case Study on Optimizing Transformers

High-Performance Communication

- Use larger clusters (10k+ GPUs)
- Model parallelism
- Complex pipeline schemes
- Optimized networks

Distribution and Parallelism

Data	Pipeline	Operator

Programming Sparse Models – Meet PyTorch Sten (arXiv:2304.07613)

Sparsity Layouts

- Dense Tensor
- Dimensions
- Strides
- Dense values

Sparse Tensor

- Dimensions
- Sparsity Format
- Compressed values

Operators

Layout → Implementation defined by input/output sparsity layout

Sparsifiers

Keep-all, Random fraction, Scalar threshold ...

Selected Available Sparsifiers:

Streaming

- Keep all
- do not drop
- Random fraction: drop if rand < 0.5
- Scalar threshold: drop if value < 0

Blocked

- Per block fraction
- Find block quartile q
- Drop if below

Materializing

- Scalar fraction
- Find quartile q
- Drop if below

Co-designing an AI Supercomputer with Unprecedented and Cheap Bandwidth

accelerator package (N1, N2, N3, N4) and packet switch (E1, E2, E3, E4) connected to a 2x2 board (S1, S2, S3, S4). The diagram shows a grid of accelerators (1,1 to 4,4) and boards (y,1 to y,X) with interconnections.

each plane fully-connected in x

four directions per plane (N,S,E,W)

4x4 board

inexpensive short PCB connections on board

AI supercomputer with 4x4 board

Prediction 1: Accelerators Converge

AI is a gravity well – HPC will follow

Prediction 2: Programming and Tools Converge

Data Science as a gravity well – HPC will follow

Prediction 3: Networks Converge

Cloud as a gravity well – HPC will follow

More of SPCL's research:

youtube.com/@spcl **180+ 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

Tarsten Hofer, ETH Zurich
Amit Hiran, Scale Computing
Duncan Russell, Hewlett Packard Enterprise

We discuss the differences and commonalities between network technologies used in supercomputers and data centers.

COVER FEATURE TECHNOLOGY PREDICTIONS

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

Tarsten Hofer, ETH Zurich
Duncan Russell, Keith Underwood, and Robert Averson, Hewlett Packard Enterprise
Mark Grawiec, Vahid Tabatabaee, Mohan Kulkarni, and Suresh Anandakrishnan, Broadcom
Suresh Anandakrishnan, ETH Zurich
Marty McLaran, Google
Abdul Kader and Steve Scott, Microsoft

Evaluation: Used Machine & Objectives

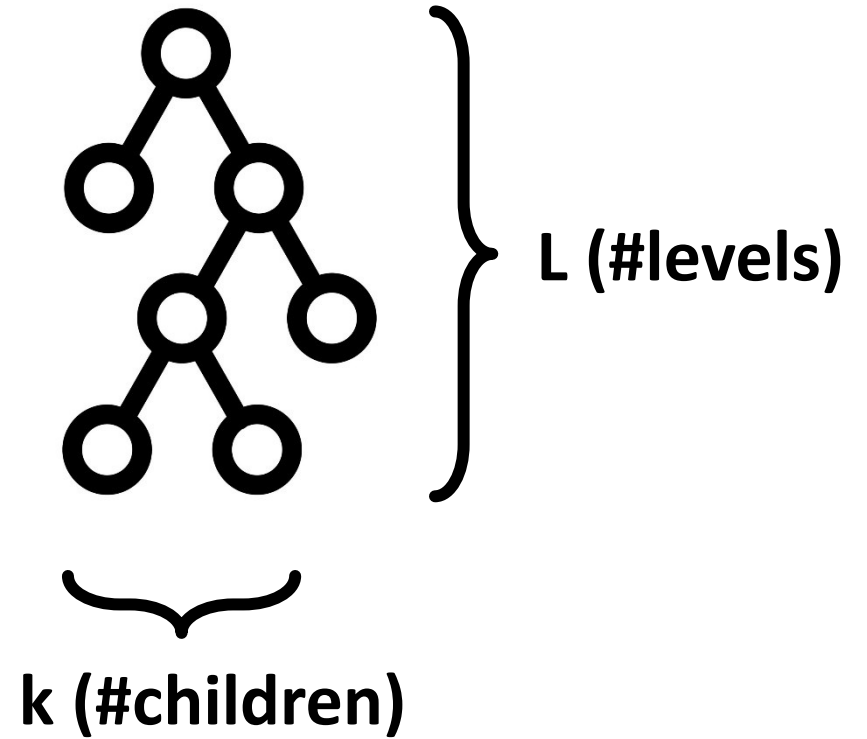
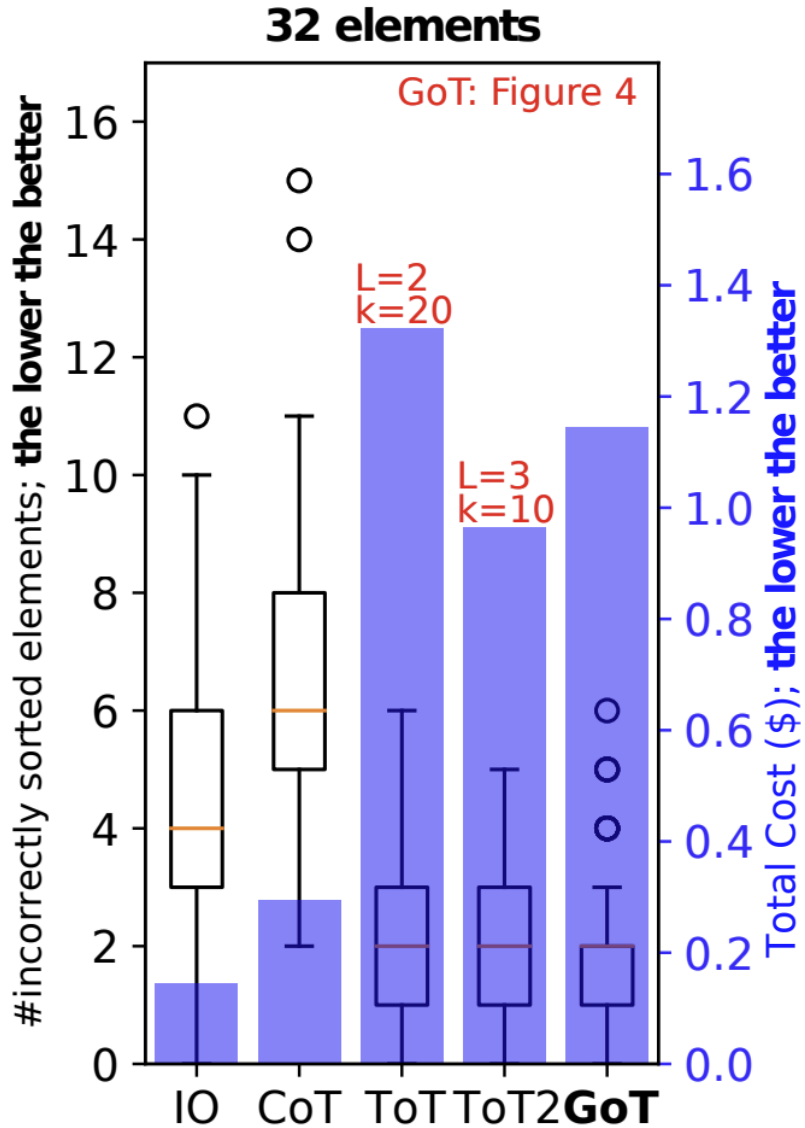
CSCS Cray Piz Daint & Ault
64GB – 2TB memory per server

Main goal: show that GoT successfully harnesses the graph abstraction to enable more efficient queries



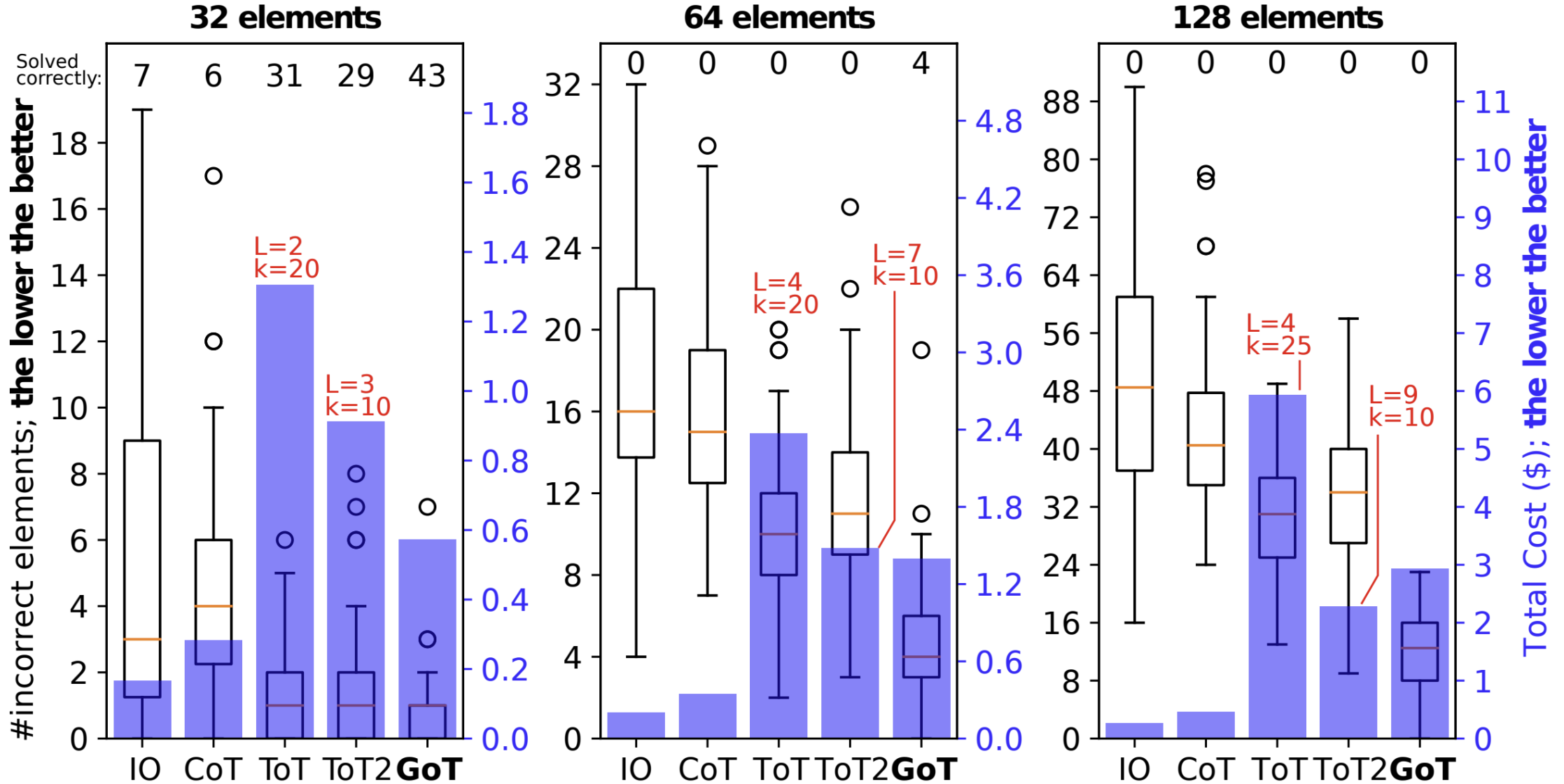
Sorting Numbers

The longer the sequence, the higher the gain

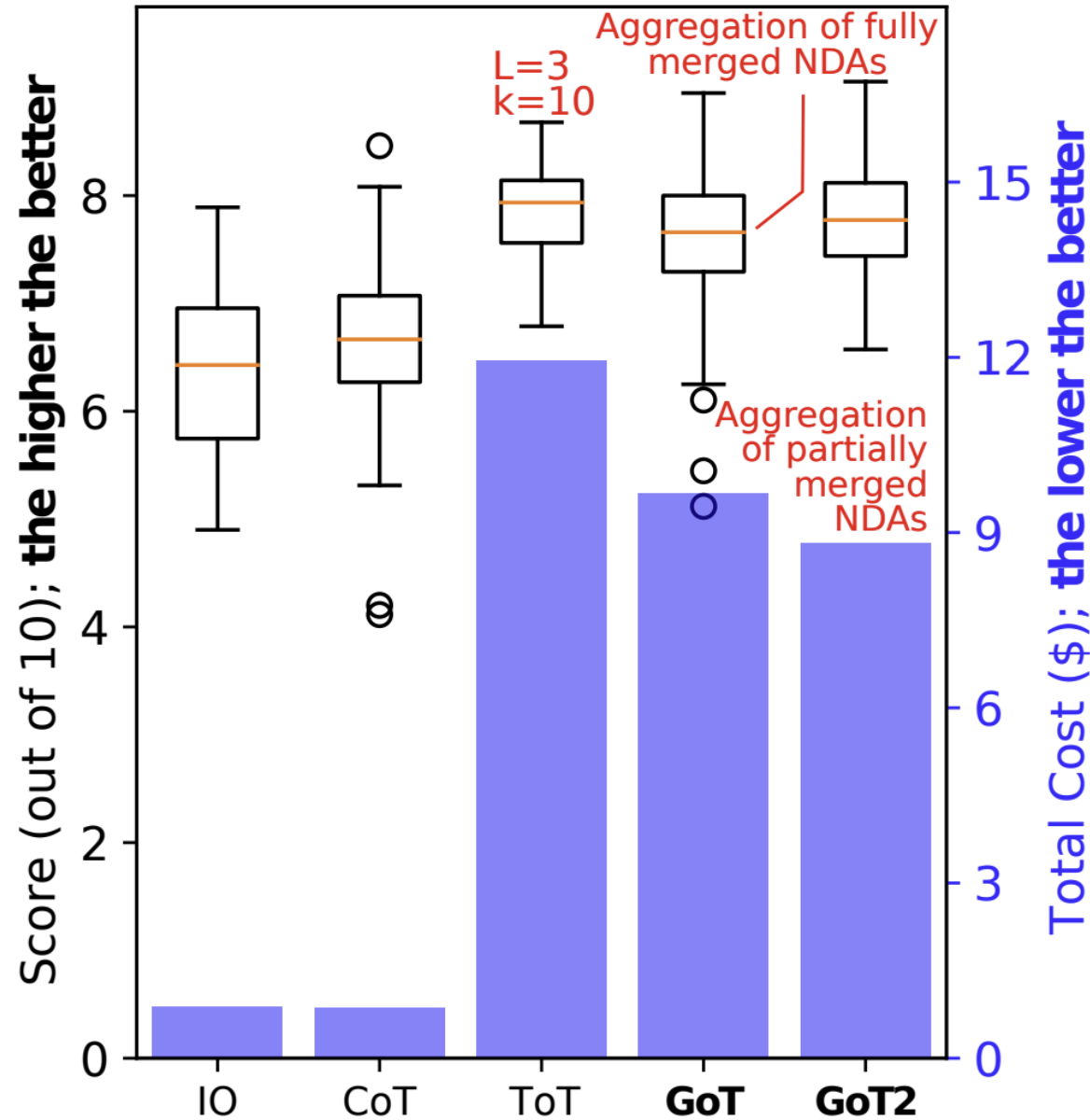


Intersecting Sets of Numbers

The longer the sequence, the higher the gain



Merging Documents



The Emergence of the „Generative AI Ecosystem”

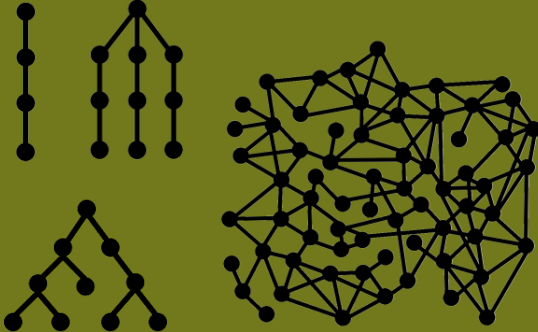
Training related




Why does structured prompting work?

Inference related

Prompting Structures



Tools



Psychology



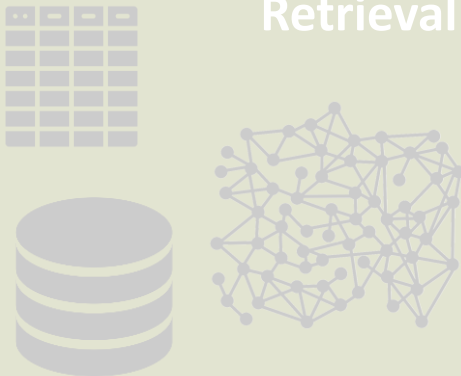
Replies („thoughts”)
Prompts



Web



Retrieval



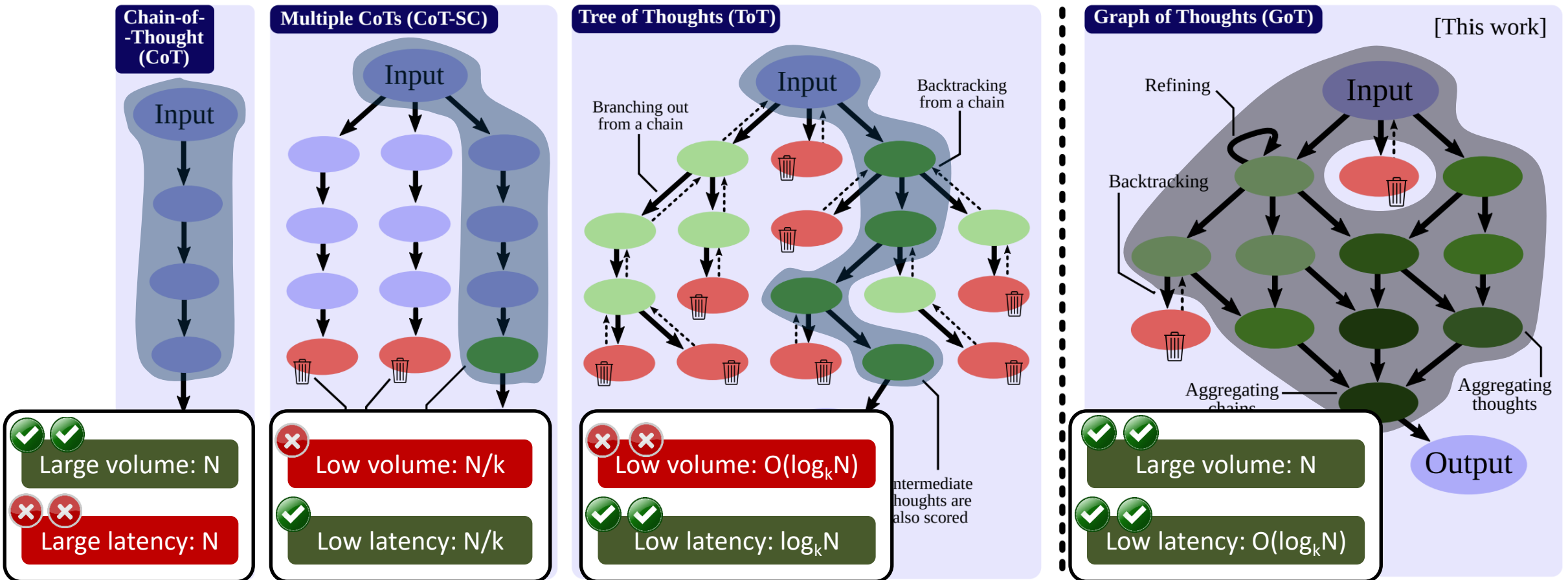
Why Does Structured Prompting Work?

Assume a **fixed thought size** (#tokens) and a **fixed context size** (#thoughts in the LLM context, denoted with N)

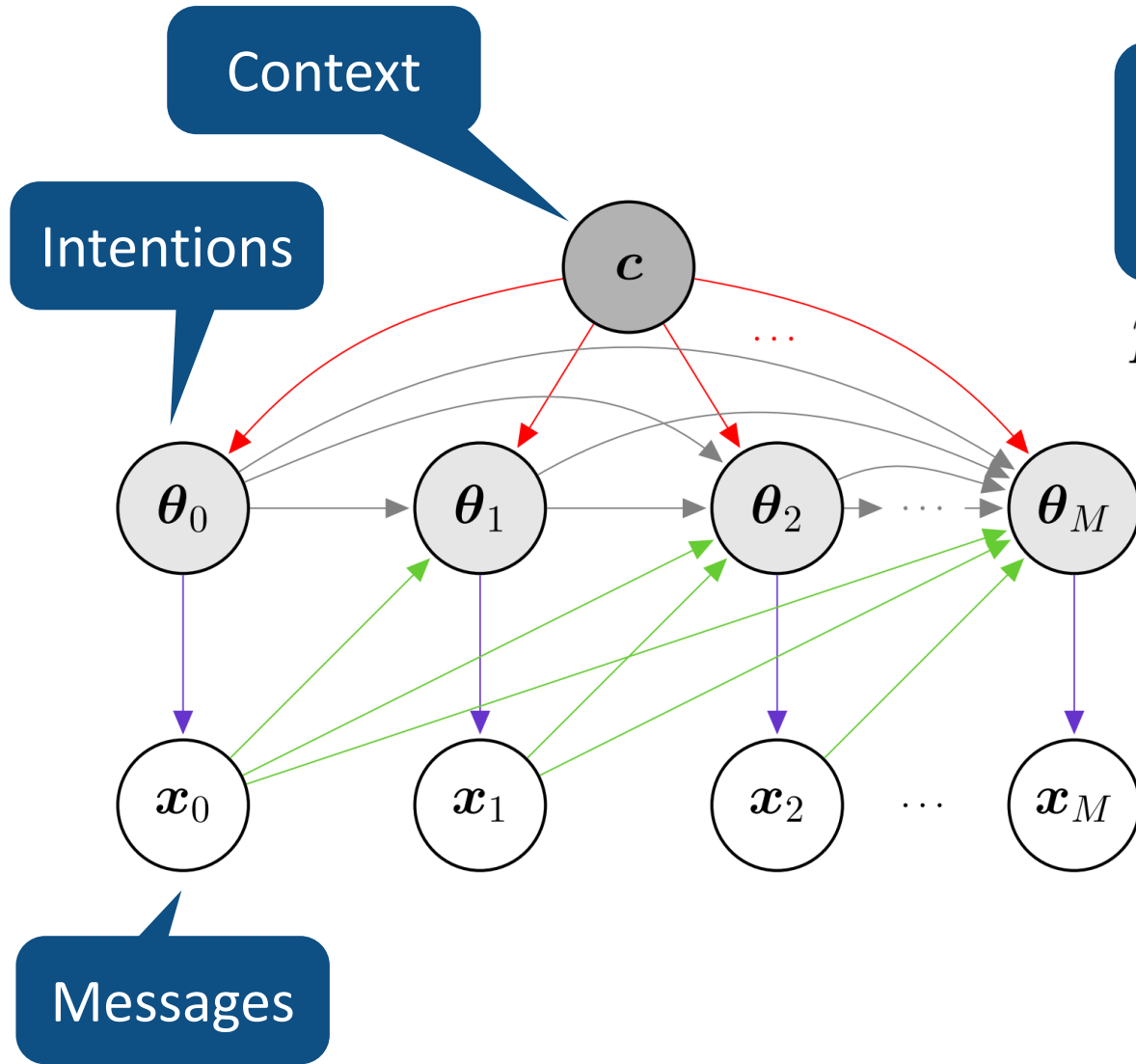
Volume – for a given thought t – is the number of preceding LLM thoughts that could have impacted t

Why Does Structured Prompting Work?

Assume a **fixed thought size** (#tokens) and a **fixed context size** (#thoughts in the LLM context, denoted with N)



Why Does Structured Prompting Work? Probabilistic Graphical Models [1]



Likelihood of generating a chain of thoughts, basing on a pre-prompt with input I and with N CoT in-context examples, but without access to the true context.

$$p_{LLM} \equiv p_{LLM}(\text{CoT} | I, \text{CoT-Examples}(N))$$

Likelihood of generating the same chain of thoughts as above, but using the true language (and context) conditioned on the same input I

$$p_{\text{True}} \equiv p_{\text{True}}(\text{CoT} | I, \text{True-Context})$$

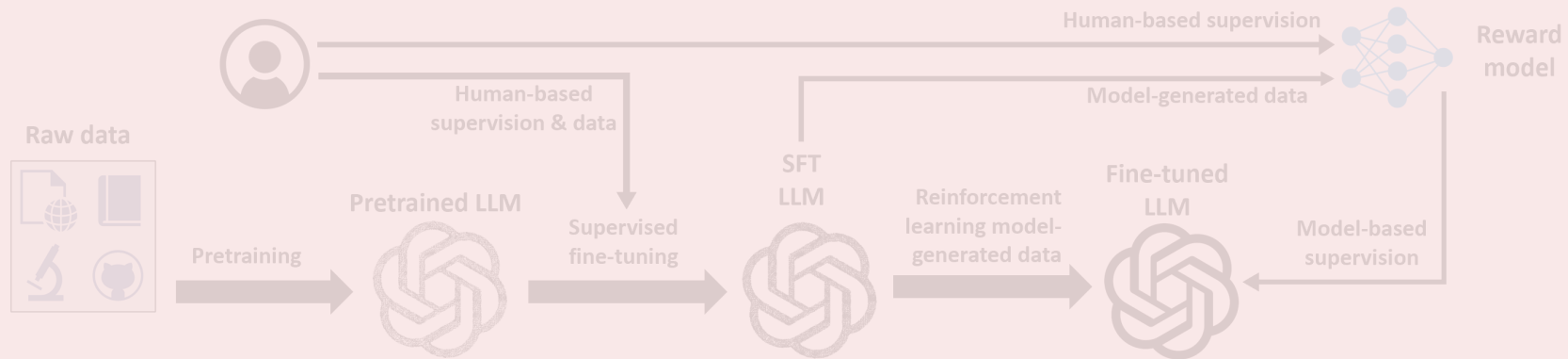
$$|p_{LLM} - p_{\text{True}}| \leq \rho^N$$

„A function of the language ambiguities“, < 1

[1] R. Tatunov et al. *Why Can Large Language Models Generate Correct Chain-of-Thoughts?* Arxiv, 30 October 2023.

The Emergence of the „Generative AI Ecosystem”

Training related



Inference related

Prompting Structures

Tools

Psychology

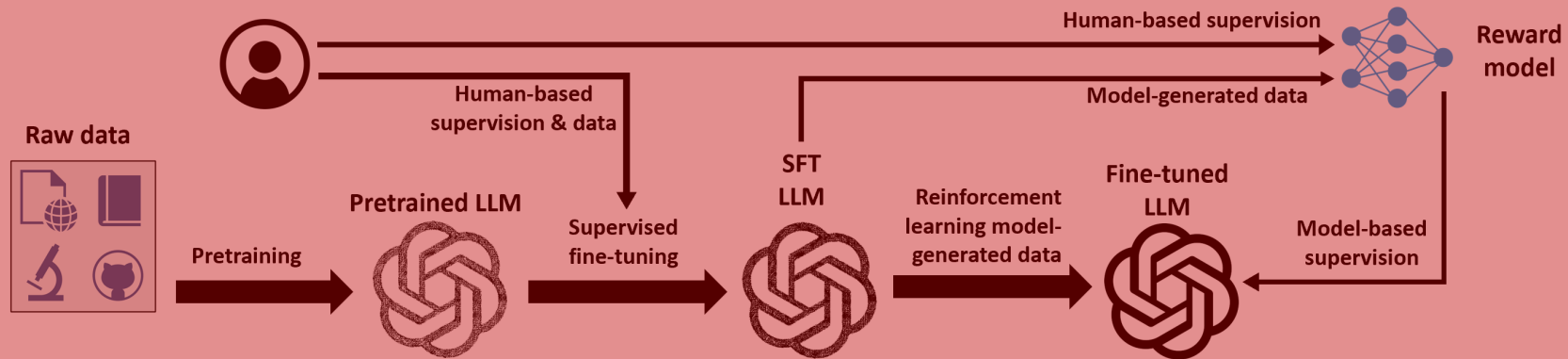
Web

Replies („thoughts”)
Prompts

Retrieval

The Emergence of the „Generative AI Ecosystem”: Training

Training related



Inference related

Prompting Structures

Tools

Psychology

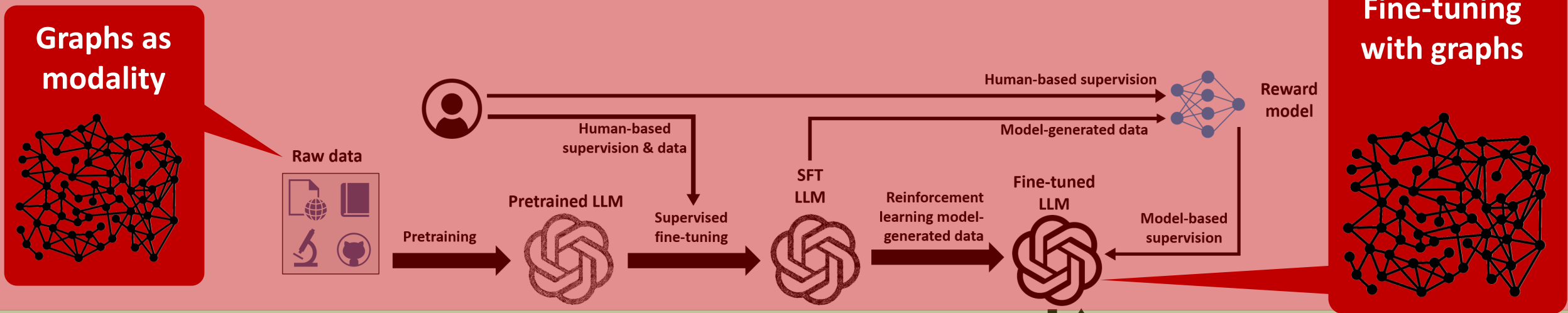
Replies („thoughts”) ↑ Prompts ↓

Web

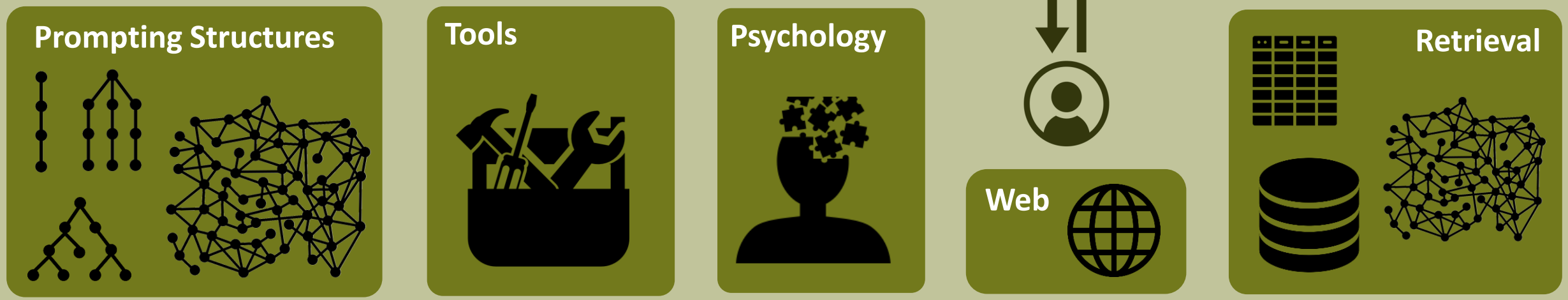
Retrieval

The Emergence of the „Generative AI Ecosystem”: Training

Training related

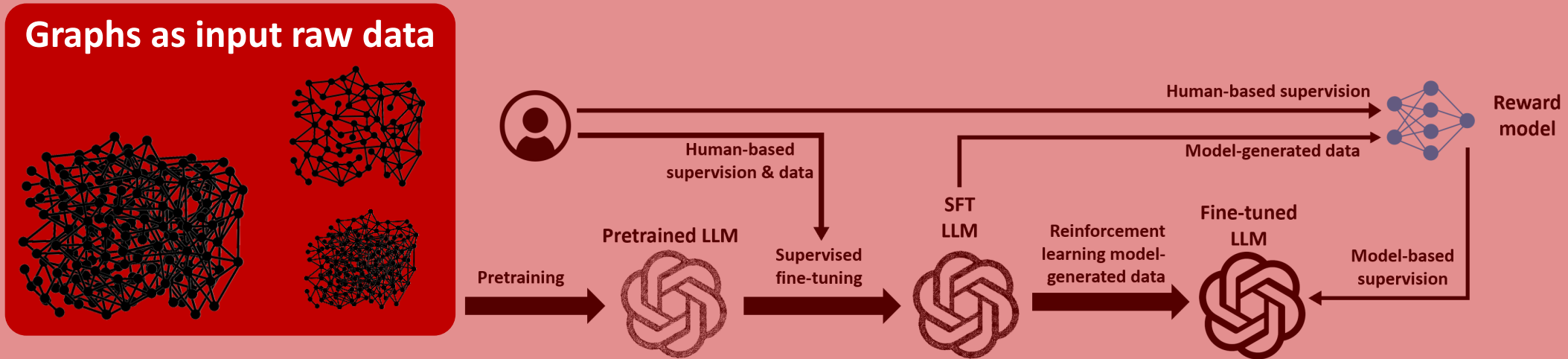


Inference related

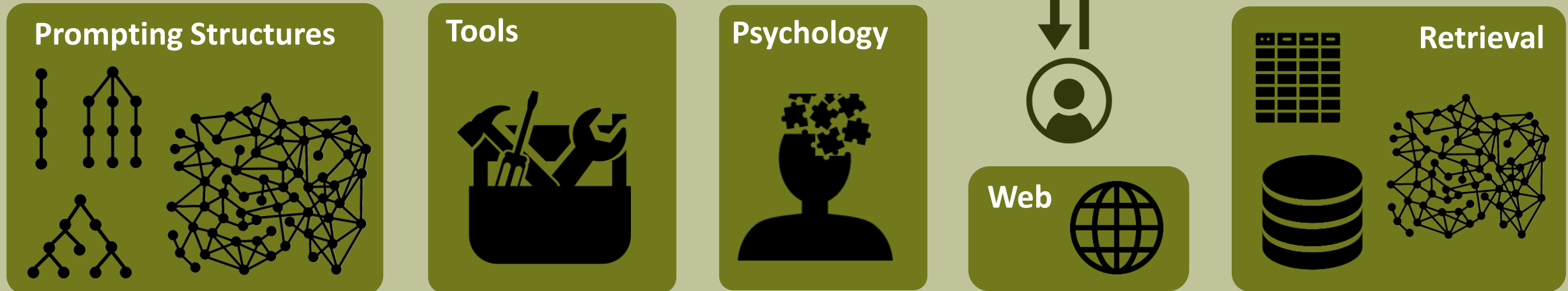


Graph Foundation Models & Graph Prompting

Training related



Inference related

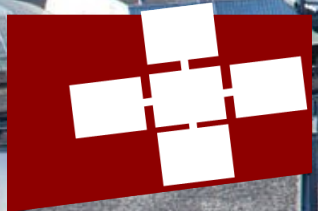


M. BESTA, T. HOEFLER

WITH N. BLACH, A. KUBICEK, R. GERSTENBERGER, AND MANY OTHERS

Graph of Thoughts: Solving Elaborate Problems with Large Language Models

Thank you



Prompting Example: Sorting



Hello. I want to sort the following input sequence of numbers: {input}

Prompting Example: Sorting

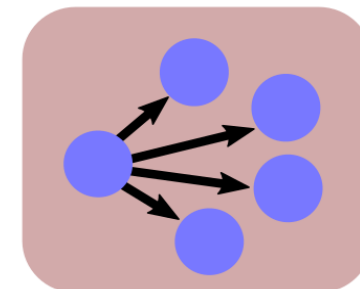
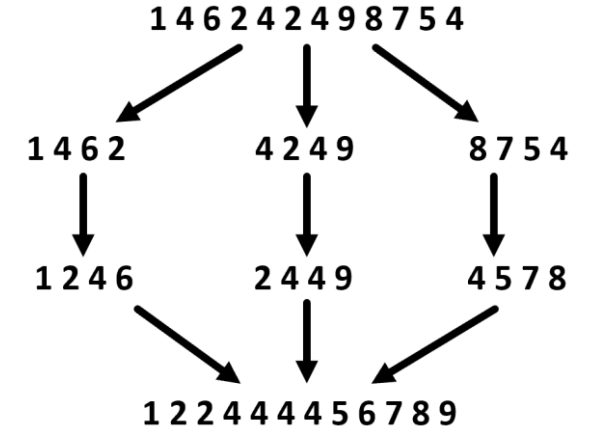


1

<Instruction> Split the following list of 64 numbers into 4 lists of 16 numbers each, the first list should contain the first 16 numbers, the second list the second 16 numbers, the third list the third 16 numbers and the fourth list the fourth 16 numbers. Only output the final 4 lists in the following format without any additional text or thoughts!

```

{
  "List 1": [3, 4, 3, 5, 7, 8, 1, ...],
  "List 2": [2, 9, 2, 4, 7, 1, 5, ...],
  "List 3": [6, 9, 8, 1, 9, 2, 4, ...],
  "List 4": [9, 0, 7, 6, 5, 6, 6, ...]
} </Instruction>
  
```

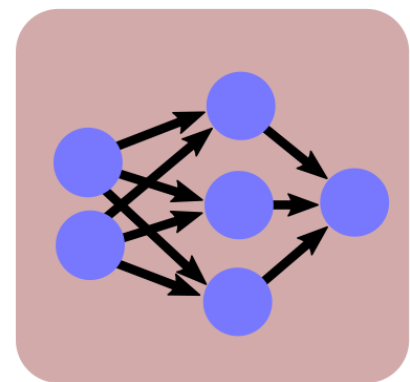
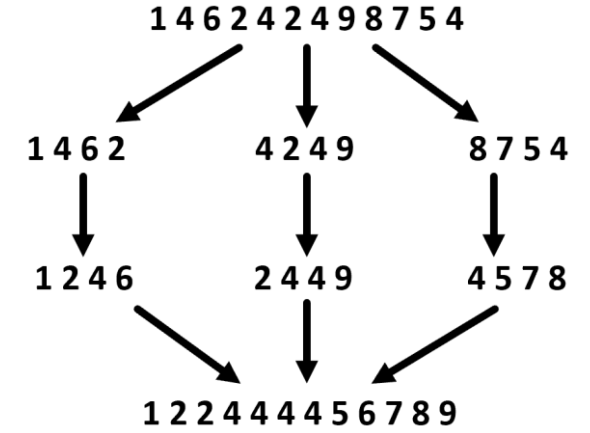


Prompting Example: Sorting

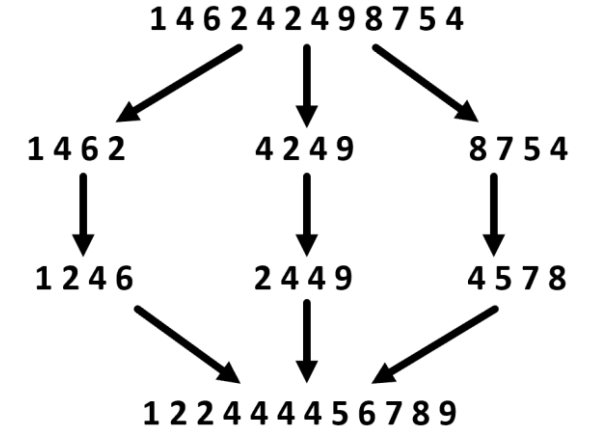
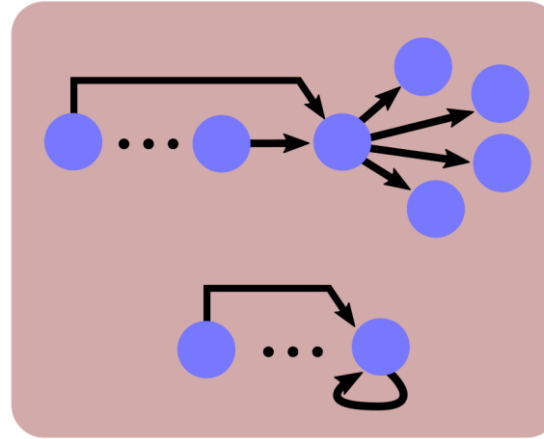
>_

2

*<Instruction> Merge the following 2 sorted lists of length {length1} each, into one sorted list of length {length2} using a merge sort style approach. Only output the final merged list without any additional text or thoughts!
</Instruction>*



Prompting Example: Sorting



>_

3

<Instruction> Sort the following list of numbers in ascending order. Output only the sorted list of numbers, no additional text. **</Instruction>**

<Example>

Input: [3, 7, 0, 2, 8, 1, 2, 2, 2, 4, 7, 8, 5, 5, 3, 9, 4, 3, 5, 6, 6, 4, 4, 5, 2, 0, 9, 3, 3, 9, 2, 1]

Output: [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 6, 6, 7, 7, 8, 8, 9, 9, 9]

</Example>

Input: {input}

The input thought t 