

CS 598

AI Efficiency: Systems and Algorithms

Minjia Zhang

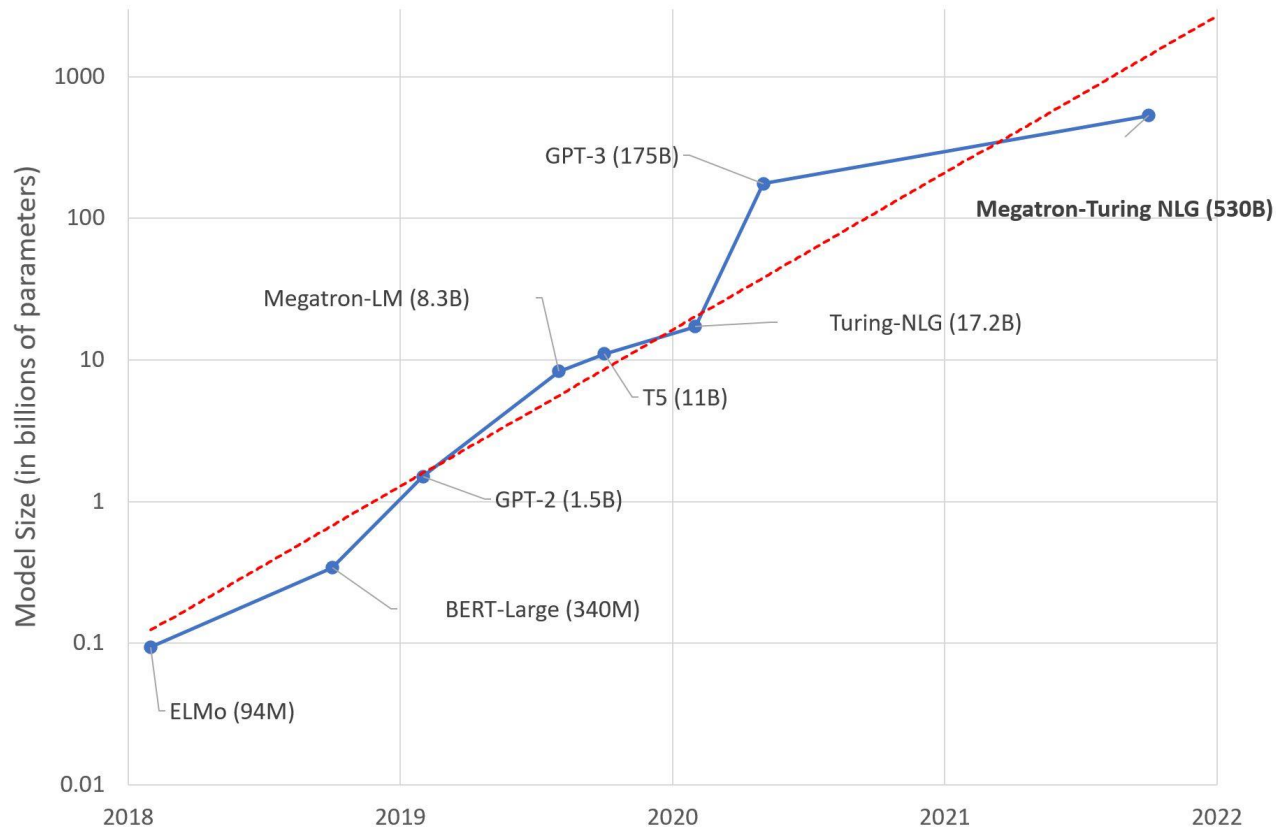
Computer Science Department
University of Illinois at Urbana-Champaign

The Large Language Model Revolution

What is Language Model?



Evolution of DNN Models



Larger models → better accuracy

Model size is still growing

Not reached the accuracy limit yet

More compute-efficient to train larger models than smaller ones to same accuracy

Code Continuation and Generation

```
"""
Python 3
Get the current value of a Bitcoin in US dollars using the bitcoincharts api
"""

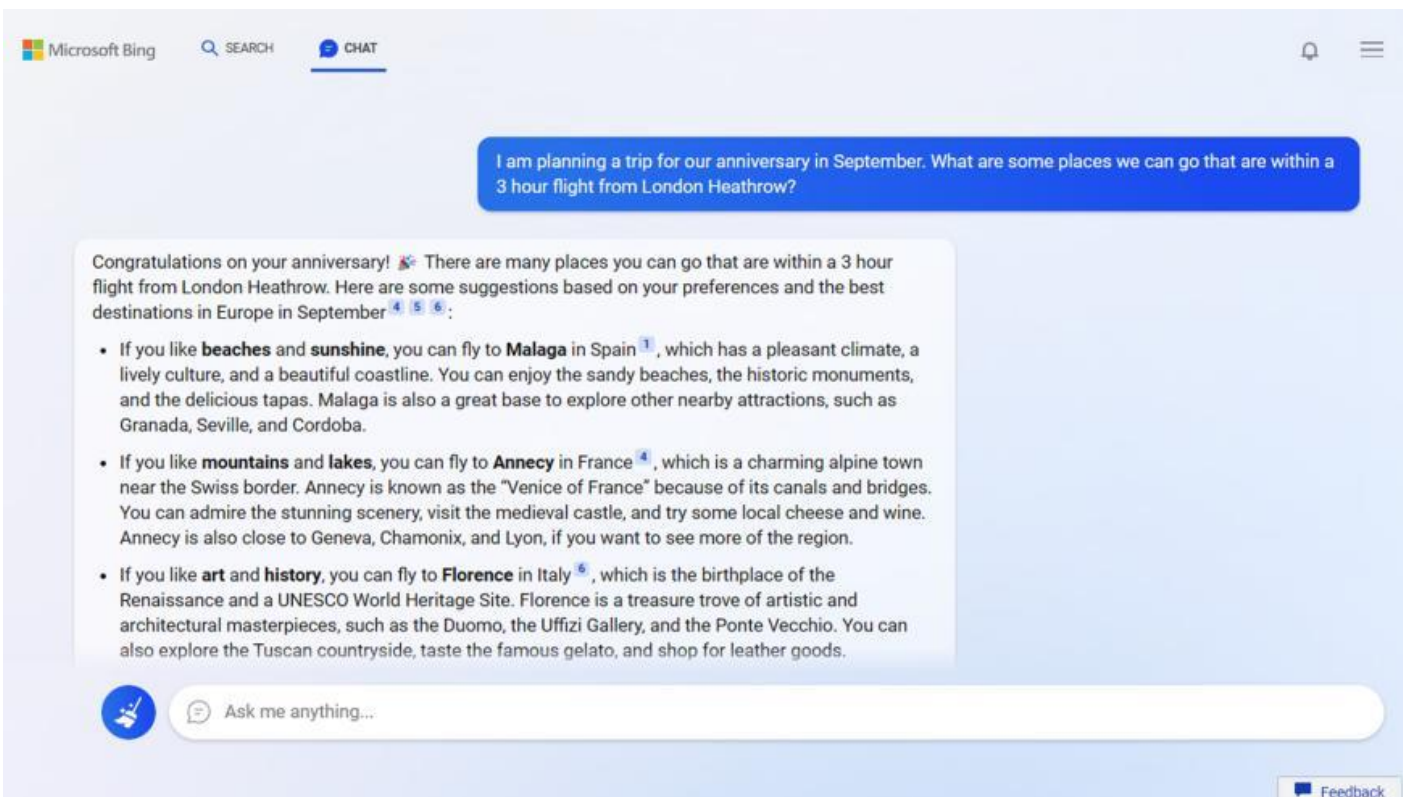
import requests
import json

def get_bitcoin_price():
    url = 'http://api.bitcoincharts.com/v1/weighted_prices.json'
    response = requests.get(url)
    data = json.loads(response.text)
    return data['USD']['7d']

if __name__ == '__main__':
    print(get_bitcoin_price())
```

[Suggest code and entire function in your editor – Github/OpenAI Codex](#)

Dialogue/New Search



Microsoft Bing SEARCH CHAT

I am planning a trip for our anniversary in September. What are some places we can go that are within a 3 hour flight from London Heathrow?

Congratulations on your anniversary! 🎉 There are many places you can go that are within a 3 hour flight from London Heathrow. Here are some suggestions based on your preferences and the best destinations in Europe in September⁴ ⁵ ⁶:

- If you like **beaches** and **sunshine**, you can fly to **Malaga** in Spain¹, which has a pleasant climate, a lively culture, and a beautiful coastline. You can enjoy the sandy beaches, the historic monuments, and the delicious tapas. Malaga is also a great base to explore other nearby attractions, such as Granada, Seville, and Cordoba.
- If you like **mountains** and **lakes**, you can fly to **Annecy** in France⁴, which is a charming alpine town near the Swiss border. Annecy is known as the "Venice of France" because of its canals and bridges. You can admire the stunning scenery, visit the medieval castle, and try some local cheese and wine. Annecy is also close to Geneva, Chamonix, and Lyon, if you want to see more of the region.
- If you like **art** and **history**, you can fly to **Florence** in Italy⁶, which is the birthplace of the Renaissance and a UNESCO World Heritage Site. Florence is a treasure trove of artistic and architectural masterpieces, such as the Duomo, the Uffizi Gallery, and the Ponte Vecchio. You can also explore the Tuscan countryside, taste the famous gelato, and shop for leather goods.

Ask me anything...

Feedback

GPTs are GPTs: An Early Look at the Labor Market Impact Potential of Large Language Models

Tyna Eloundou¹, Sam Manning^{1,2}, Pamela Mishkin*¹, and Daniel Rock³

¹OpenAI

²OpenResearch

³University of Pennsylvania

March 21, 2023

Abstract

We investigate the potential implications of Generative Pre-trained Transformer (GPT) models and related technologies on the U.S. labor market. Using a new rubric, we assess occupations based on their correspondence with GPT capabilities, incorporating both human expertise and classifications from GPT-4. Our findings indicate that approximately 80% of the U.S. workforce could have at least 10% of their work tasks affected by the introduction of GPTs, while around 19% of workers may see at least 50% of their tasks impacted. The influence spans all wage levels, with higher-income jobs potentially facing greater exposure. Notably, the impact is not limited to industries with higher recent productivity growth. We conclude that Generative Pre-trained Transformers exhibit characteristics of general-purpose technologies (GPTs), suggesting that as these models could have notable economic, social, and policy implications.

Image Generation from Text

TEXT DESCRIPTION

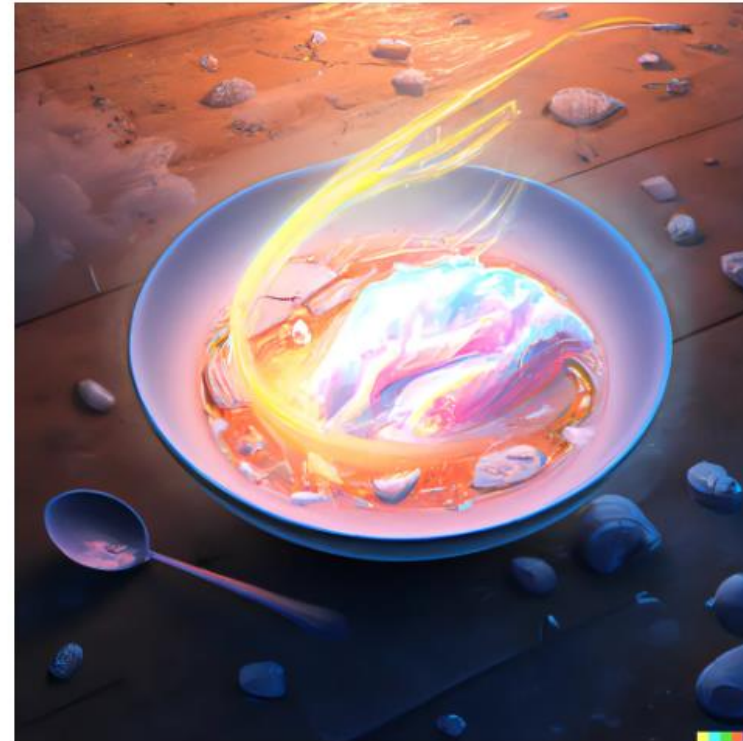
An astronaut Teddy bears A bowl of soup

that is a portal to another dimension that looks like a monster as a planet in the universe

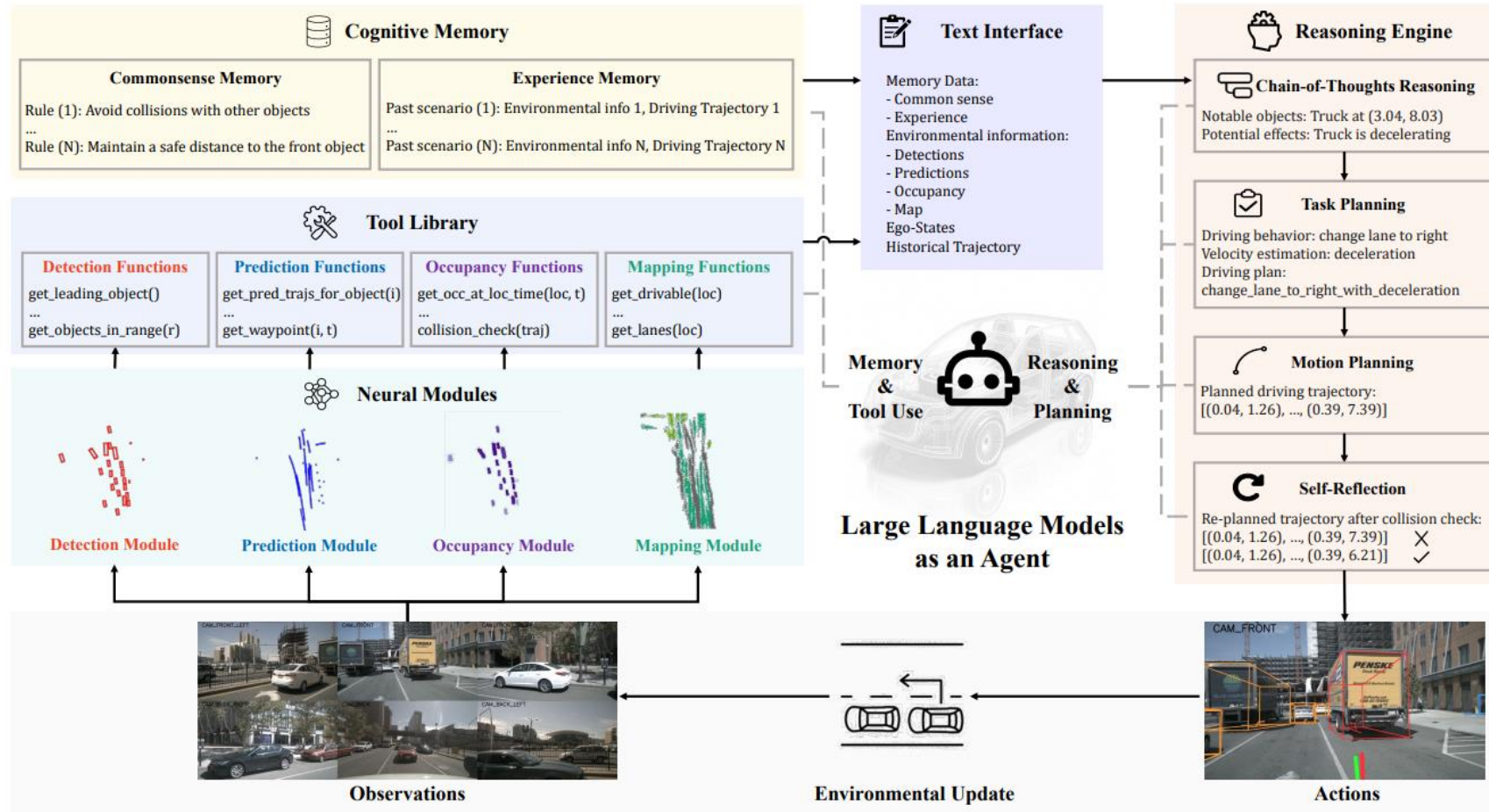
as digital art in the style of Basquiat drawn on a cave wall



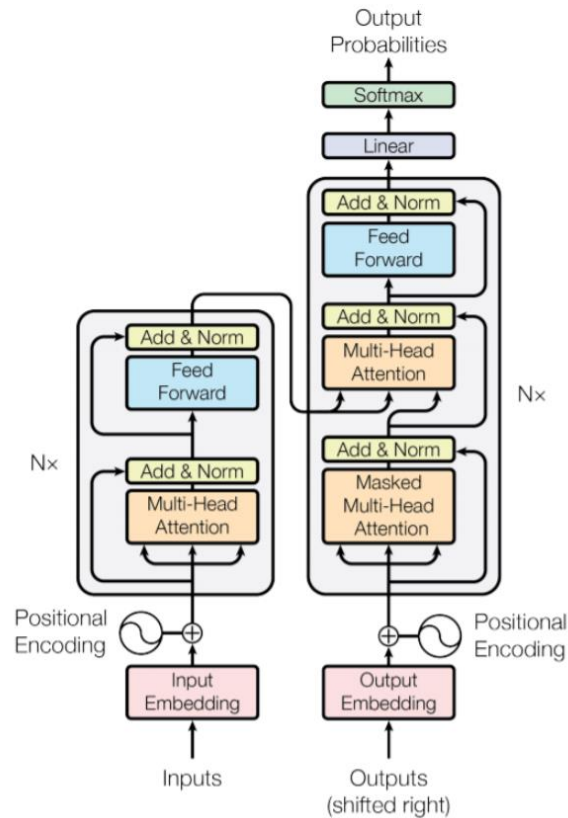
DALL·E 2



Autonomous Driving



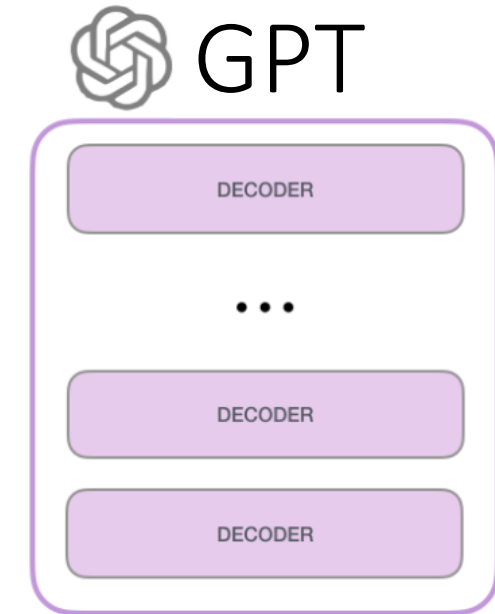
Transformers for Language Modeling



[1]



[2]



[3]

[1] Vaswani et al. "Attention Is All You Need", <https://arxiv.org/abs/1706.03762>, 2018

[2] Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", 2019, <https://arxiv.org/abs/1810.04805>

[3] Brown et al. "Language Models are Few-Shot Learners", 2020, <https://arxiv.org/abs/2005.14165>

AI Efficiency Challenges

AI Efficiency Challenges

- Too slow to train high-quality models on massive data
 - More hardware \neq higher throughput, bigger model
 - Higher throughput \neq better accuracy, faster convergence, lower cost
 - Better techniques \neq handy to use
- Slow and expensive to deploy the models

DL System Desired Capabilities (3E)

Efficiency: Efficient use of hardware for high scalability and throughput

Effectiveness: High accuracy and fast convergence, lowering cost

Easy to use: Improve development productivity of model scientists

Research Focus (3Es)

Efficiency

Efficient use of hardware for low latency, high scalability and throughput

☐ Training efficiency

- High performance and cost-efficient training [ATC'21]

(Training **10x larger** model with **same GPUs**, adopted by major DL frameworks)

- Graph neural networks [ASPLOS'23]

- Training w. spot instances [NSDI'23, SOSP'24]

☐ Inference efficiency

- Recurrent neural networks [ATC'18]

(**10x faster** latency and **hundreds of millions of dollar saving**, Microsoft 2018 **top-3 “Cool Tech”** showcase)

- Transformers [SC'22]

Research Focus (3Es)

Efficiency

Efficient use of hardware for low latency, high scalability and throughput

☐ Training efficiency

- High performance and cost-efficient training [ATC'21]
(Training **10x larger** model with **same GPUs**, adopted by major DL frameworks)
- Graph neural networks [ASPLOS'23]
- Training w. spot instances [NSDI'23, SOSP'24]

☐ Inference efficiency

- Recurrent neural networks [ATC'18]
(**10x faster** latency and **hundreds of millions of dollar saving**, Microsoft 2018 **top-3 "Cool Tech"** showcase)
- Transformers [SC'22]

Effectiveness

High accuracy and fast convergence, lowering cost

☐ Model compression

- Extreme compression [NeurIPS'22 **Oral**]
(**50x smaller** model size, similar accuracy)
- Zero-cost quant. [NeurIPS'22 **spotlight**]
- 1-bit communication [ICLR'23]

☐ Data efficiency

- Curriculum learning [NeurIPS'22, **spotlight**]
(Retaining 99% accuracy with **10x less data**)
- Adversarial learning [AAAI'22]

☐ Efficient architecture

- Mixture-of-Experts [ICML'22]

Research Focus (3Es)

Efficiency

Efficient use of hardware for low latency, high scalability and throughput

☐ Training efficiency

- High performance and cost-efficient training [ATC'21]
(Training **10x larger** model with **same GPUs**, adopted by major DL frameworks)
- Graph neural networks [ASPLOS'23]
- Training w. spot instances [NSDI'23, SOSP'24]

☐ Inference efficiency

- Recurrent neural networks [ATC'18]
(**10x faster** latency and **hundreds of millions of dollar saving**, Microsoft 2018 top-3 “Cool Tech” showcase)
- Transformers [SC'22]

Effectiveness

High accuracy and fast convergence, lowering cost

☐ Model compression

- Extreme compression [NeurIPS'22 Oral]
(**50x smaller** model size, similar accuracy)
- Zero-cost quant. [NeurIPS'22 spotlight]
- 1-bit communication [ICLR'23]

☐ Data efficiency

- Curriculum learning [NeurIPS'22, spotlight]
(Retaining 99% accuracy with **10x less data**)
- Adversarial learning [AAAI'22]

☐ Efficient architecture

- Mixture-of-Experts [ICML'22]

Easy-to-Use

Improve development productivity of model scientists

☐ DL Compilation

- Hardware heterogeneity [IPDPS'21]

☐ Auto-Tuner

- Adaptive tuning [NeurIPS'20]
(**3.9x faster** optimization speed)
- Multi-task tuning [ICLR'21]

Research Focus (3Es)

Efficiency

Efficient use of hardware for low latency, high scalability and throughput

☐ Training efficiency

- High performance and cost-efficient training [ATC'21]
(Training **10x larger** model with **same GPUs**, adopted by major DL frameworks)
- Graph neural networks [ASPLOS'23]
- Training w. spot instances [NSDI'23, SOSP'24]

☐ Inference efficiency

- Recurrent neural networks [ATC'18]
(**10x faster** latency and **hundreds of millions of dollar saving**, Microsoft 2018 top-3 “Cool Tech” showcase)
- Transformers [SC'22]

Effectiveness

High accuracy and fast convergence, lowering cost

☐ Model compression

- Extreme compression [NeurIPS'22 Oral]
(**50x smaller** model size, similar accuracy)
- Zero-cost quant. [NeurIPS'22 spotlight]
- 1-bit communication [ICLR'23]

☐ Data efficiency

- Curriculum learning [NeurIPS'22, spotlight]
(Retaining 99% accuracy with **10x less data**)
- Adversarial learning [AAAI'22]

☐ Efficient architecture

- Mixture-of-Experts [ICML'22]

Easy-to-Use

Improve development productivity of model scientists

☐ DL Compilation

- Hardware heterogeneity [IPDPS'21]

☐ Auto-Tuner

- Adaptive tuning [NeurIPS'20]
(**3.9x faster** optimization speed)
- Multi-task tuning [ICLR'21]

Industry products:  Bing, Ads, Azure, Office



DeepSpeed



Open-source software

Training Efficiency: Breaking the Memory Wall

ML/DL Training Problem Definition Recap

- Given model f , data set $\{x_i, y_i\}_{i=1}^N$
- Minimize the loss between predicted labels and true labels:

$$\text{Min } \frac{1}{N} \sum_{i=1}^N \text{loss}(f(x_i, y_i))$$

- Common loss function
 - Cross-entropy, MSE (mean squared error)
- Common way to solve the minimization problem
 - Stochastic gradient descent (SGD)
 - Adaptive learning rates optimizers (e.g., Adam)

Gradient Descent

- Model f_w is parameterized by weight w
- $\eta > 0$ is the learning rate

For $t = 1$ to T

Backward pass Forward pass

$\Delta w = \eta \times \frac{1}{N} \sum_{i=1}^N \nabla \left(\text{loss}(f_w(x_i, y_i)) \right)$ // compute derivative and update

$w -= \Delta w$ // apply update

End

Adaptive Learning Rates (Adam)

- Model f_w is parameterized by weight w
- $\eta > 0$ is the learning rate

For $t = 1$ to T

$$\Delta w = \eta \times \frac{1}{N} \sum_{i=1}^N \nabla \left(\text{loss}(f_w(x_i, y_i)) \right)$$

$w \leftarrow \Delta w$ // apply update

End

$$\nu_t = \beta_1 * \nu_{t-1} + (1 - \beta_1) * g_t$$

$$s_t = \beta_2 * s_{t-1} + (1 - \beta_2) * g_t^2$$

$$\Delta \omega_t = -\eta \frac{\nu_t}{\sqrt{s_t + \epsilon}} * g_t$$

g_t : Gradient at time t along ω^j

ν_t : Exponential Average of gradients along ω_j

s_t : Exponential Average of squares of gradients along ω_j

β_1, β_2 : Hyperparameters

Parallel/Distributed Gradient Descent

- Model f_w is parameterized by weight w
- $\eta > 0$ is the learning rate

For $t = 1$ to T

$$\Delta w = \eta \times \frac{1}{N} \sum_{i=1}^N \nabla \left(\text{loss}(f_w(x_i, y_i)) \right) \quad // \text{ compute derivative and update}$$

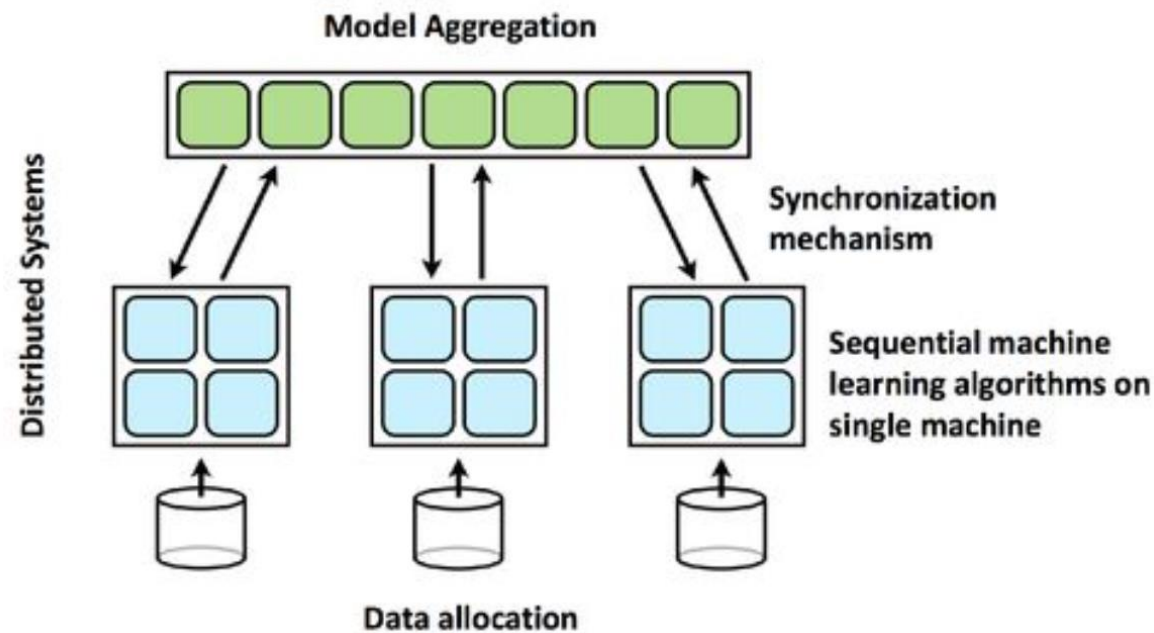
$w -= \Delta w$ // apply update

End

Can we parallelize it?



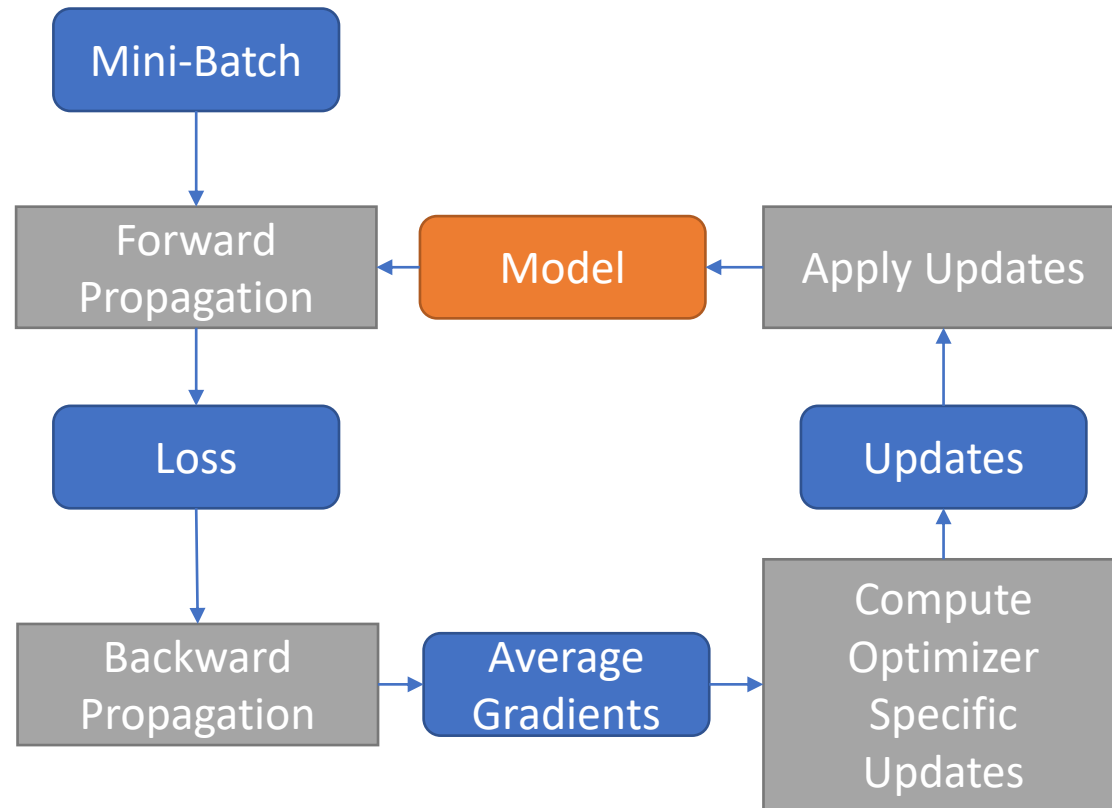
Data Parallelism (DP)



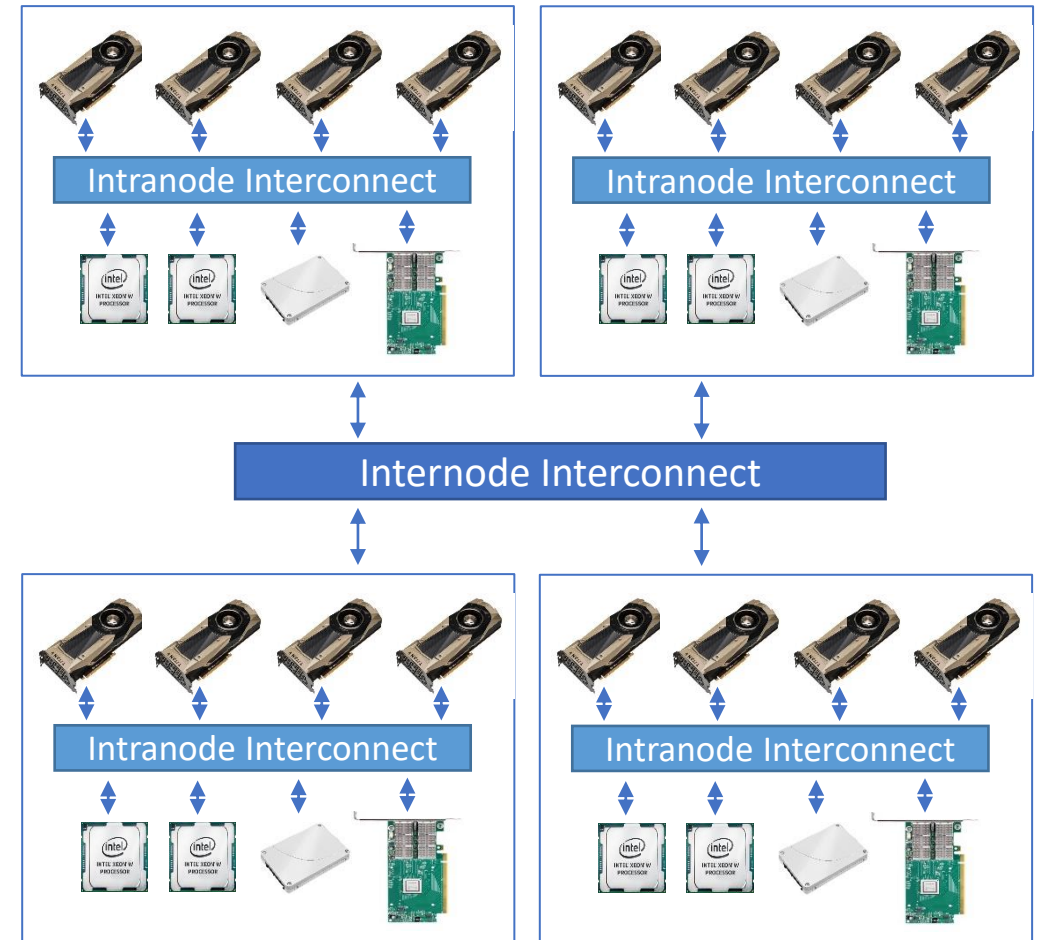
1. Partition the training data
2. Parallel training on different machines
3. Synchronize the local updates
4. Refresh local model with new parameters, then go to 2

Implemented as standard component in DL training frameworks, such as PyTorch DDP

Distributed Data Parallel Training in GPU Clusters



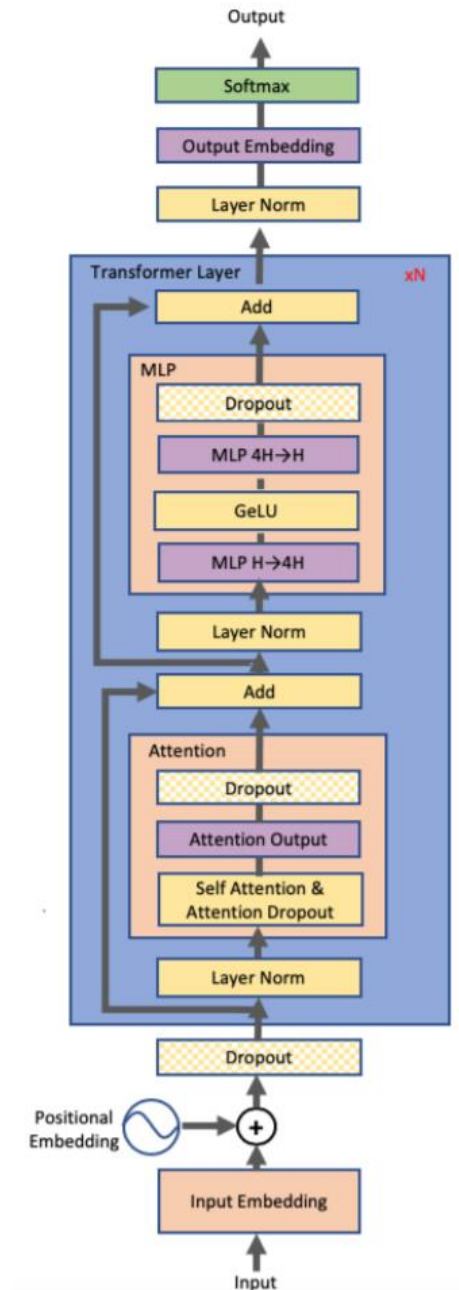
Data Parallel Training Loop



Distributed GPU Cluster

Large Model Training Challenges

	Bert-Large	GPT-2	Turing 17.2 NLG	GPT-3
Parameters	0.32B	1.5B	17.2B	175B
Layers	24	48	78	96
Hidden Dimension	1024	1600	4256	12288
Relative Computation	1x	4.7x	54x	547x
Memory Footprint	5.12GB	24GB	275GB	2800GB

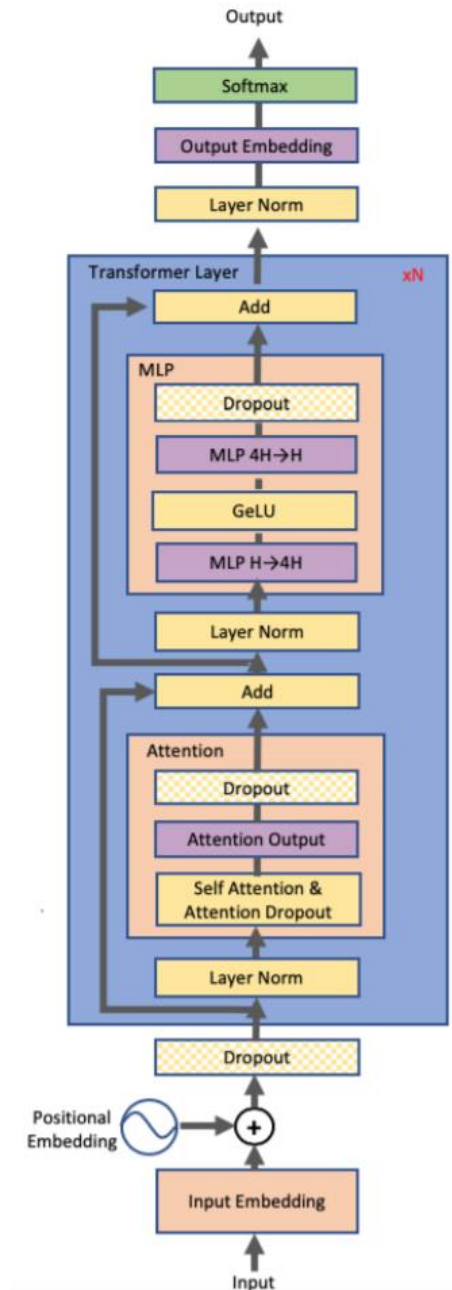


Large Model Training Challenges

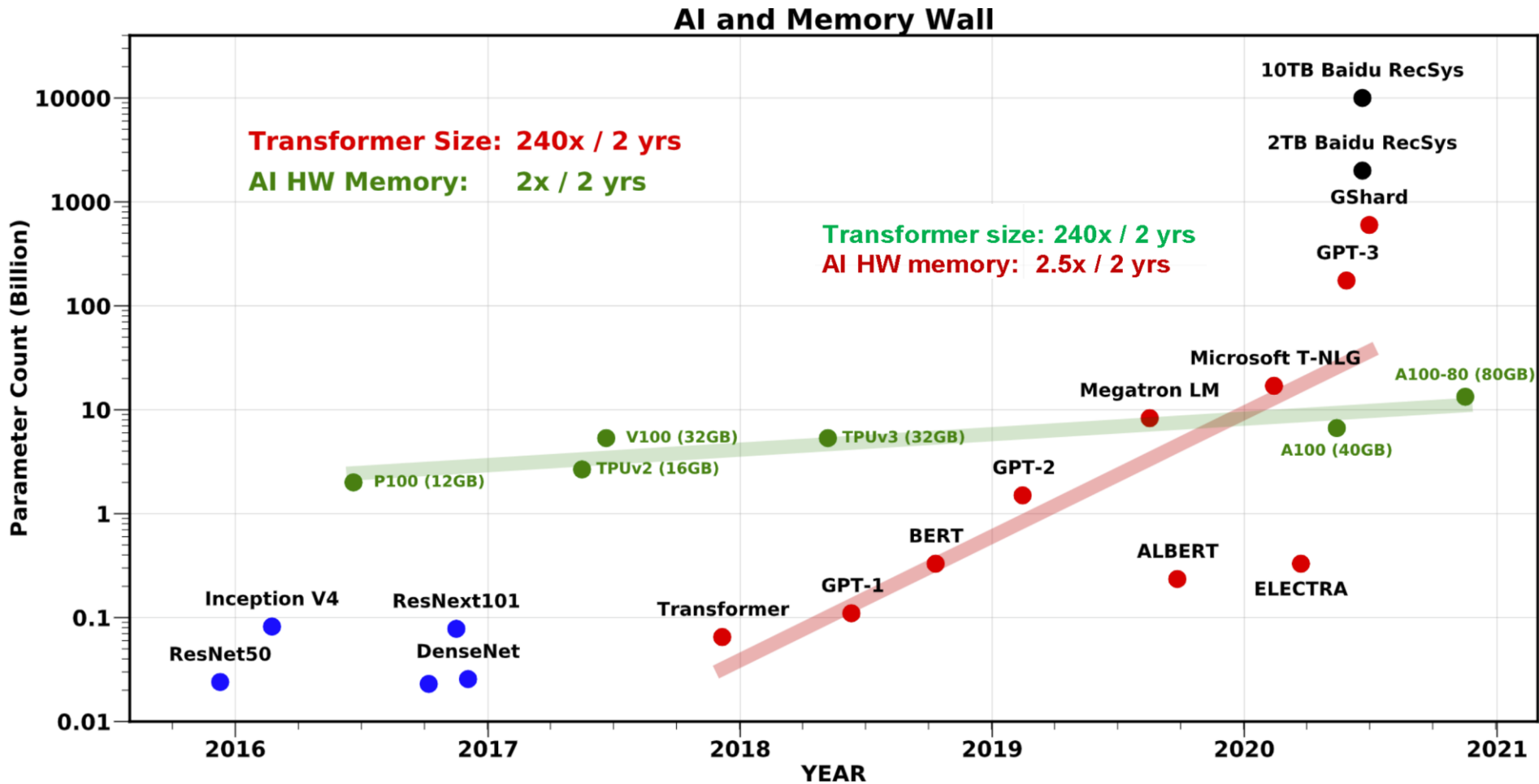
	Bert-Large	GPT-2	Turing 17.2 NLG	GPT-3
Parameters	0.32B	1.5B	17.2B	175B
Layers	24	48	78	96
Hidden Dimension	1024	1600	4256	12288
Relative Computation	1x	4.7x	54x	547x
Memory Footprint	5.12GB	24GB	275GB	2800GB

NVIDIA V100 GPU memory capacity: 16G/32G
 NVIDIA A100 GPU memory capacity: 40G/80G

Out of Memory



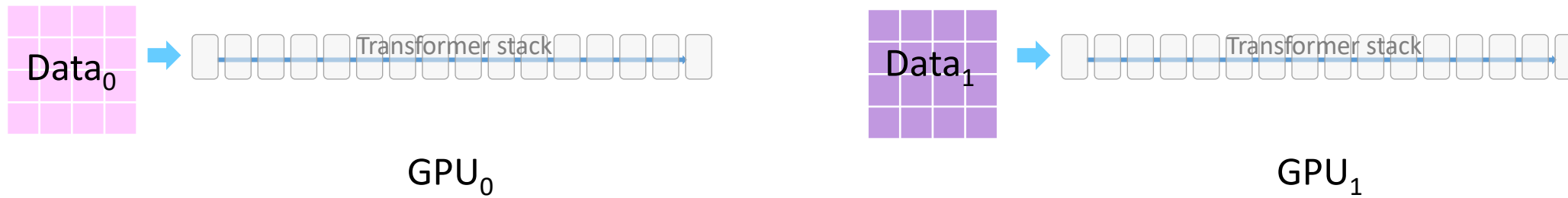
DNN Training Hit the Memory Wall



*AI and Memory Wall. (This blogpost has been written in... | by Amir Gholami | riselab | Medium

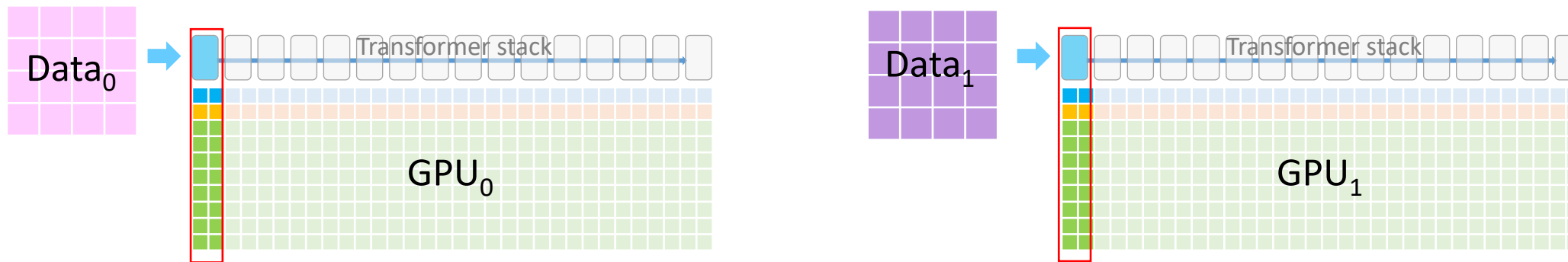
How to Break the Memory Wall?


Understanding Memory Consumption



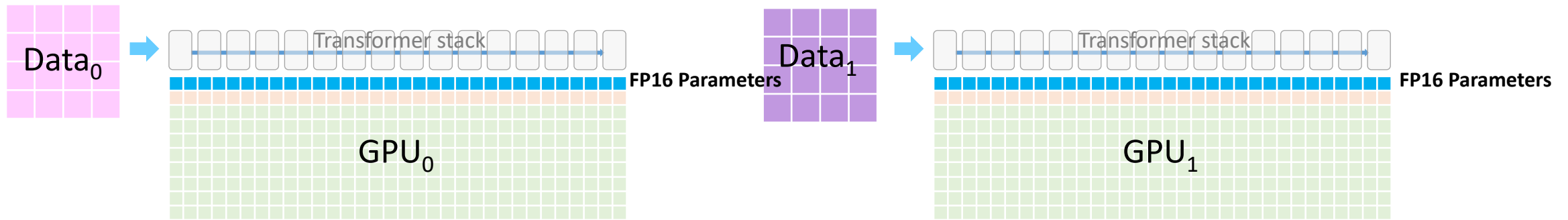
A 16-layer transformer model  = 1 layer

Understanding Memory Consumption



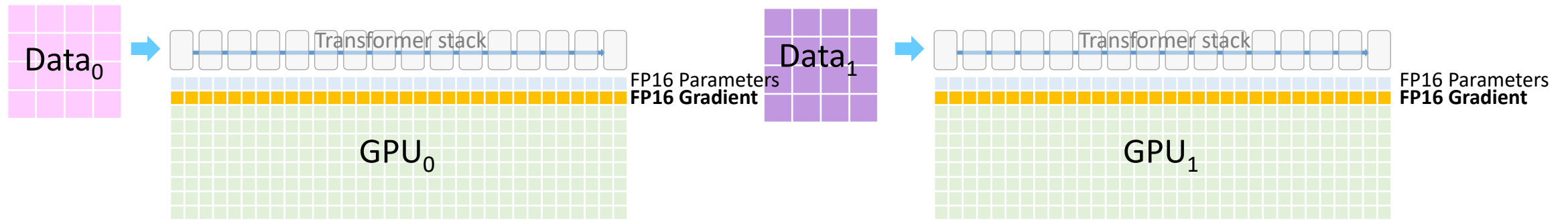
Each cell   represents GPU memory used by its corresponding transformer layer 

Understanding Memory Consumption



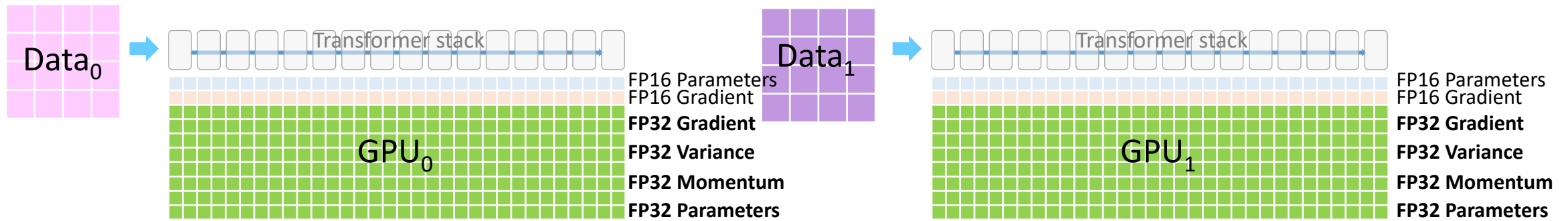
- FP16 parameter

Understanding Memory Consumption



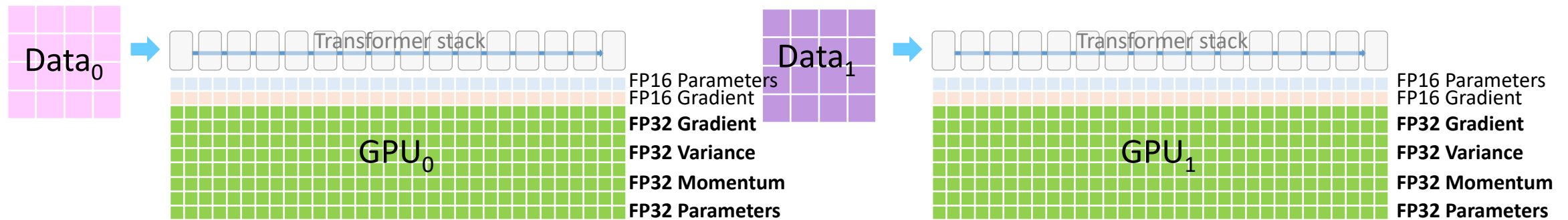
- FP16 parameter
- FP16 Gradients

Understanding Memory Consumption



- FP16 parameter
- FP16 Gradients
- FP32 Optimizer States
 - Gradients, Variance, Momentum, Parameters

Understanding Memory Consumption



- FP16 parameter : **2M bytes**
- FP16 Gradients : **2M bytes**
- FP32 Optimizer States : **16M bytes**
 - Gradients, Variance, Momentum, Parameters

Example 1B parameter model -> 20GB/GPU

Memory consumption doesn't include:

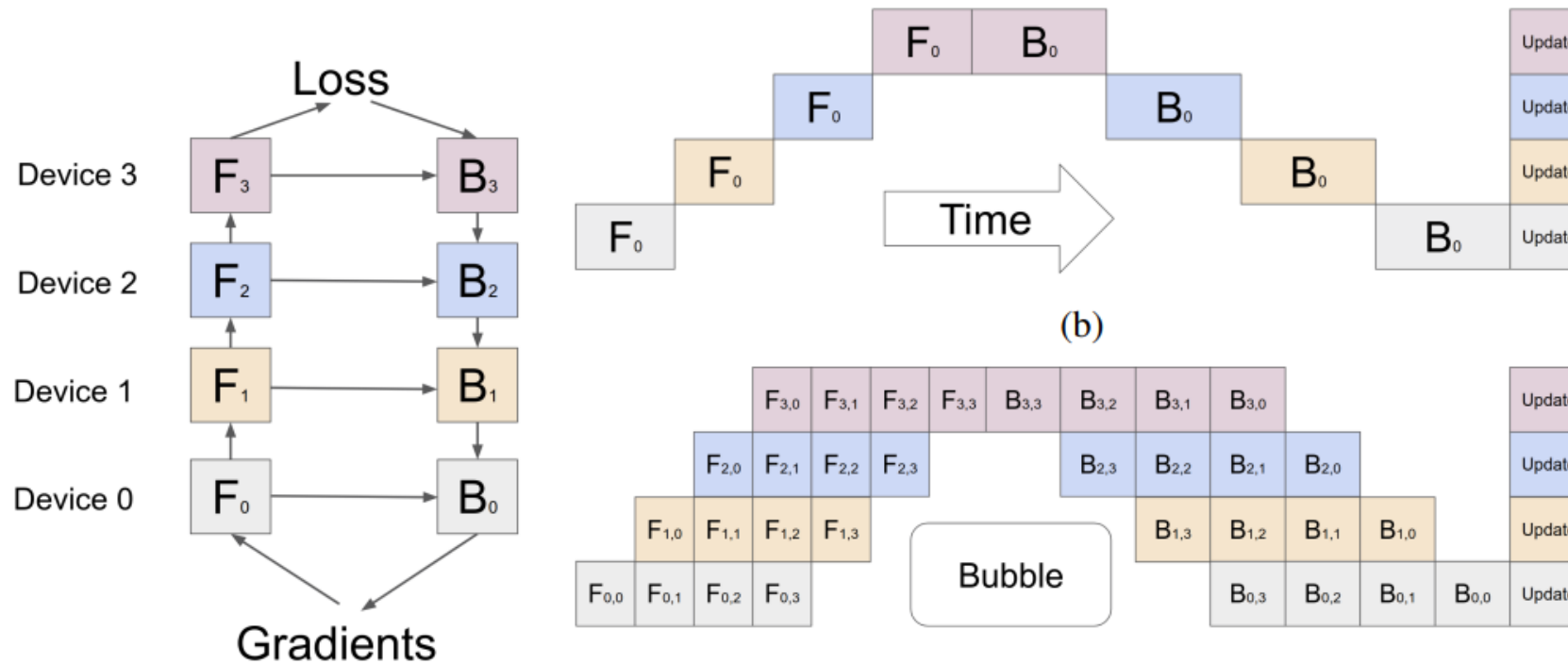
- Input batch + activations

M = number of parameters in the model

Distributed Training Strategies

- Pipeline Parallelism
- Tensor Parallelism
- 3D Parallelism
- ZeRO-Style Data Parallelism

Pipeline Parallelism



Supported in:

- [PyTorch](#)
- [DeepSpeed](#)
- [Megatron-LM](#)

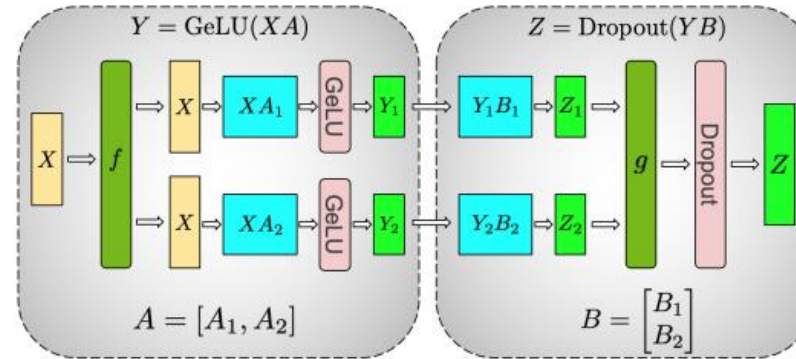
- Naïve model parallelism leads to severe underutilization
- Gpipe divides batch into micro-batches, enabling different device to work on different micro-batches, reducing pipeline bubbles and improving utilization

Tensor Parallelism

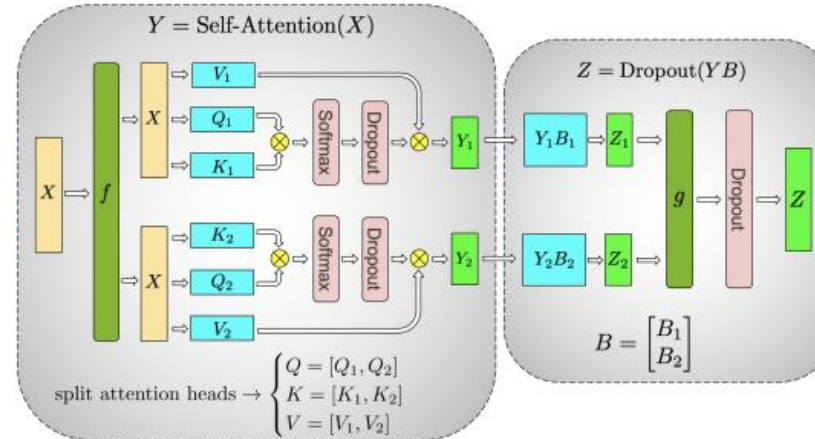
Splice tensors across GPUs

+

synchronization primitives
(e.g., all-reduce)



(a) MLP

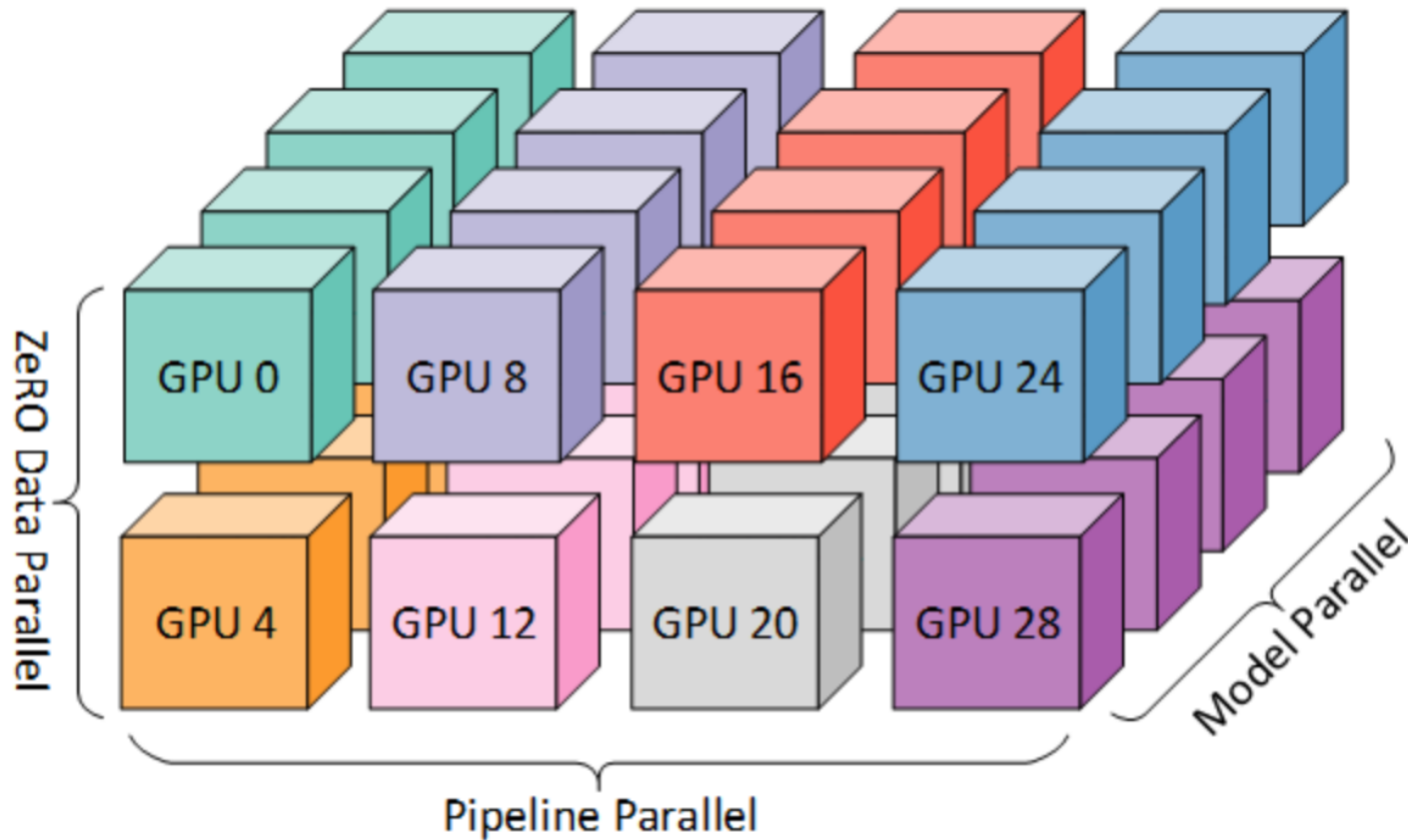


(b) Self-Attention

Supported in:

- [DeepSpeed](#)
- [Megatron-LM](#)

3D Parallelism

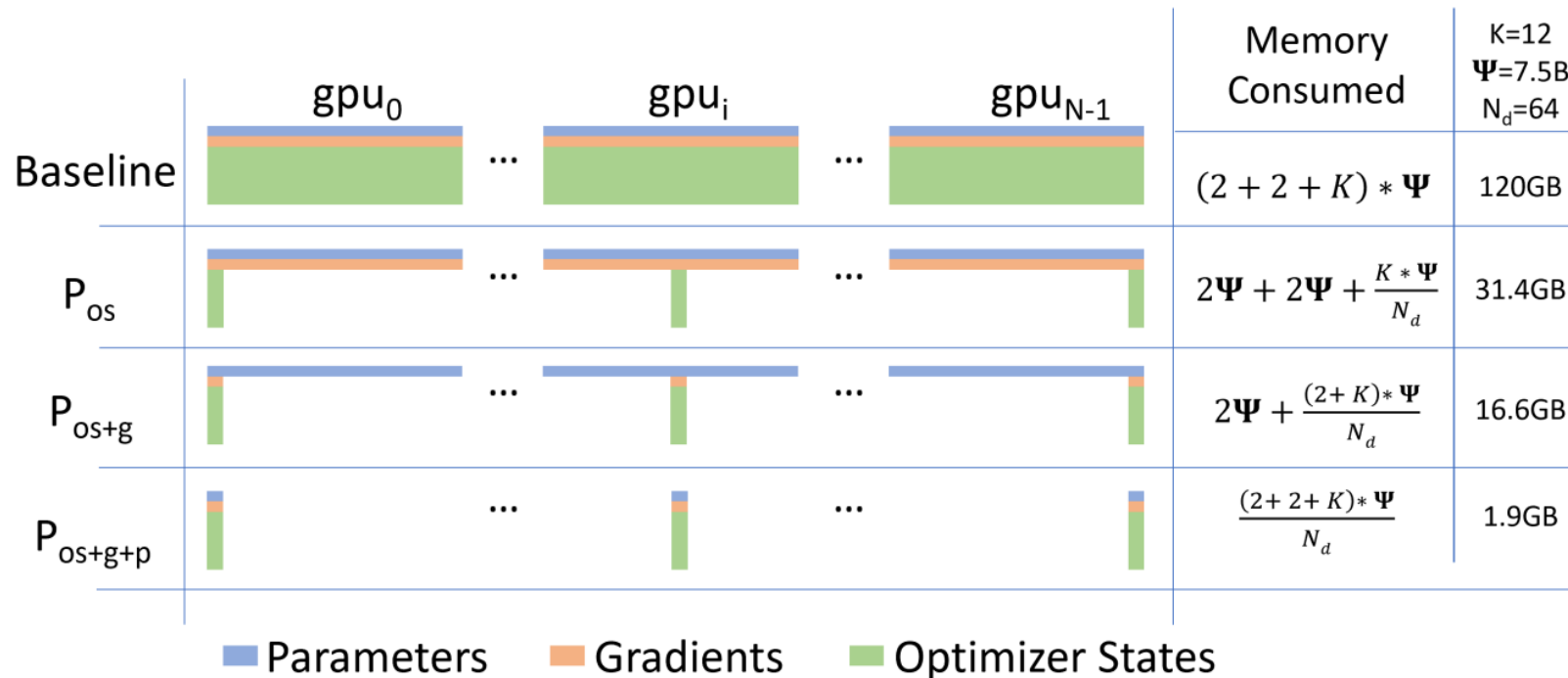


Supported in:
• [DeepSpeed](#)
• [Megatron-LM](#)

[DeepSpeed-extreme-scale-model-training-for-everyone](#)

ZeRO-Style Data Parallelism

- ZeRO removes the redundancy across data parallel process
- Partitioning optimizer states, gradients and parameters (3 stages)



Supported in:

- [DeepSpeed](#)
- [PyTorch](#)

Large Models Need Parallelism

	Max Parameter (in billions)	Max Parallelism	Compute Efficiency	Usability (Model Rewrite)
Data Parallel (DP)	Approx. 1.2	>1000	Very Good	Great
Tensor Parallel (TP)	Approx. 20	Approx. 16	Good	Needs Model Rewrite
TP + DP	Approx. 20	> 1000	Good	Needs Model Rewrite
Pipeline Parallel (PP)	Approx. 100	Approx. 128	Very Good	Needs Model Rewrite
PP + DP	Approx. 100	> 1000	Very Good	Needs Model Rewrite
TP + PP + DP	> 1000	> 1000	Very Good	Needs Significant Model Rewrite
ZeRO	> 1000	> 1000	Very Good	Great

More Interesting Work on Training

- Alpa: Automating Inter- and Intra-Operator Parallelism for Distributed Deep Learning, OSDI 22
- Sequence Parallelism: Making 4D Parallelism Possible, ACL 2023
- Tutel: An efficient mixture-of-experts implementation for large DNN model training, 2023
- Bamboo: Making Preemptible Instances Resilient for Affordable Training of Large DNNs, NSDI 2023
- ...

Tentative Schedule

Hardware Resource

Delta

- Home page:
<https://www.ncsa.illinois.edu/research/project-highlights/delta/>
- 100 quad A100 GPU node, each with 4 A100
- 100 quad A40 GPU node, each with 4 A40
- 5 8-way A100 GPU, each with 8 A100
- 1 MI100 node, 8 MI100



Delta Onboarding Page

- https://docs.ncsa.illinois.edu/systems/delta/en/latest/user_guide/accessing.html

The screenshot displays the 'Delta Login Methods' page. On the left is a dark navigation sidebar with the following items: 'Frequently Asked Questions', 'Support and Services', 'Getting Help', 'USER GUIDE' (highlighted in blue), 'System Architecture', 'Account Administration', 'Delta Login Methods' (expanded), 'Direct Access Login Nodes' (expanded), 'Open OnDemand', 'VS Code', 'Good Cluster Citizenship', 'Data Management', 'Programming Environment (Building Software)', 'Job Accounting', 'Running Jobs', 'Installed Software', 'Visualization', 'Containers', 'Services', 'Debugging and Performance Analysis', and 'Acknowledging Delta'. The main content area has a breadcrumb 'Delta Login Methods' and a link to 'Edit on GitHub'. The title is 'Delta Login Methods' and the sub-section is 'Direct Access Login Nodes'. The text explains that direct access is via SSH using NCSA credentials and Duo MFA, and provides a link to the 'NCSA Allocation and Account Management' page. It also mentions that for ACCESS awarded projects, users should find their local NCSA username on the 'ACCESS Profile page' and refer to the 'Resource Provider Site Usernames' table. Below this is a table titled 'Login Node Hostnames' with two columns: 'Login Node Hostname' and 'Example Usage with SSH'. The table contains one row with the hostname 'dt-login01.delta.ncsa.illinois.edu' and the example command 'ssh -Y username@dt-login01.delta.ncsa.illinois.edu', with a note that '-Y allows X11 forwarding from Linux hosts'.

Delta Login Methods

Delta Login Methods

Direct Access Login Nodes

Direct access to the Delta login nodes is via SSH using your NCSA username, password, and NCSA Duo MFA. See the [NCSA Allocation and Account Management](#) page for links to NCSA Identity and NCSA Duo services. The login nodes provide access to the CPU and GPU resources on Delta.

See [NCSA Allocation and Account Management](#) for the steps to change your NCSA password for direct access and set up NCSA Duo.

For ACCESS awarded projects, to find your local NCSA username go to your [ACCESS Profile page](#) and scroll to the bottom for the **Resource Provider Site Usernames** table. If you do not know your NCSA username, submit a support request ([Getting Help](#)) for assistance.

Login Node Hostname	Example Usage with SSH
dt-login01.delta.ncsa.illinois.edu	<pre>ssh -Y username@dt-login01.delta.ncsa.illinois.edu</pre> <p>(-Y allows X11 forwarding from Linux hosts)</p>

Step 1: Create ACCESS ID



Need access to computing, data analysis, or storage resources?

You're in the right place! Read more below, or [login](#) to get started.

What is an **allocation**?

To get started, you need an ACCESS project and some resource units you can spend.

Your ACCESS project and resource units are what we refer to as an Allocation. An allocation is your project to use a

Which **resources**?

We've got modeling and analysis systems, GPU-oriented systems, large-memory nodes, storage, and more. Resource providers have designed their systems to serve a wide range of research and education needs — including

Ready to get **started**?


It costs you nothing (really!), and you don't need an NSF award. To begin, you just need to

[LOGIN](#)

or

Step 2: Submit Resource Requests

https://allocations.access-ci.org



My Projects Get Started Available Resources ACCESS Impact Policies & How-To About

My Projects

REQUEST NEW PROJECT GET HELP

▼ : AI efficiency research projects **Incomplete**

Accelerate: Submitted Jan 17, 2024 EDIT DELETE

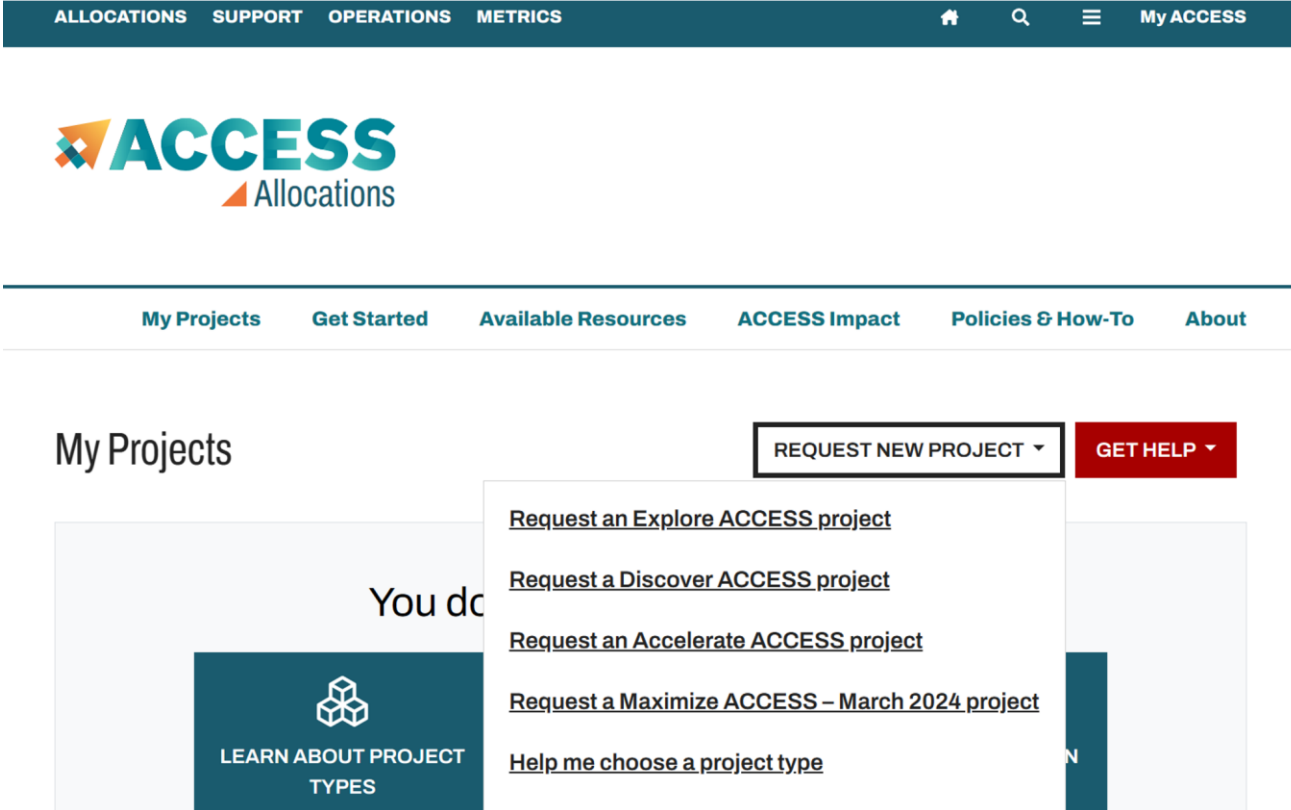
You are viewing an incomplete request. You cannot manage resources or users for this request.

Overview Credits + Resources Users + Roles History

Role	Users	Action Details	Status
PI	Minjia Zhang	New: Jan 17, 2024 EDIT DELETE	Incomplete
Allocation Manager	Minjia Zhang		

Step 2: Submit Resource Requests

- **EXPLORE** — Great for resource evaluation, graduate student projects, small classes and training events, benchmarking, code development and porting, and similar small-scale uses.
- **DISCOVER** — Designed for research grants with modest resource needs, Campus Champions, large classes and training events, NSF graduate fellowships, benchmarking and code testing at scale, and gateway development.
- **ACCELERATE** — Best for experienced users with mid-scale resource needs, consolidating multi-grant programs, collaborative projects, preparing for Maximize ACCESS requests, and gateways with growing communities.
- **MAXIMIZE** — The choice for large-scale research activities that need more resources than the limit for Accelerate ACCESS projects.



The screenshot displays the ACCESS Allocations website interface. At the top, a dark blue navigation bar contains the menu items: ALLOCATIONS, SUPPORT, OPERATIONS, METRICS, a home icon, a search icon, a hamburger menu icon, and My ACCESS. Below this is the ACCESS Allocations logo. A secondary navigation bar features links for My Projects, Get Started, Available Resources, ACCESS Impact, Policies & How-To, and About. The main content area is titled 'My Projects' and includes a 'REQUEST NEW PROJECT' dropdown menu and a 'GET HELP' button. The dropdown menu is open, showing options: 'Request an Explore ACCESS project', 'Request a Discover ACCESS project', 'Request an Accelerate ACCESS project', 'Request a Maximize ACCESS – March 2024 project', and 'Help me choose a project type'. A 'LEARN ABOUT PROJECT TYPES' button is also visible in the background.

Step 3: SSH Login

- Support ssh login
- Maintaining Persistent Sessions: tmux

Delta Login Methods

Direct Access Login Nodes

Direct access to the Delta login nodes is via SSH using your NCSA username, password, and NCSA Duo MFA. See the [NCSA Allocation and Account Management](#) page for links to NCSA Identity and NCSA Duo services. The login nodes provide access to the CPU and GPU resources on Delta.

See [NCSA Allocation and Account Management](#) for the steps to change your NCSA password for direct access and set up NCSA Duo.

For ACCESS awarded projects, to find your local NCSA username go to your [ACCESS Profile page](#) and scroll to the bottom for the **Resource Provider Site Usernames** table. If you do not know your NCSA username, submit a support request ([Getting Help](#)) for assistance.

Login Node Hostnames

Login Node Hostname	Example Usage with SSH
dt-login01.delta.ncsa.illinois.edu	<pre>ssh -Y username@dt-login01.delta.ncsa.illinois.edu</pre> <p>(-Y allows X11 forwarding from Linux hosts)</p>
dt-login02.delta.ncsa.illinois.edu	<pre>ssh -l username dt-login02.delta.ncsa.illinois.edu</pre> <p>(-l username alt. syntax for <code>user@host</code>)</p>
login.delta.ncsa.illinois.edu (round robin DNS name for the set of login nodes)	<pre>ssh username@login.delta.ncsa.illinois.edu</pre>

Delta

- Please give it a try
 - Request access (A100, A40, AMD,...)
 - Ssh login
 - Run a small training/inference job, say PyTorch examples
 - Do preliminary performance profiling
- Let me know if you run into any issues
 - Single GPU allocation
 - Multi-GPU allocation
 - Interactive development
 - Isolation
 - Persistent storage