UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

# CS 498: Machine Learning System Spring 2026

Minjia Zhang

The Grainger College of Engineering

# Today

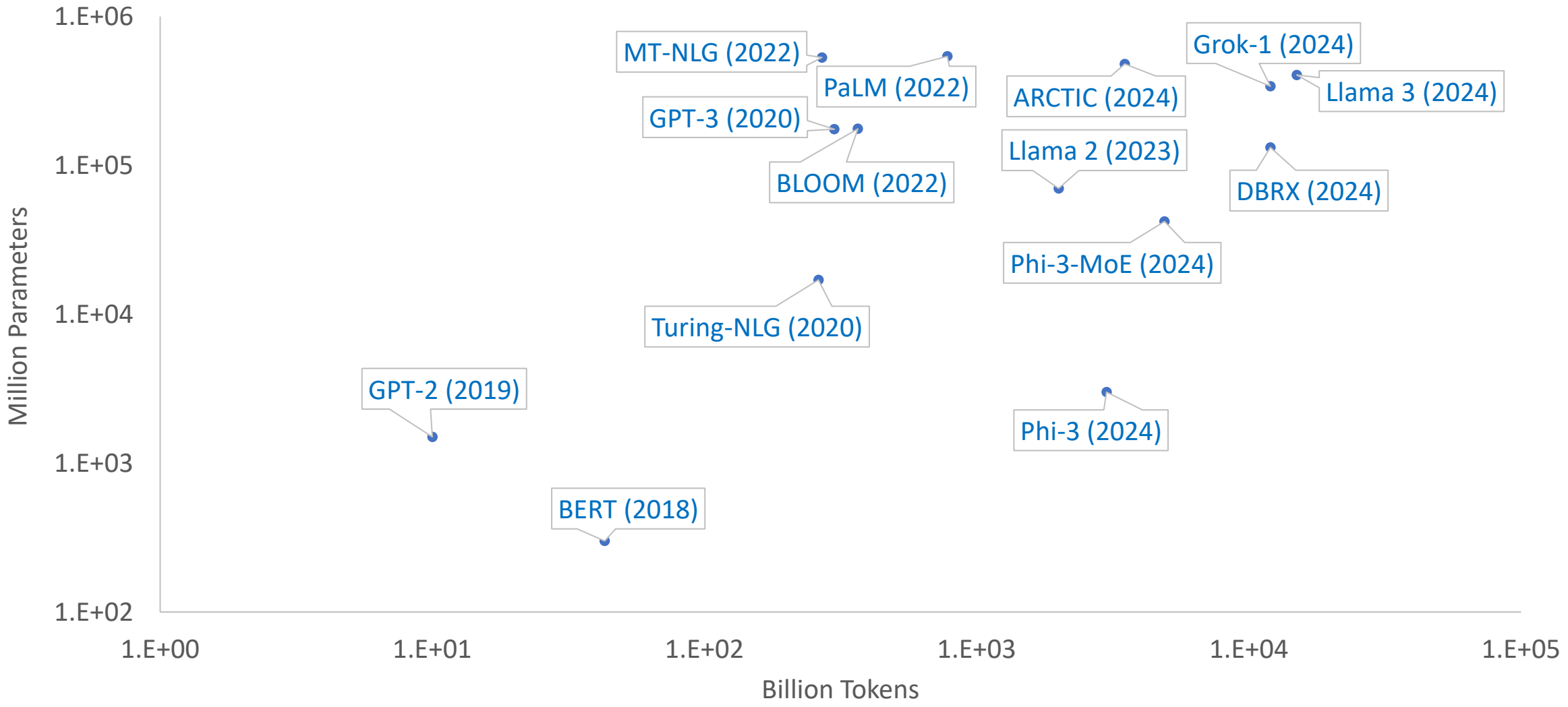
**ZeRO-Style Data Parallelism (Fully-Sharded Data Parallelism)**

- Motivation
- ZeRO capability overview
- Understanding Memory Consumption
- ZeRO-DP: ZeRO powered data parallelism

**Learning Objectives:**

- Understand more deeply the memory bottlenecks of DDP training
- Explain how ZeRO progressively removes memory redundancy

# Evolution of AI Models



# State-of-art and its limitations

| | Max Parameter (in billions) | Max Parallelism | Compute Efficiency | Usability (Model Rewrite) |
|---|---|---|---|---|
| **Data Parallel (DP)** | Approx. 1.2 | >1000 | Very Good | Great |

*Mixed precision Adam on Cluster of DGX-2 with NVIDIA 32 GB V100 GPUs

# State-of-art and its limitations

| | Max Parameter (in billions) | Max Parallelism | Compute Efficiency | Usability (Model Rewrite) |
|---|---|---|---|---|
| **Data Parallel (DP)** | Approx. 1.2 | >1000 | Very Good | Great |

Democratization of large model training requires easy to use solutions that can support large model sizes and parallelism degree while remaining highly efficient

*Mixed precision Adam on Cluster of DGX-2 with NVIDIA 32 GB V100 GPUs

# State-of-art and its limitations

| | Max Parameter (in billions) | Max Parallelism | Compute Efficiency | Usability (Model Rewrite) |
|---|---|---|---|---|
| **Data Parallel (DP)** | Approx. 1.2 | >1000 | Very Good | Great |
| **Model Parallel (MP)** | Approx. 20 | Approx. 16 | Good | Needs Model Rewrite |

Democratization of large model training requires easy to use solutions that can support large model sizes and parallelism degree while remaining highly efficient

*Mixed precision Adam on Cluster of DGX-2 with NVIDIA 32 GB V100 GPUs

# State-of-art and its limitations

| | Max Parameter (in billions) | Max Parallelism | Compute Efficiency | Usability (Model Rewrite) |
|---|---|---|---|---|
| **Data Parallel (DP)** | Approx. 1.2 | >1000 | Very Good | Great |
| **Model Parallel (MP)** | Approx. 20 | Approx. 16 | Good | Needs Model Rewrite |
| **MP + DP** | Approx. 20 | > 1000 | Good | Needs Model Rewrite |

Democratization of large model training requires easy to use solutions that can support large model sizes and parallelism degree while remaining highly efficient

*Mixed precision Adam on Cluster of DGX-2 with NVIDIA 32 GB V100 GPUs

# State-of-art and its limitations

| | Max Parameter (in billions) | Max Parallelism | Compute Efficiency | Usability (Model Rewrite) |
|---|---|---|---|---|
| **Data Parallel (DP)** | Approx. 1.2 | >1000 | Very Good | Great |
| **Model Parallel (MP)** | Approx. 20 | Approx. 16 | Good | Needs Model Rewrite |
| **MP + DP** | Approx. 20 | > 1000 | Good | Needs Model Rewrite |
| **Pipeline Parallel (PP)** | Approx. 100 | Approx. 128 | Very Good | Needs Model Rewrite |

Democratization of large model training requires easy to use solutions that can support large model sizes and parallelism degree while remaining highly efficient

*Mixed precision Adam on Cluster of DGX-2 with NVIDIA 32 GB V100 GPUs

# State-of-art and its limitations

Democratization of large model training requires easy to use solutions that can support large model sizes and parallelism degree while remaining highly efficient

| | Max Parameter (in billions) | Max Parallelism | Compute Efficiency | Usability (Model Rewrite) |
|---|---|---|---|---|
| **Data Parallel (DP)** | Approx. 1.2 | >1000 | Very Good | Great |
| **Model Parallel (MP)** | Approx. 20 | Approx. 16 | Good | Needs Model Rewrite |
| **MP + 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 |

*Mixed precision Adam on Cluster of DGX-2 with NVIDIA 32 GB V100 GPUs

# State-of-art and its limitations

Democratization of large model training requires easy to use solutions that can support large model sizes and parallelism degree while remaining highly efficient

| | Max Parameter (in billions) | Max Parallelism | Compute Efficiency | Usability (Model Rewrite) |
|---|---|---|---|---|
| **Data Parallel (DP)** | Approx. 1.2 | >1000 | Very Good | Great |
| **Model Parallel (MP)** | Approx. 20 | Approx. 16 | Good | Needs Model Rewrite |
| **MP + 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 |
| **MP + PP + DP** | > 1000 | > 1000 | Very Good | Needs Significant Model Rewrite |

*Mixed precision Adam on Cluster of DGX-2 with NVIDIA 32 GB V100 GPUs

# Motivation

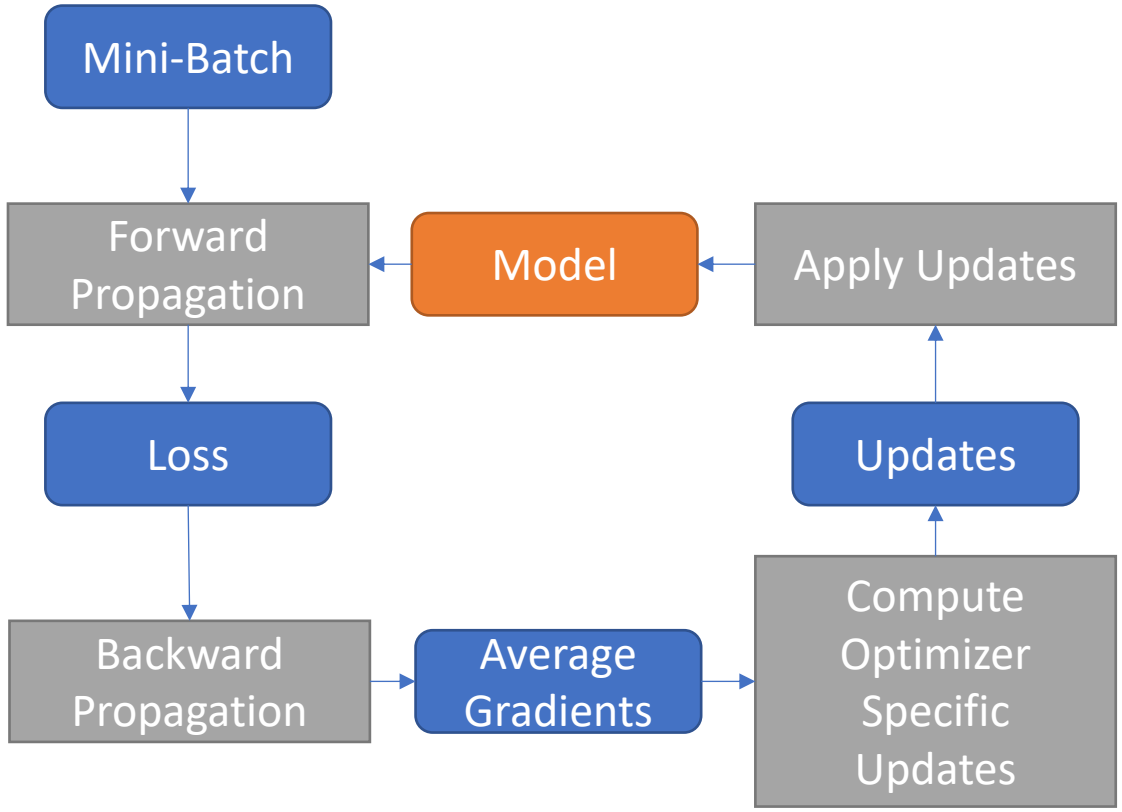
Democratization of large model training requires easy to use solutions that can support large model sizes and parallelism degree while remaining highly efficient

| | Max Parameter (in billions) | Max Parallelism | Compute Efficiency | Usability (Model Rewrite) |
|---|---|---|---|---|
| **Data Parallel (DP)** | Approx. 1.2 | >1000 | Very Good | Great |
| **Model Parallel (MP)** | Approx. 20 | Approx. 16 | Good | Needs Model Rewrite |
| **MP + 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 |
| **MP + PP + DP** | > 1000 | > 1000 | Very Good | Needs Significant Model Rewrite |
| | | | | |
| ***ZeRO*** | *> 1000* | *> 1000* | Very Good | Great |

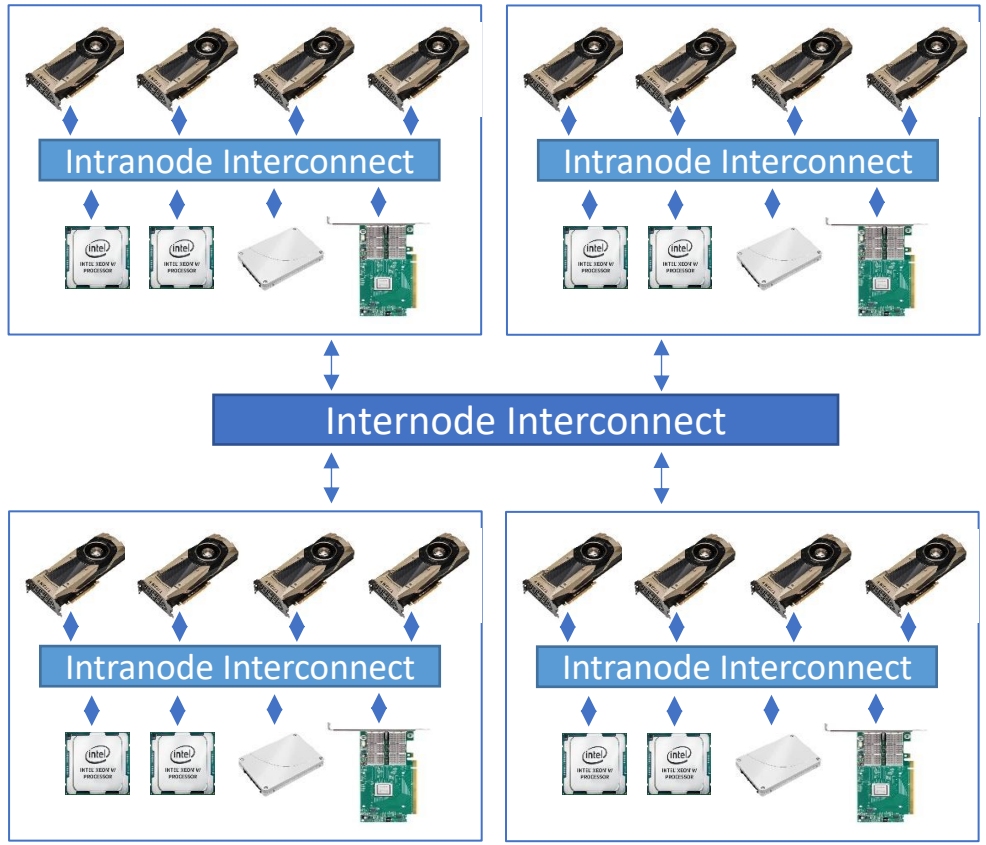
# Outline

- Motivation
- ZeRO capability overview
- **Understanding Memory Consumption**
- ZeRO-DP: ZeRO powered data parallelism
- Evaluation

# Distributed Data Parallel Training Overview

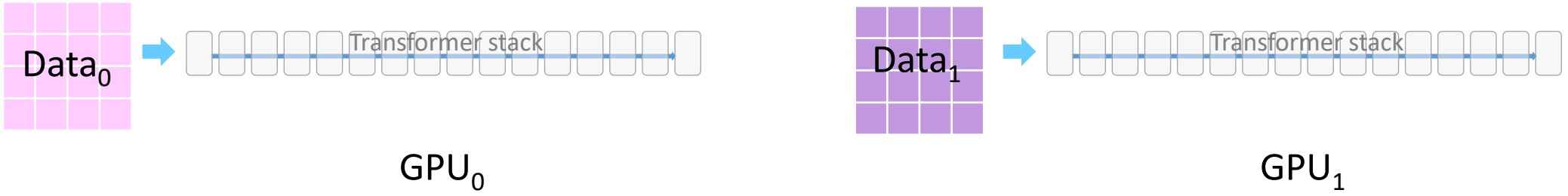



Data Parallel Training Loop




Distributed GPU Cluster

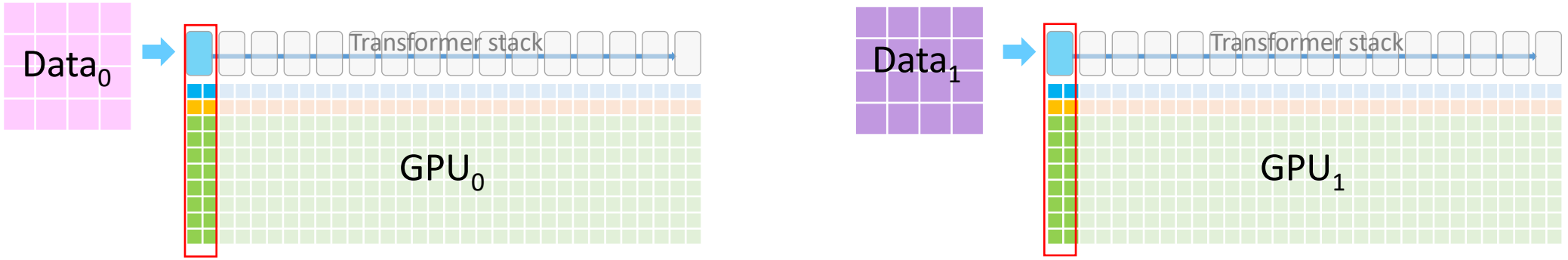
# Understanding Memory Consumption




A 16-layer transformer model = 1 layer

*Mixed Precision Training (ICLR '18) with Adam Optimizer

# Understanding Memory Consumption




Each cell represents GPU memory used by its corresponding transformer layer

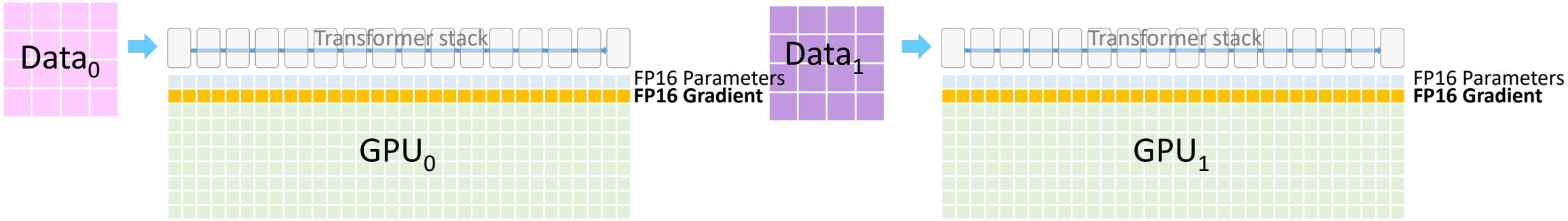
*Mixed Precision Training (ICLR '18) with Adam Optimizer

# Understanding Memory Consumption


- FP16 parameter

*Mixed Precision Training (ICLR '18) with Adam Optimizer

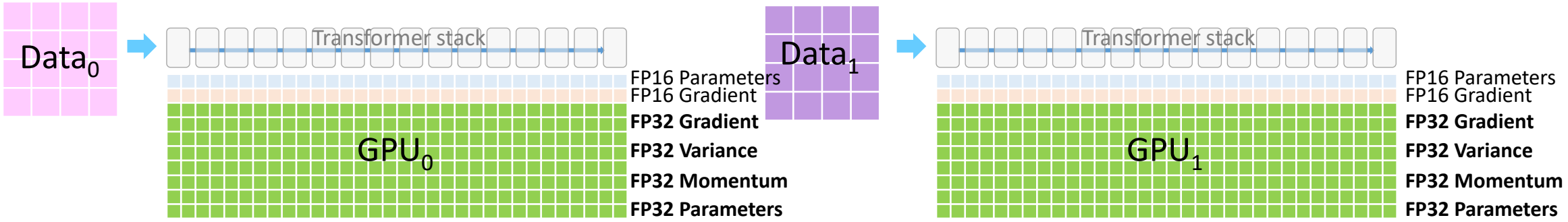
# Understanding Memory Consumption




- FP16 parameter
- FP16 Gradients

*Mixed Precision Training (ICLR '18) with Adam Optimizer

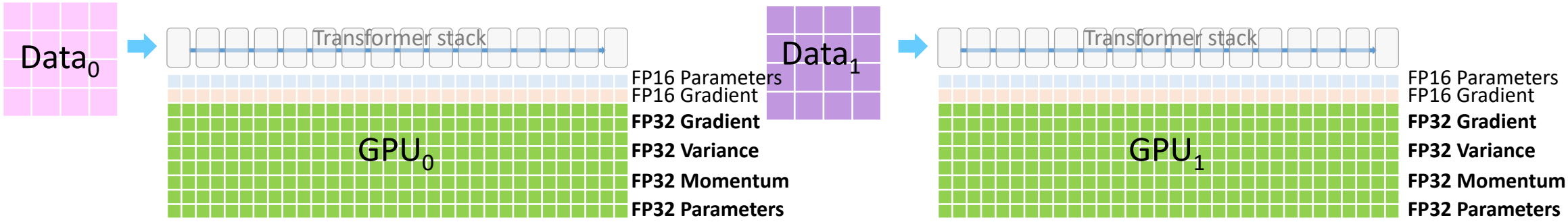
# Understanding Memory Consumption




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

*Mixed Precision Training (ICLR '18) with Adam Optimizer

# Understanding Memory Consumption

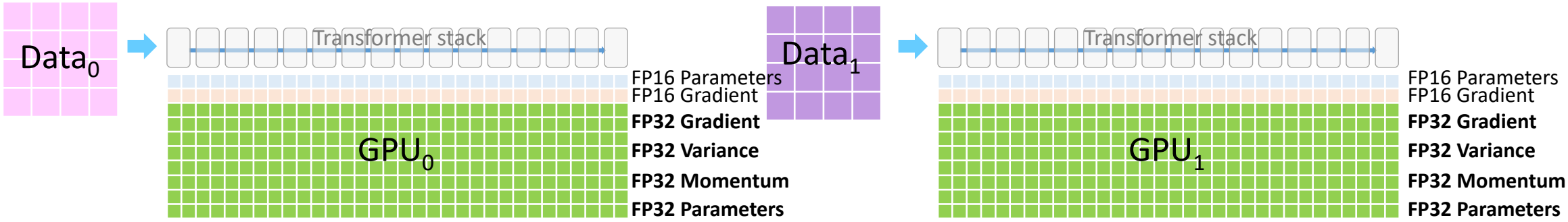



- FP16 parameter **: 2M bytes**
- FP16 Gradients **: 2M bytes**
- FP32 Optimizer States **: 16M bytes**
  - Gradients, Variance, Momentum, Parameters

M = number of parameters in the model

*Mixed Precision Training (ICLR '18) with Adam Optimizer

# Understanding Memory Consumption




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

M = number of parameters in the model

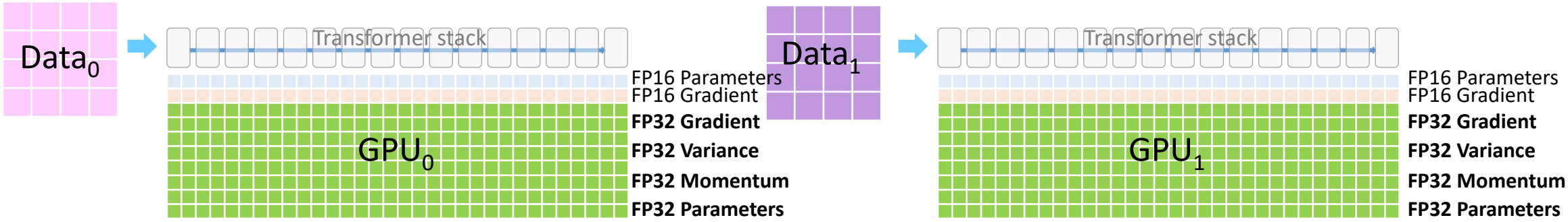
Example 1B parameter model -> 20GB/GPU

Memory consumption doesn't include:

- Input batch + activations

*Mixed Precision Training (ICLR '18) with Adam Optimizer

# 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

M = number of parameters in the model

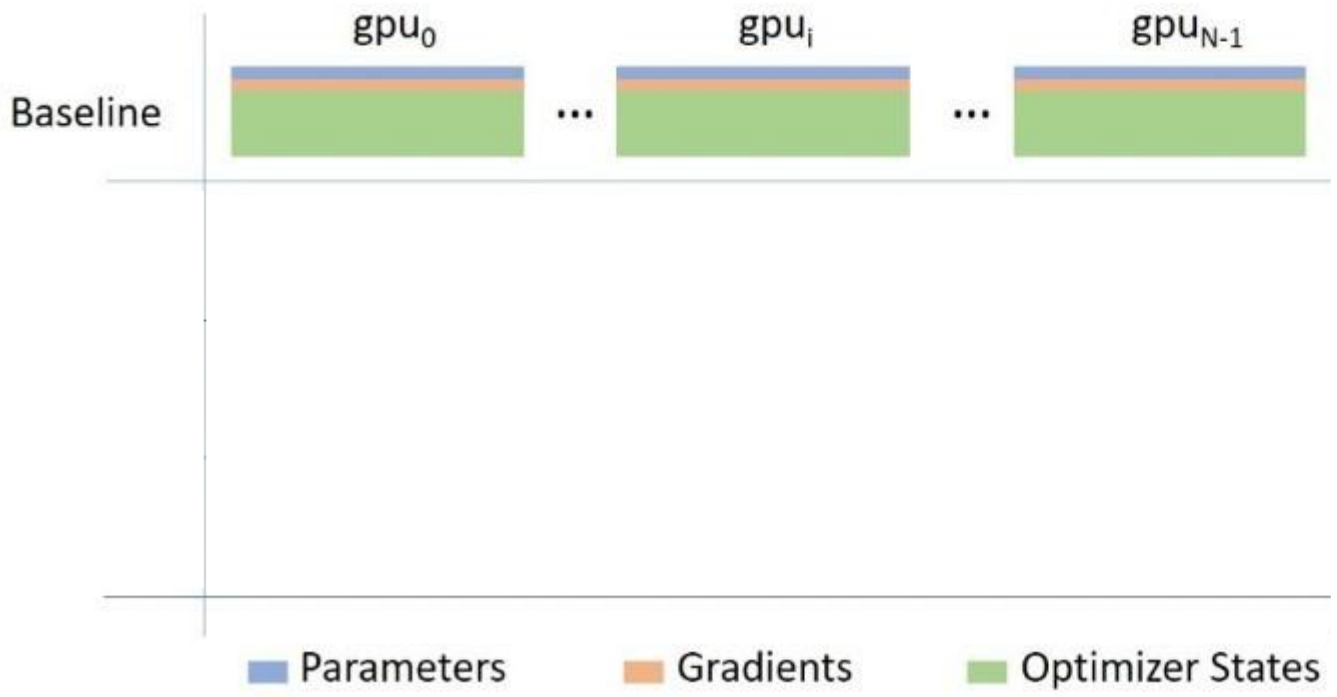
*Mixed Precision Training (ICLR '18) with Adam Optimizer

# Outline

- Motivation
- ZeRO capability overview
- Understanding Memory Consumption
- **ZeRO-DP: ZeRO powered data parallelism**
- Evaluation

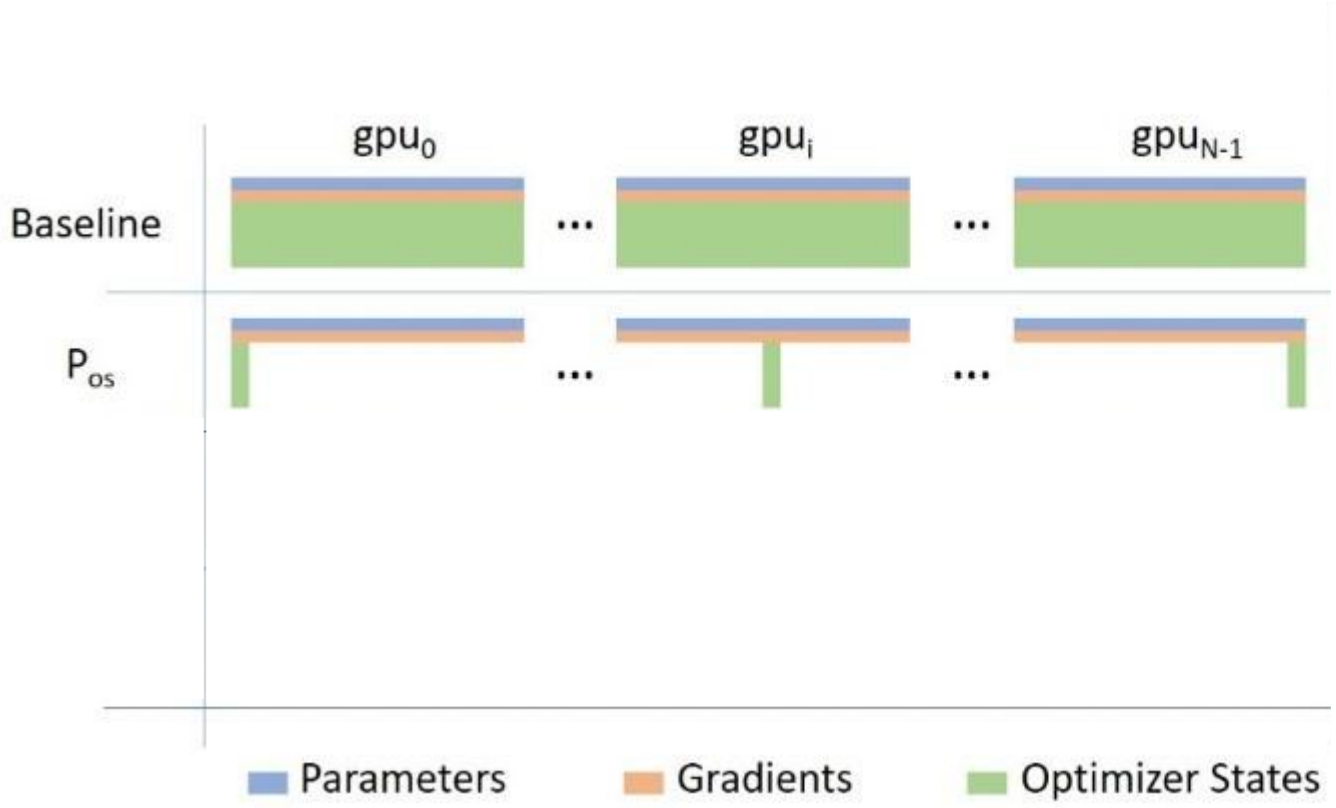
# ZeRO-DP: ZeRO powered Data Parallelism

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



# ZeRO-DP: ZeRO powered Data Parallelism

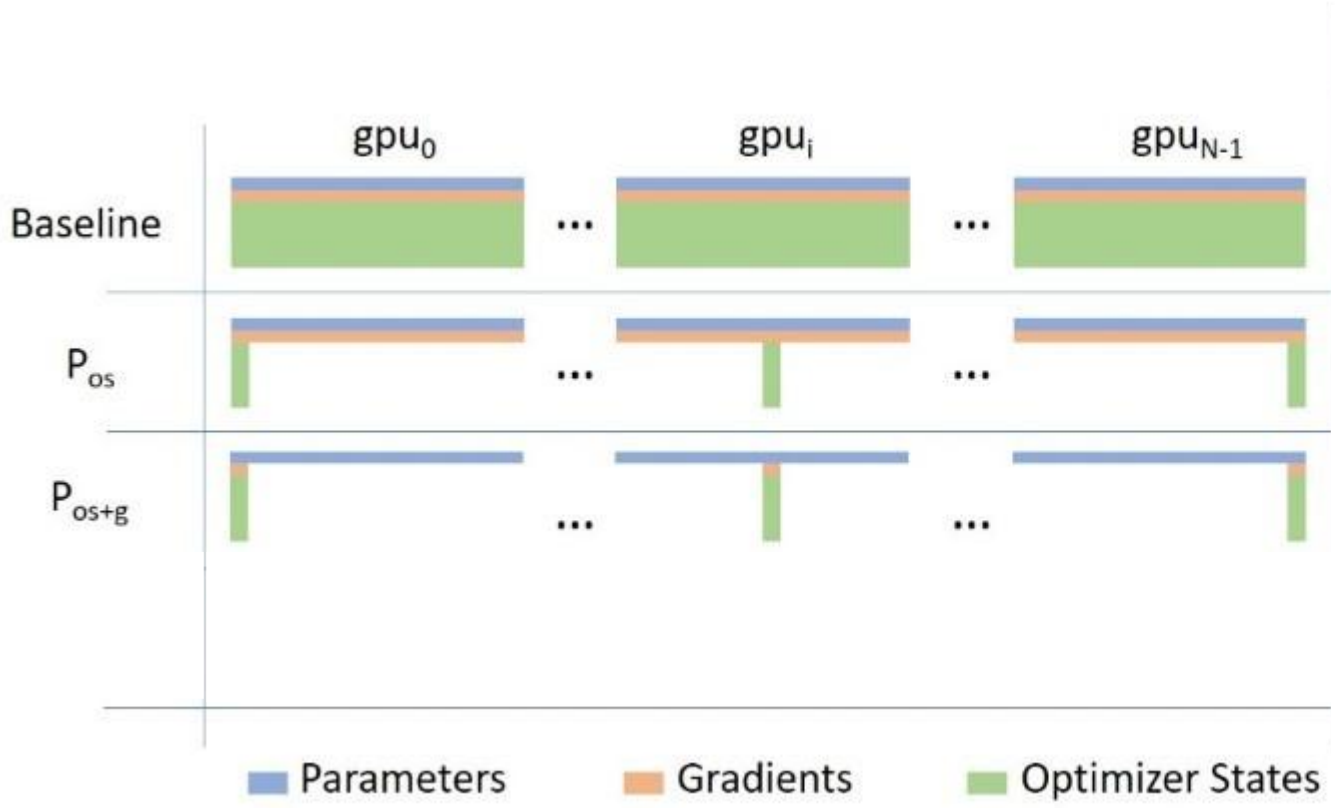
- ZeRO removes the redundancy across data parallel process
- Partitioning optimizer states, gradients and parameters (3 stages)




**Stage 1 ($P_{os}$)**

# ZeRO-DP: ZeRO powered Data Parallelism

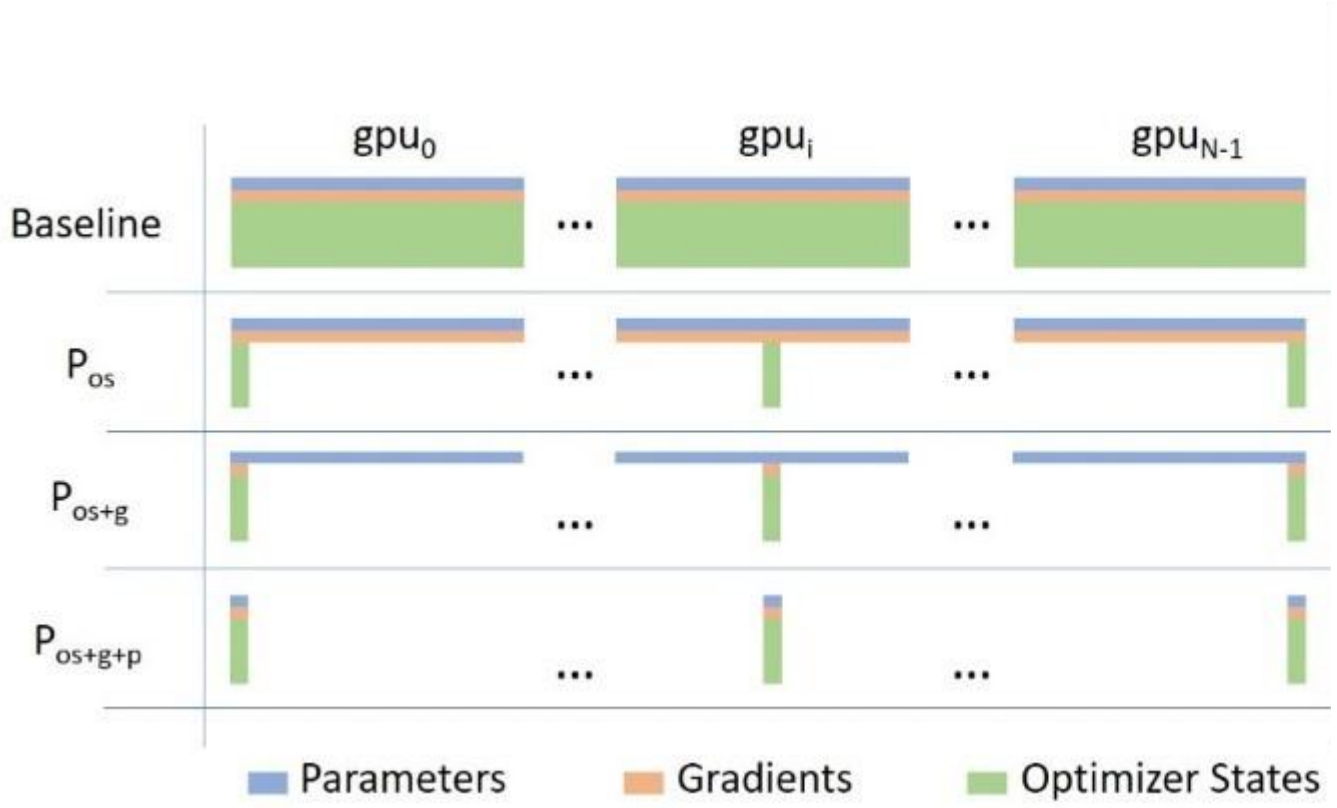
- ZeRO removes the redundancy across data parallel process
- Partitioning optimizer states, gradients and parameters (3 stages)




**Stage 2 ($P_{os+g}$)**

# ZeRO-DP: ZeRO powered Data Parallelism

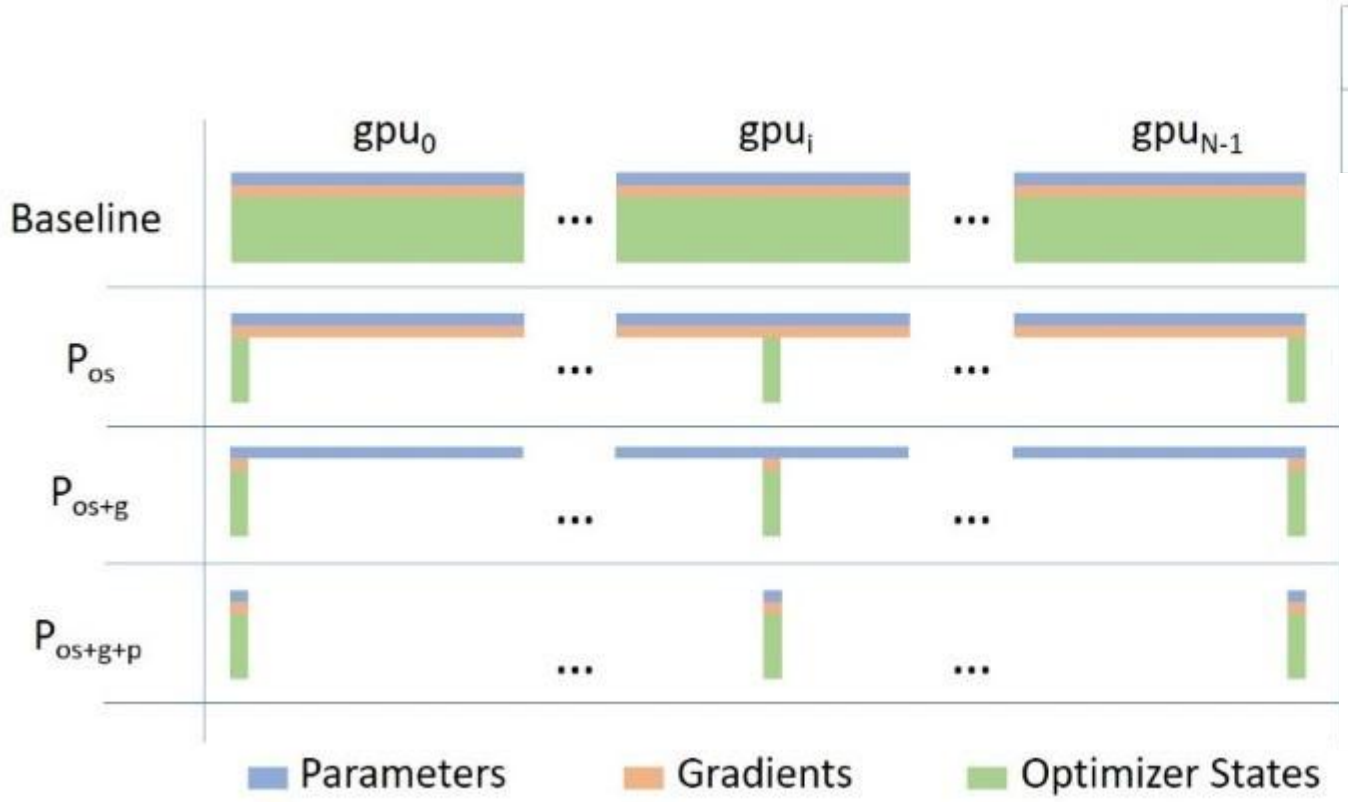
- ZeRO removes the redundancy across data parallel process
- Partitioning optimizer states, gradients and parameters (3 stages)




**Stage 3 ($P_{os+g+p}$)**

# ZeRO-DP: ZeRO powered Data Parallelism

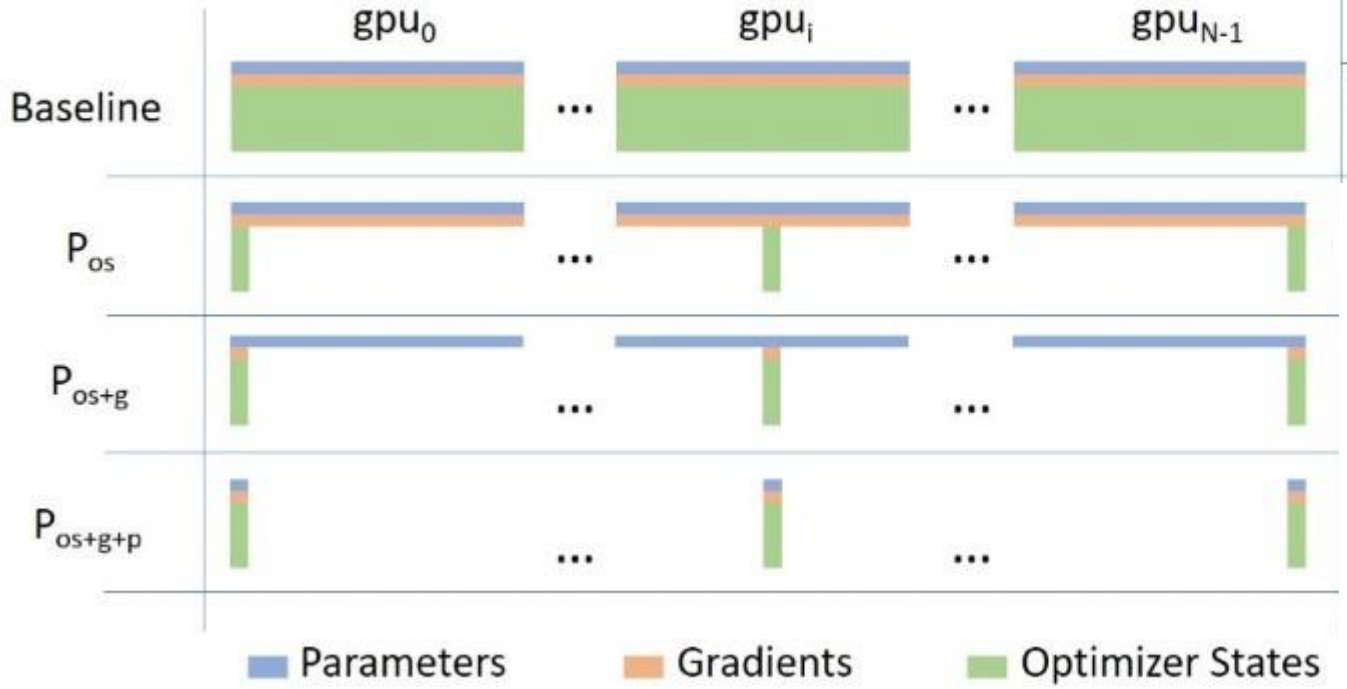
- ZeRO removes the redundancy across data parallel process
- Partitioning optimizer states, gradients and parameters (3 stages)




| Memory Consumption | |
|---|---|
| Formulation | Specific Example<br>K=12 Ψ=7.5B $N_d$=64 |

# ZeRO-DP: ZeRO powered Data Parallelism

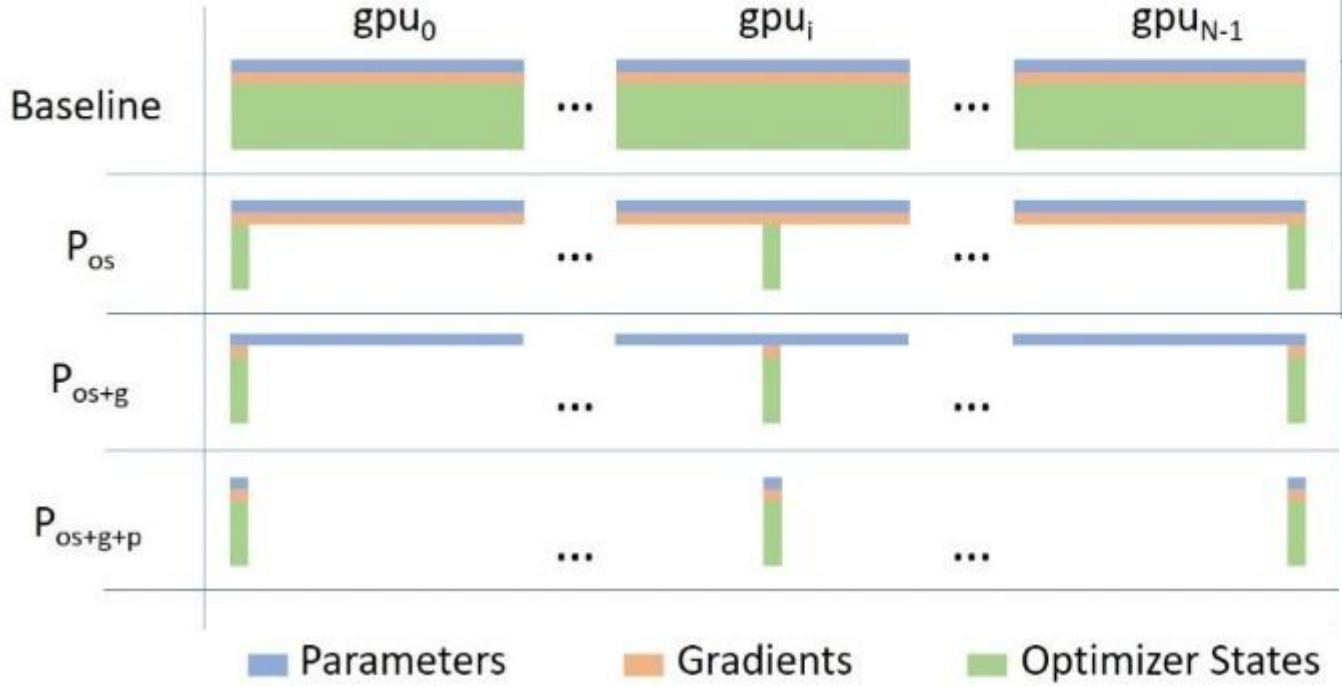
- ZeRO removes the redundancy across data parallel process
- Partitioning optimizer states, gradients and parameters (3 stages)




| Memory Consumption | |
|---|---|
| Formulation | Specific Example K=12 Ψ=7.5B $N_d$=64 |
| $(2+2+K)*\Psi$ | 120GB |

# ZeRO-DP: ZeRO powered Data Parallelism

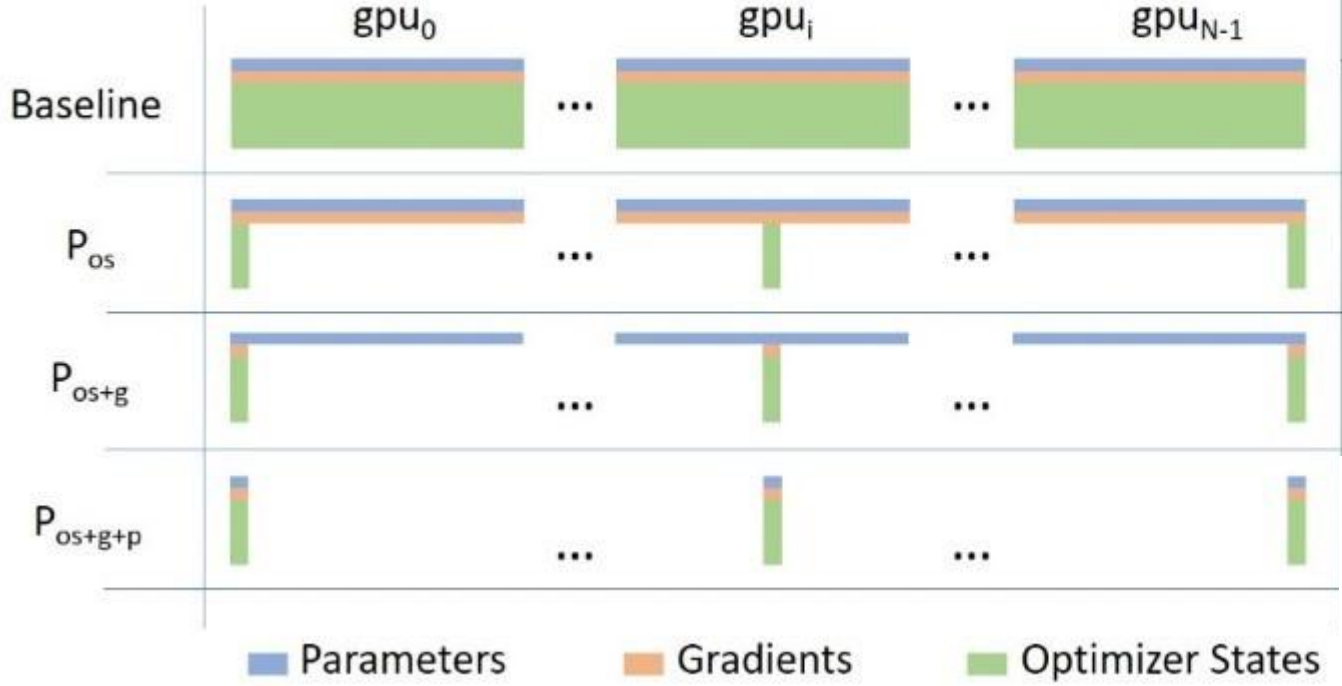
- ZeRO removes the redundancy across data parallel process
- Partitioning optimizer states, gradients and parameters (3 stages)




| Memory Consumption | |
|---|---|
| Formulation | Specific Example K=12 Ψ=7.5B $N_d$=64 |
| $(2+2+K)*\Psi$ | 120GB |
| $2\Psi+2\Psi+\frac{K*\Psi}{N_d}$ | 31.4GB |

# ZeRO-DP: ZeRO powered Data Parallelism

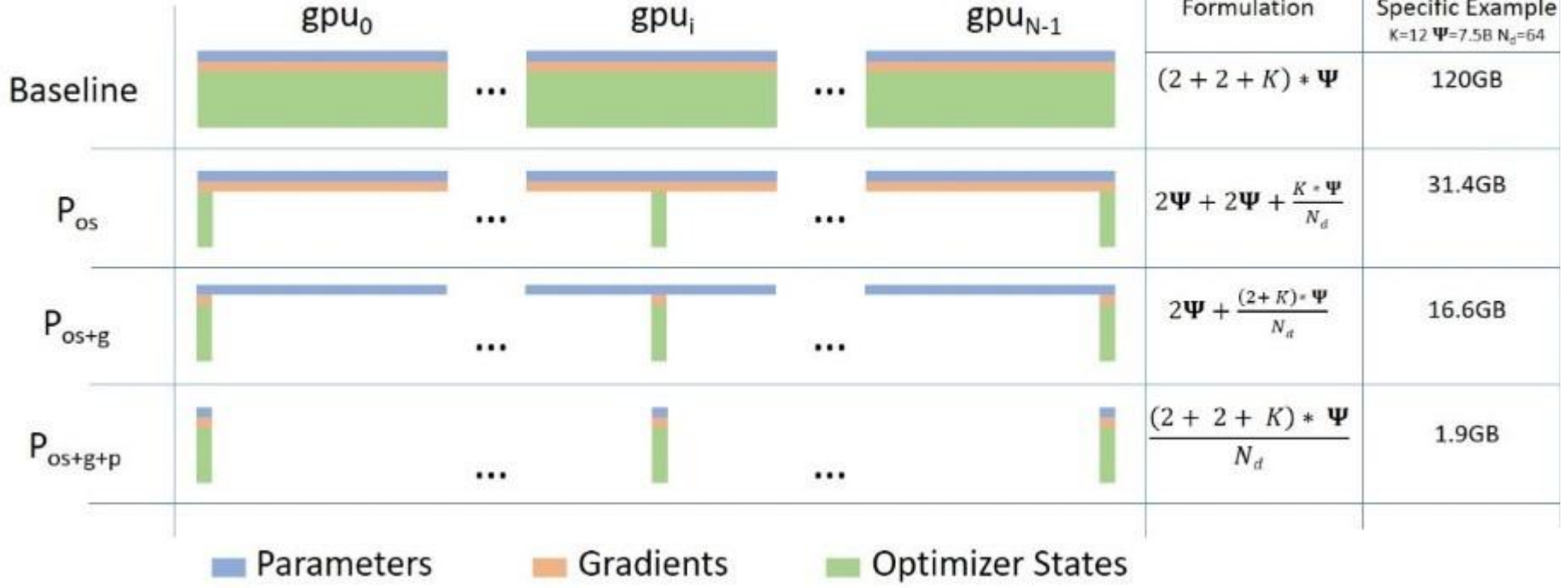
- ZeRO removes the redundancy across data parallel process
- Partitioning optimizer states, gradients and parameters (3 stages)




| Memory Consumption | |
|---|---|
| Formulation | Specific Example K=12 Ψ=7.5B $N_d$=64 |
| $(2+2+K)*\Psi$ | 120GB |
| $2\Psi+2\Psi+\frac{K*\Psi}{N_d}$ | 31.4GB |
| $2\Psi+\frac{(2+K)*\Psi}{N_d}$ | 16.6GB |

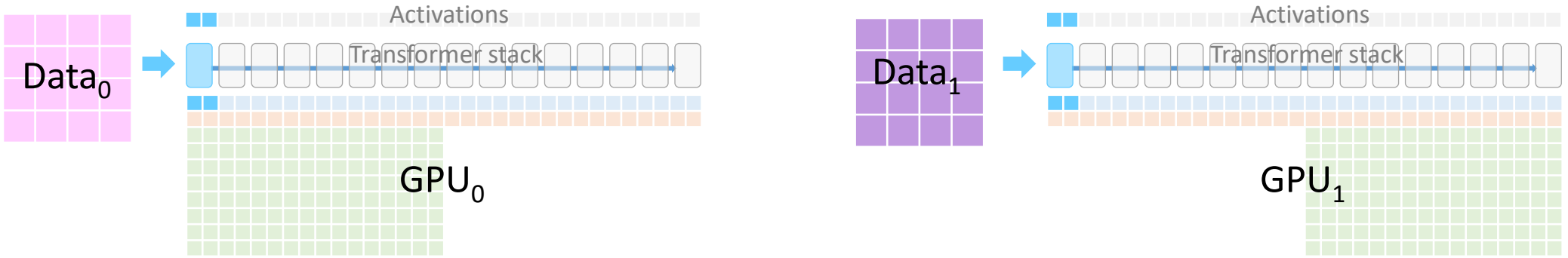
# ZeRO-DP: ZeRO powered Data Parallelism

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



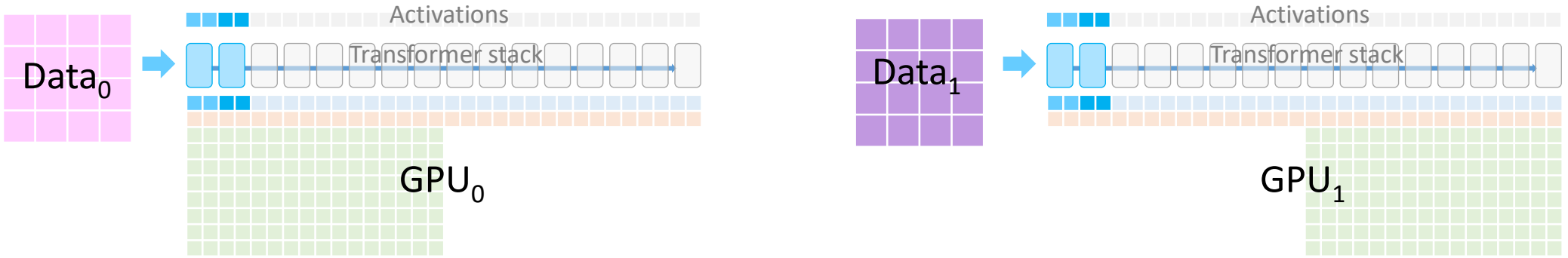

| Memory Consumption | |
|---|---|
| Formulation | Specific Example K=12 Ψ=7.5B $N_d$=64 |
| $(2+2+K)*\Psi$ | 120GB |
| $2\Psi+2\Psi+\frac{K*\Psi}{N_d}$ | 31.4GB |
| $2\Psi+\frac{(2+K)*\Psi}{N_d}$ | 16.6GB |
| $\frac{(2+2+K)*\Psi}{N_d}$ | 1.9GB |

# ZeRO-DP: ZeRO powered Data Parallelism



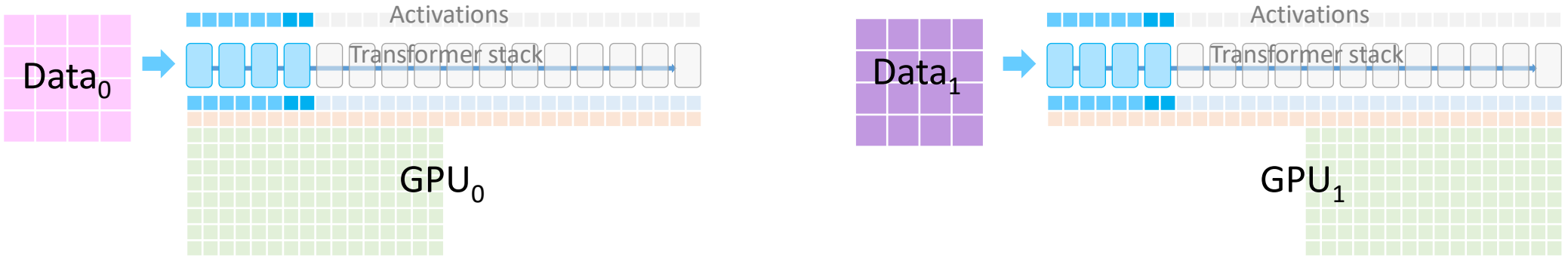

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks

# ZeRO-DP: ZeRO powered Data Parallelism



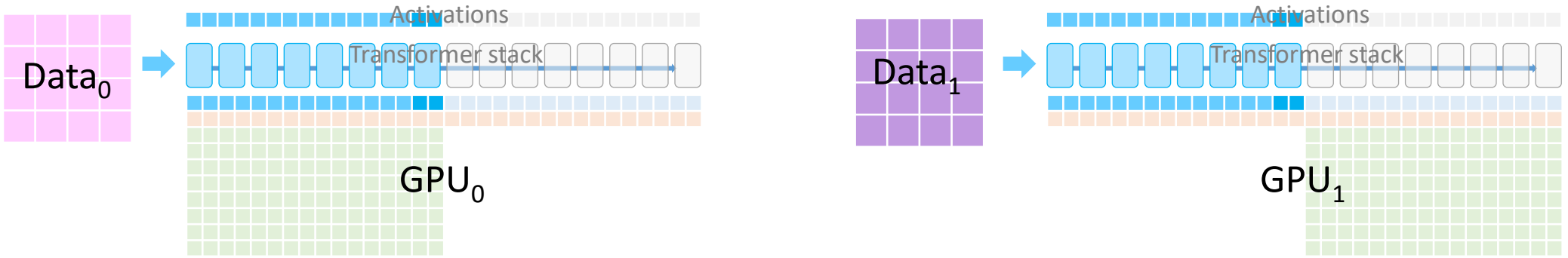

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks

# ZeRO-DP: ZeRO powered Data Parallelism



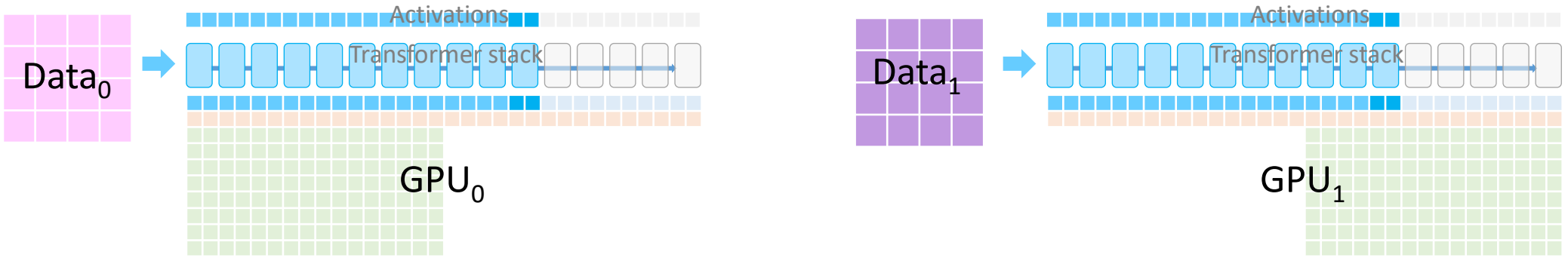

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks

# ZeRO-DP: ZeRO powered Data Parallelism



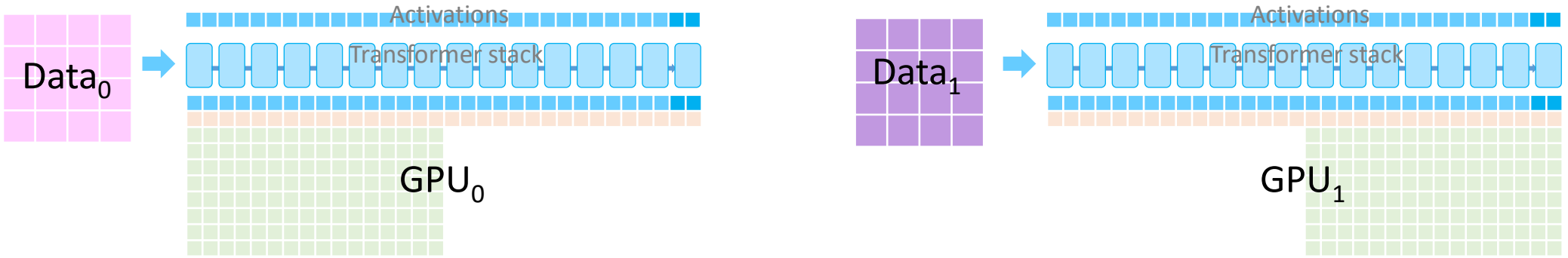

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks

# ZeRO-DP: ZeRO powered Data Parallelism



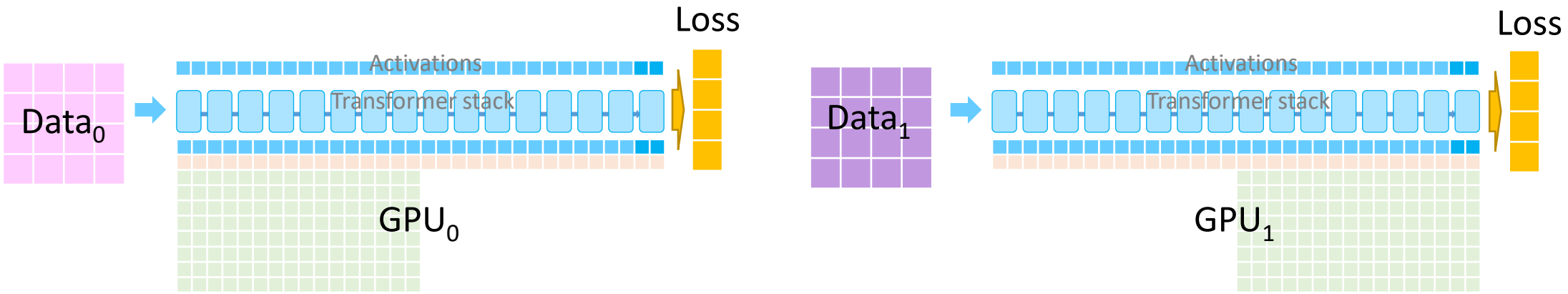

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks

# ZeRO-DP: ZeRO powered Data Parallelism



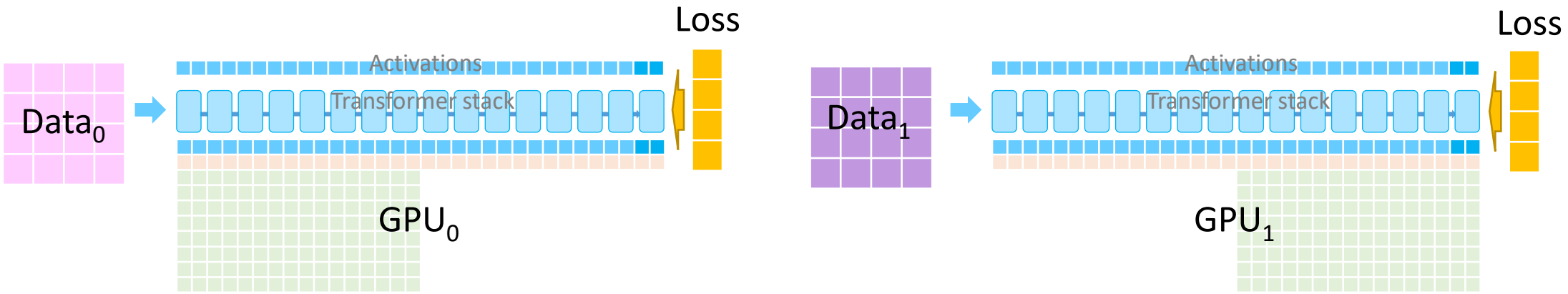

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks

# ZeRO-DP: ZeRO powered Data Parallelism



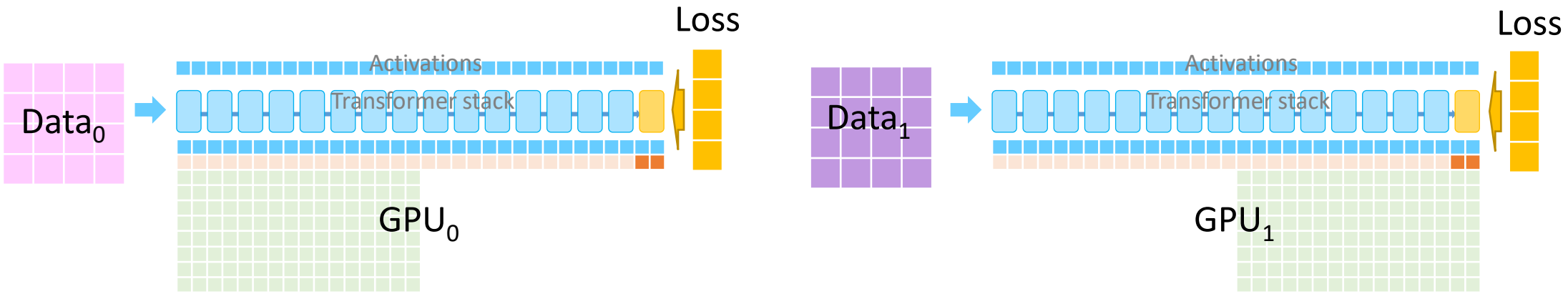

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks

# ZeRO-DP: ZeRO powered Data Parallelism



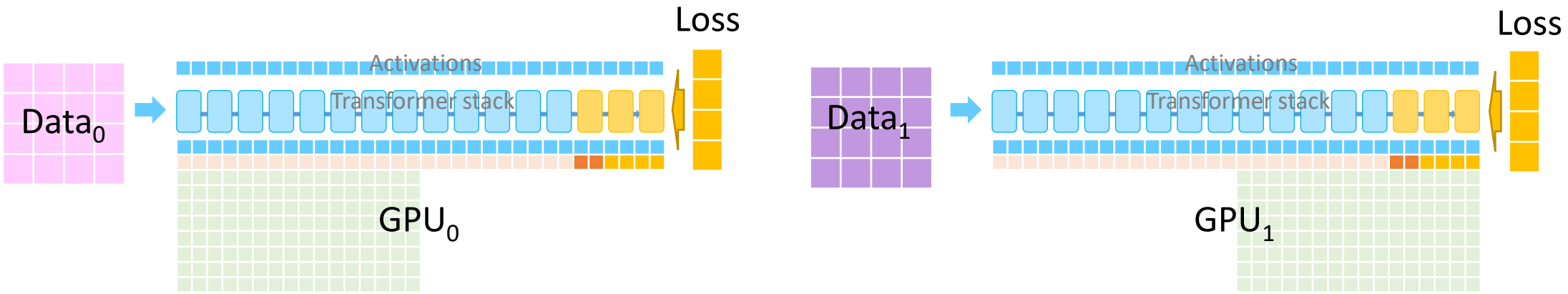

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks
- Backward propagation to generate FP16 gradients

# ZeRO-DP: ZeRO powered Data Parallelism



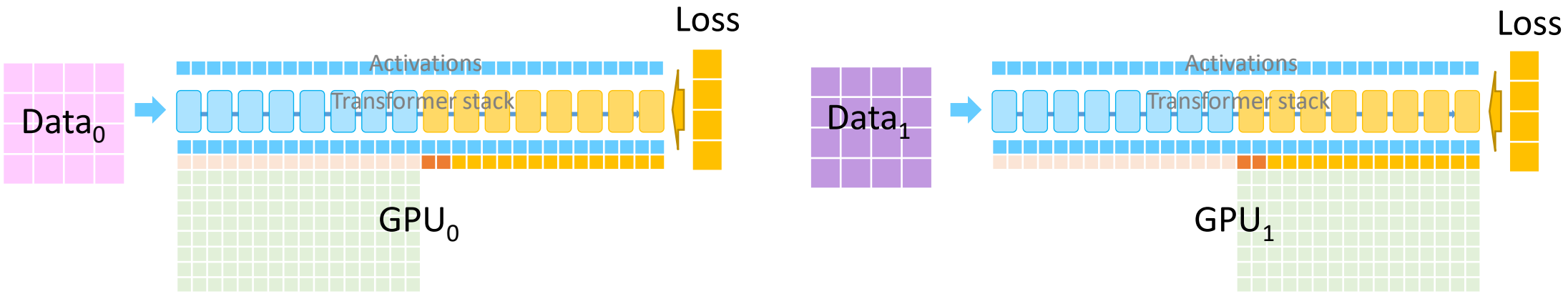

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks
- Backward propagation to generate FP16 gradients

# ZeRO-DP: ZeRO powered Data Parallelism



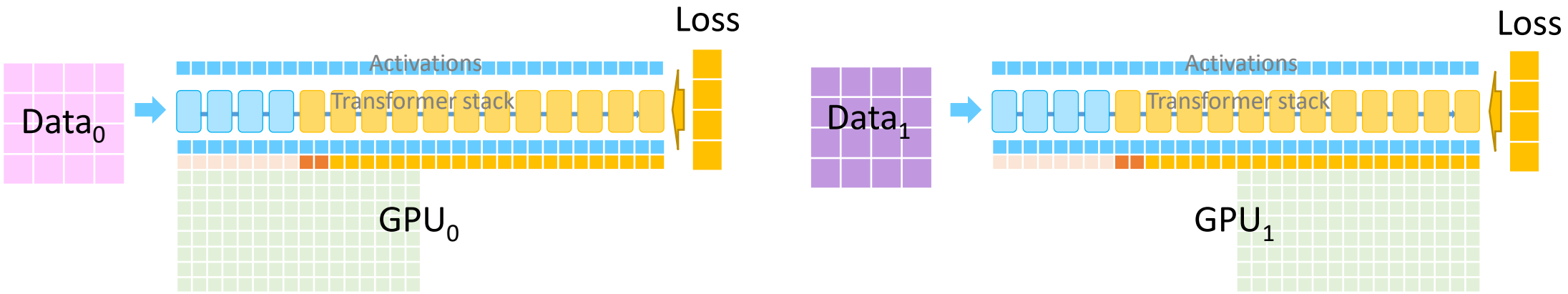

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks
- Backward propagation to generate FP16 gradients

# ZeRO-DP: ZeRO powered Data Parallelism



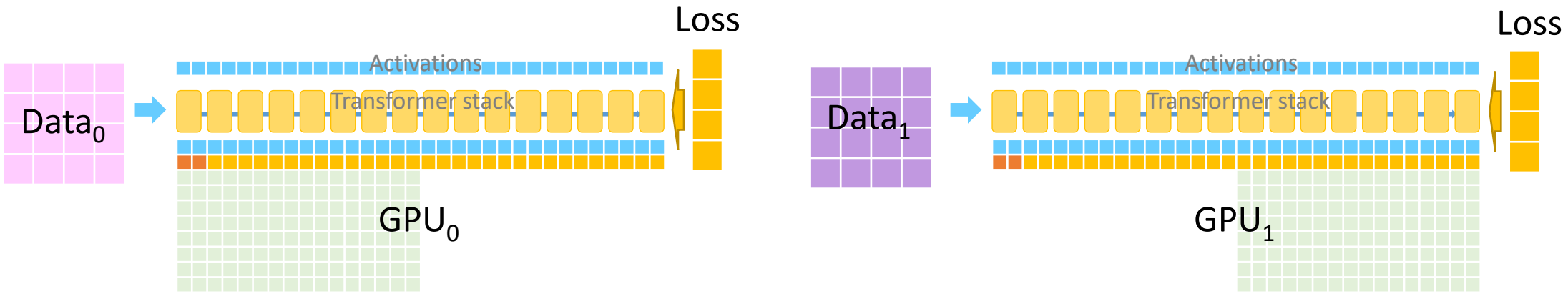

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks
- Backward propagation to generate FP16 gradients

# ZeRO-DP: ZeRO powered Data Parallelism



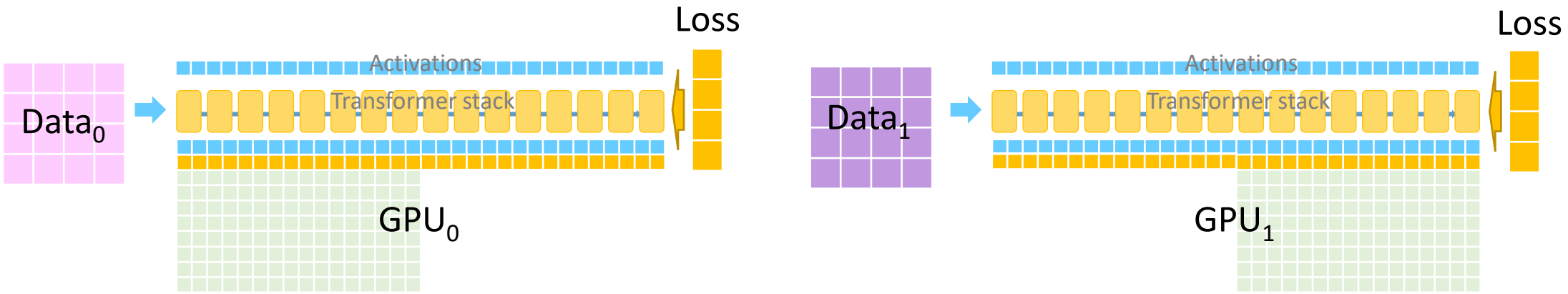

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks
- Backward propagation to generate FP16 gradients

# ZeRO-DP: ZeRO powered Data Parallelism



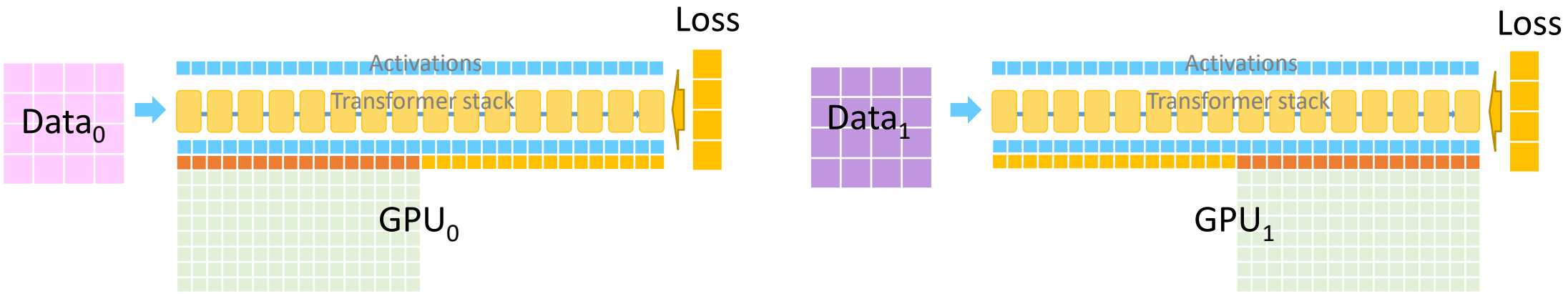

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks
- Backward propagation to generate FP16 gradients

# ZeRO-DP: ZeRO powered Data Parallelism



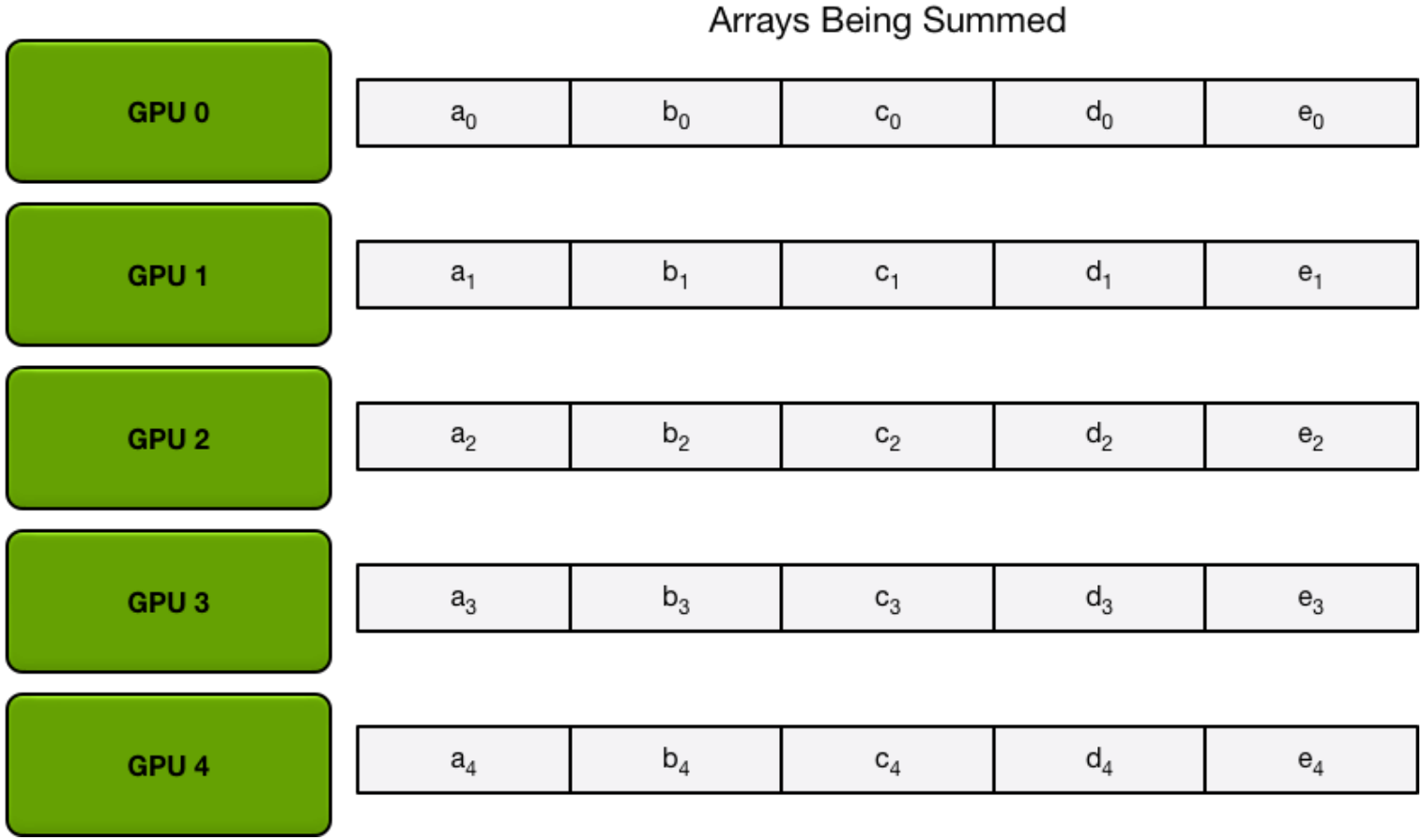

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks
- Backward propagation to generate FP16 gradients and reduce scatter to average

# ZeRO-DP: ZeRO powered Data Parallelism



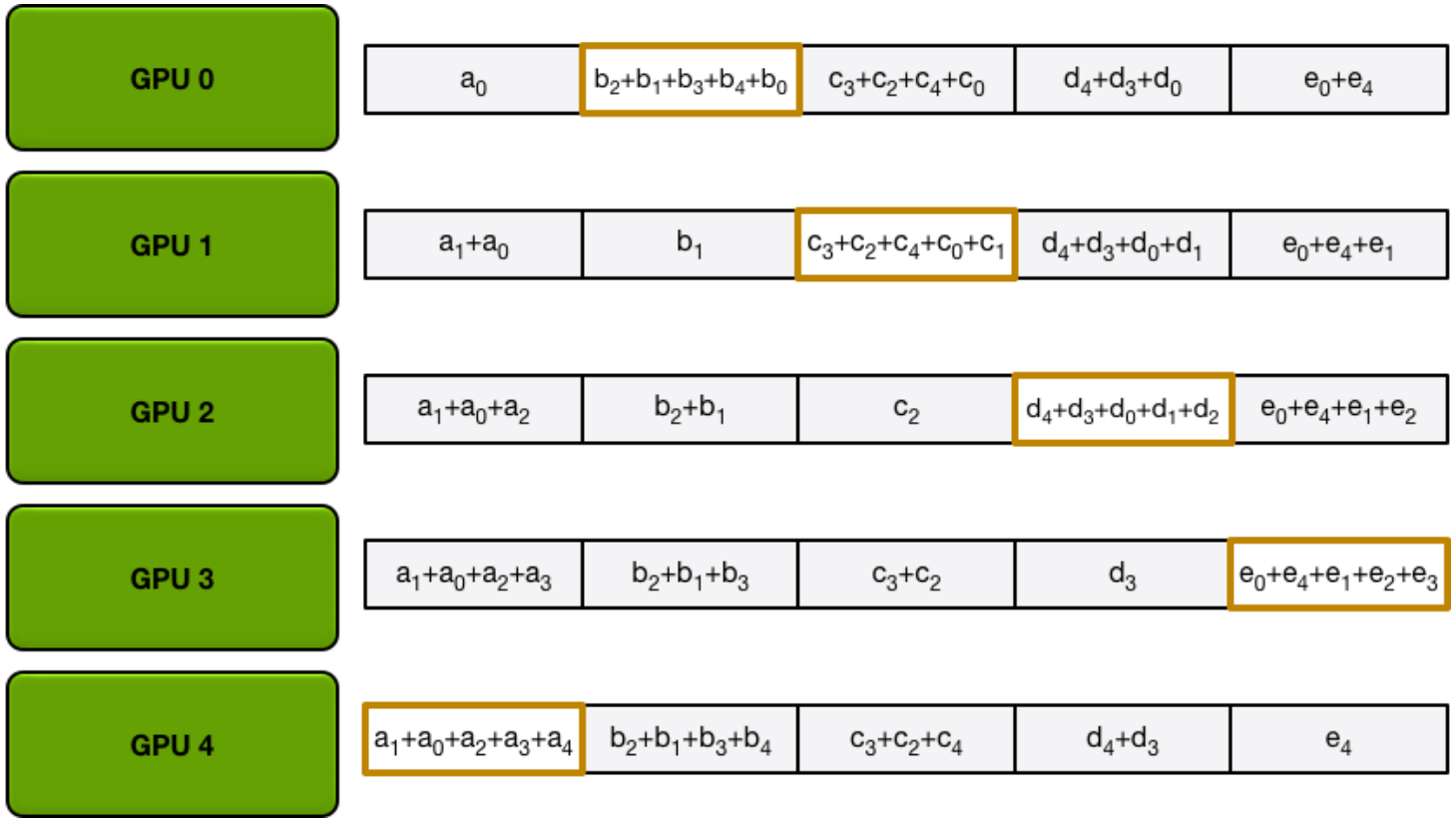

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks
- Backward propagation to generate FP16 gradients and reduce scatter to average

# Recall: Reduce-Scatter



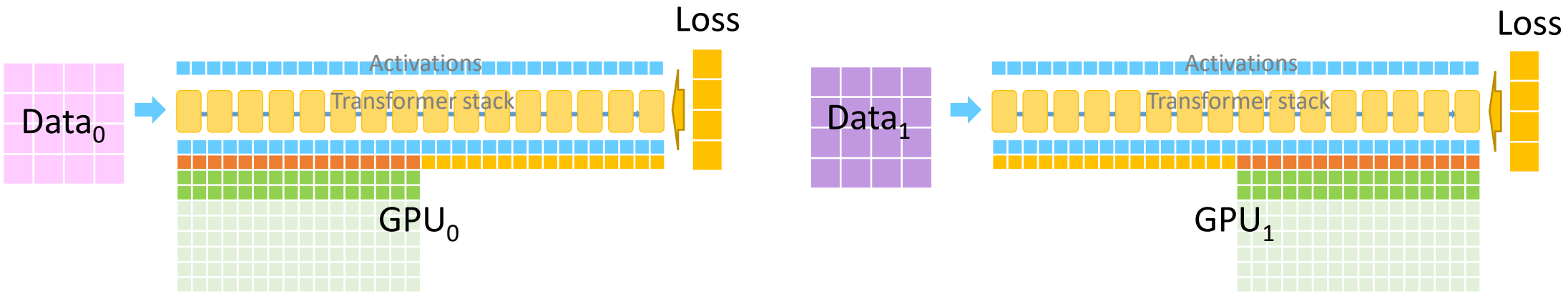

Partitioning of an array into N chunks

# Recall: Reduce-Scatter



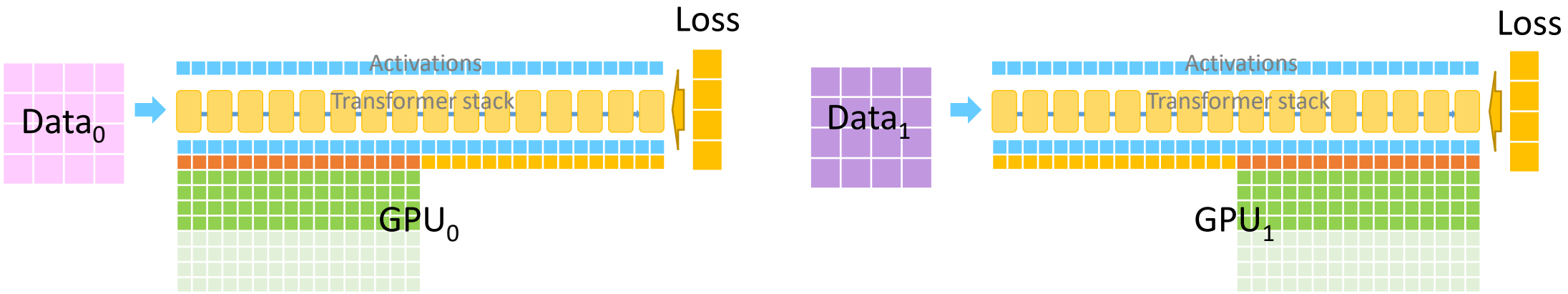

Reduce-scatter data transfer (after iteration 4)

# ZeRO-DP: ZeRO powered Data Parallelism



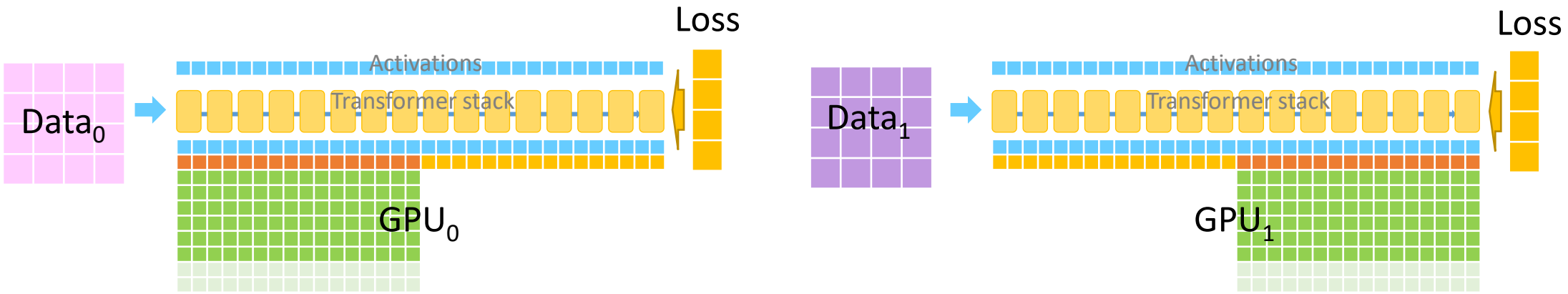

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks
- Backward propagation to generate FP16 gradients and reduce scatter to average
- Update the FP32 weights with ADAM optimizer

# ZeRO-DP: ZeRO powered Data Parallelism



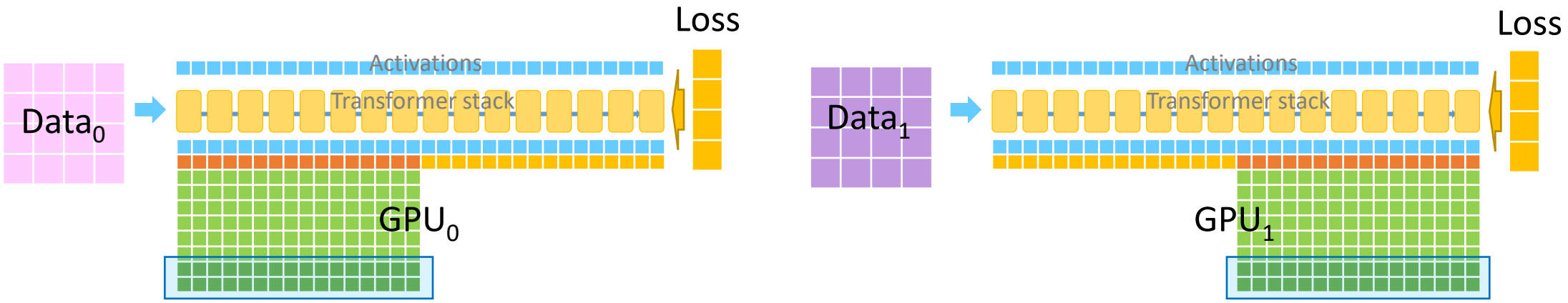

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks
- Backward propagation to generate FP16 gradients and reduce scatter to average
- Update the FP32 weights with ADAM optimizer

# ZeRO-DP: ZeRO powered Data Parallelism



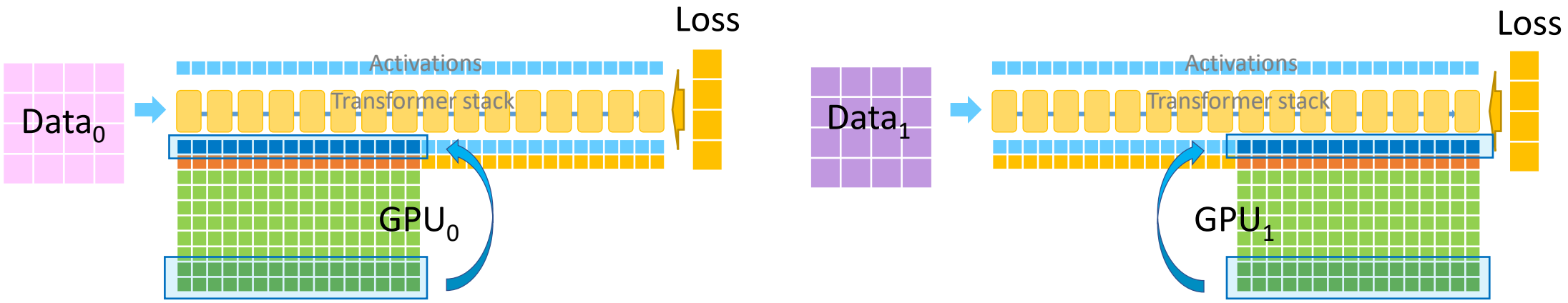

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks
- Backward propagation to generate FP16 gradients and reduce scatter to average
- Update the FP32 weights with ADAM optimizer

# ZeRO-DP: ZeRO powered Data Parallelism



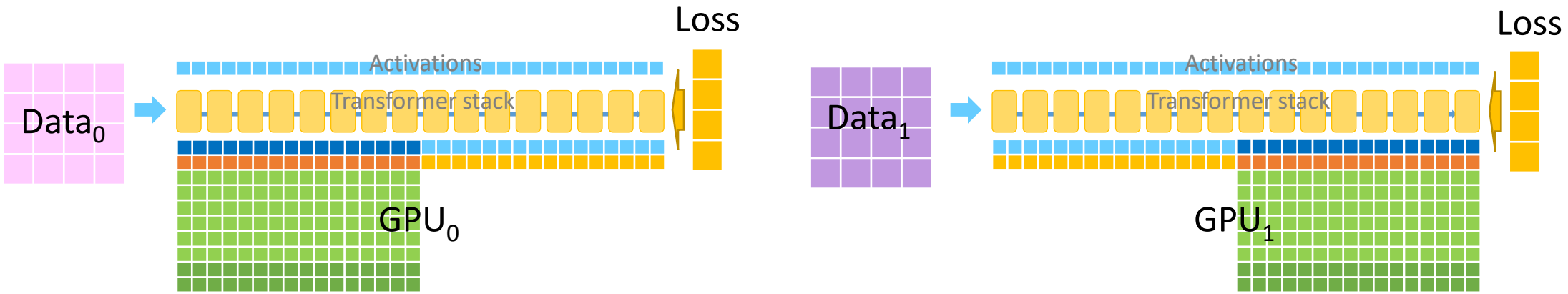

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks
- Backward propagation to generate FP16 gradients and reduce scatter to average
- Update the FP32 weights with ADAM optimizer

# ZeRO-DP: ZeRO powered Data Parallelism



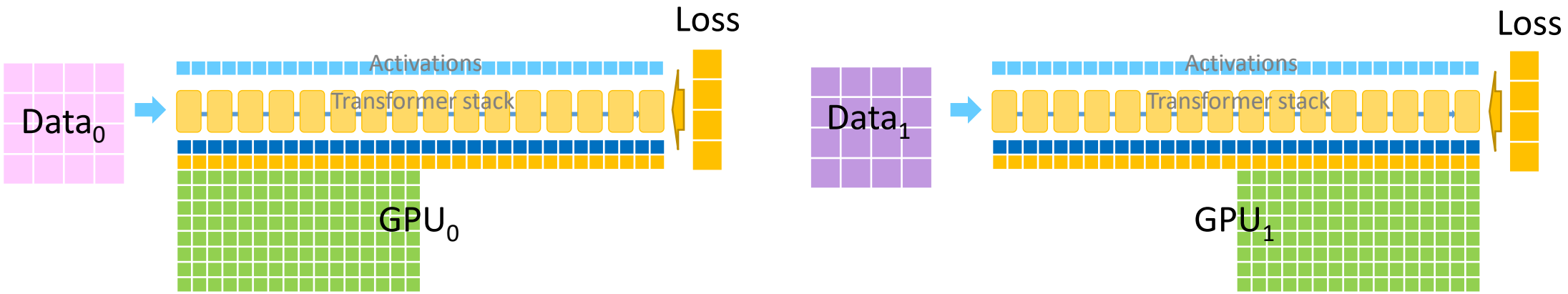

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks
- Backward propagation to generate FP16 gradients and reduce scatter to average
- Update the FP32 weights with ADAM optimizer
- Update the FP16 weights

# ZeRO-DP: ZeRO powered Data Parallelism



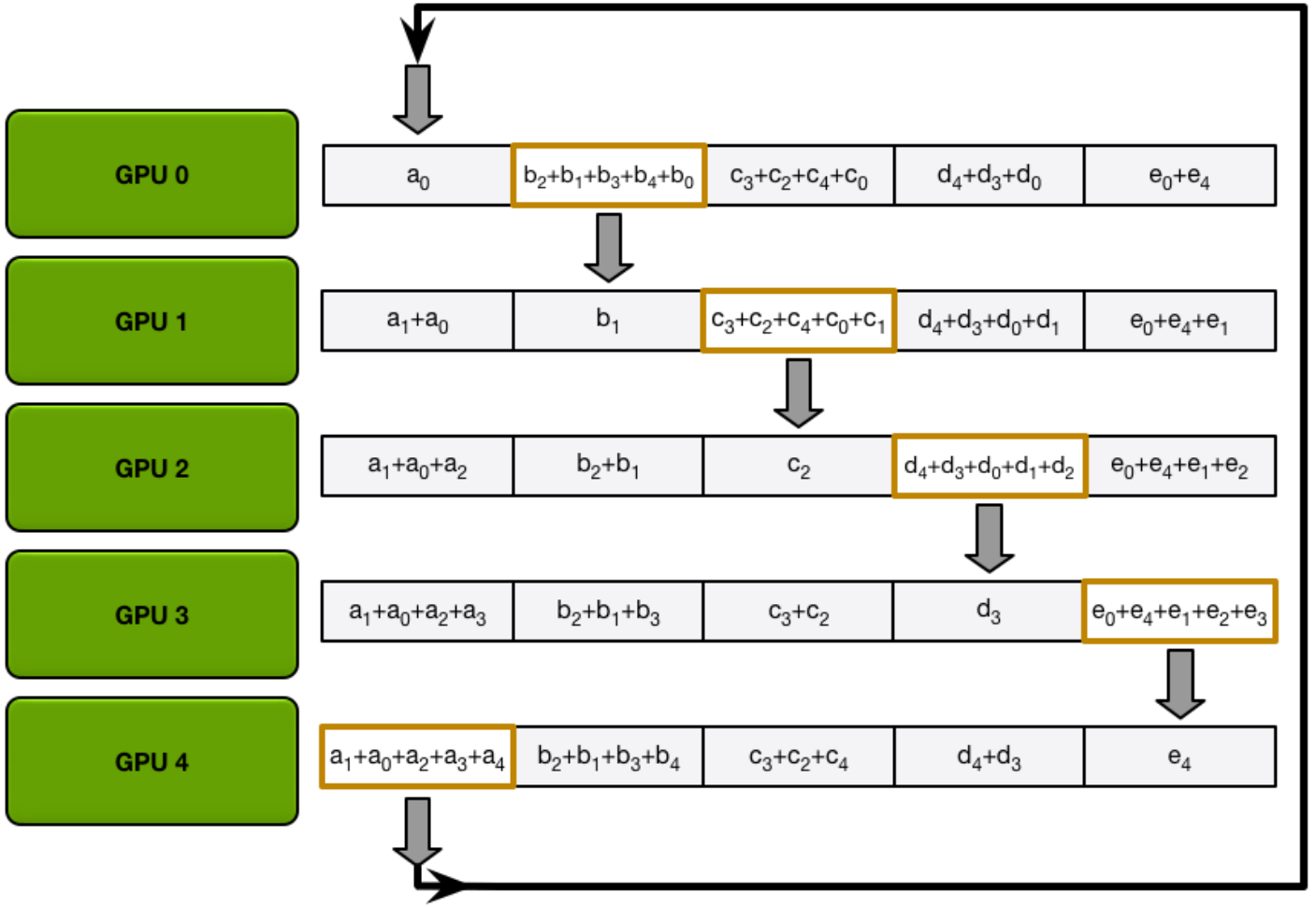

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks
- Backward propagation to generate FP16 gradients and reduce scatter to average
- Update the FP32 weights with ADAM optimizer
- Update the FP16 weights
- All Gather the FP16 weights to complete the iteration

# ZeRO-DP: ZeRO powered Data Parallelism



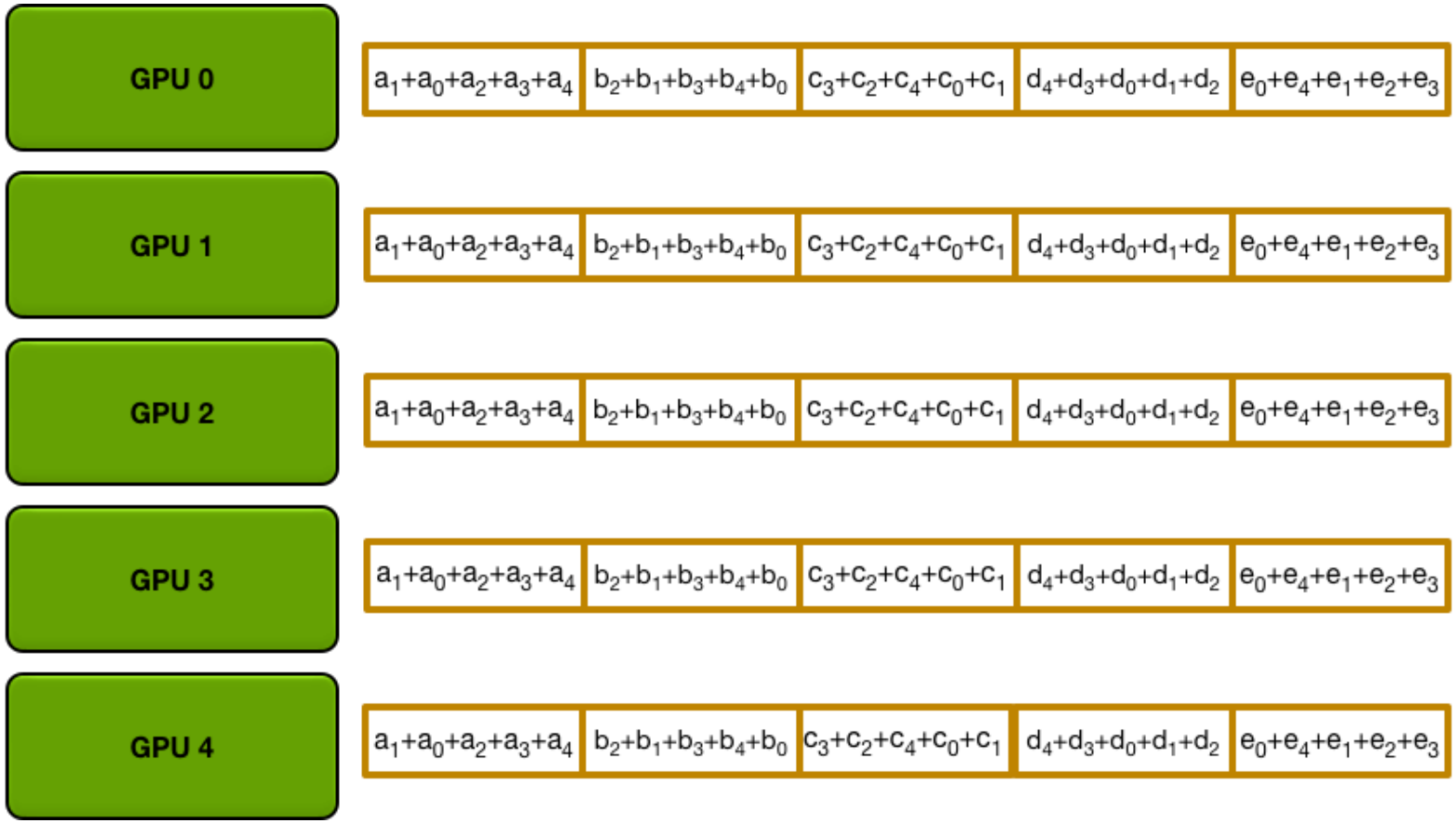

- ZeRO Stage 1
- Partitions optimizer states across GPUs
- Run Forward across the transformer blocks
- Backward propagation to generate FP16 gradients and reduce scatter to average
- Update the FP32 weights with ADAM optimizer
- Update the FP16 weights
- All Gather the FP16 weights to complete the iteration

# Recall: Allgather



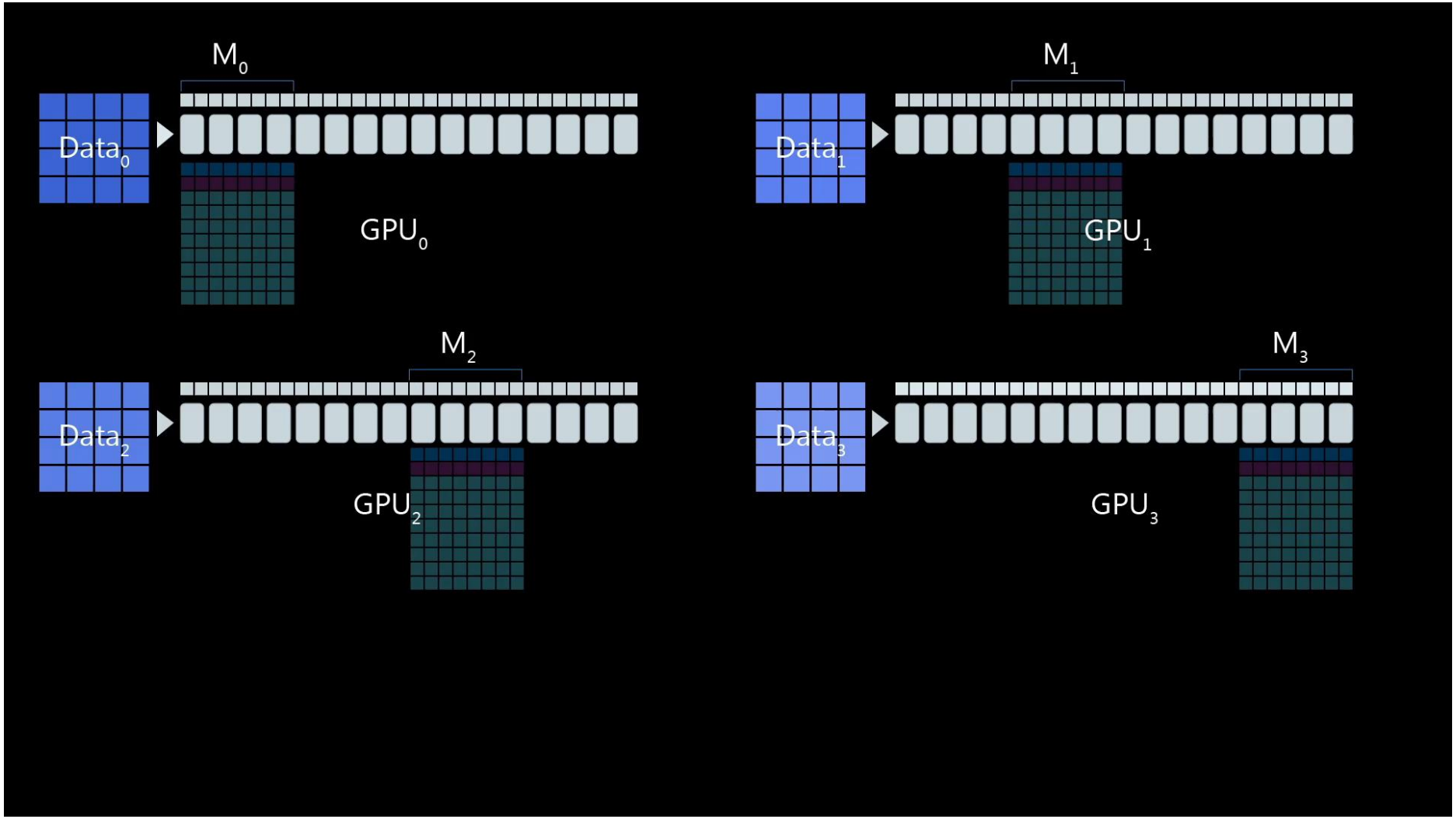

All processes can obtain the complete array by circulating again without reduction operations.

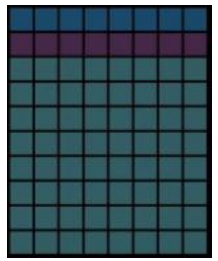
# Recall: Allgather




Final state after allgather transfer

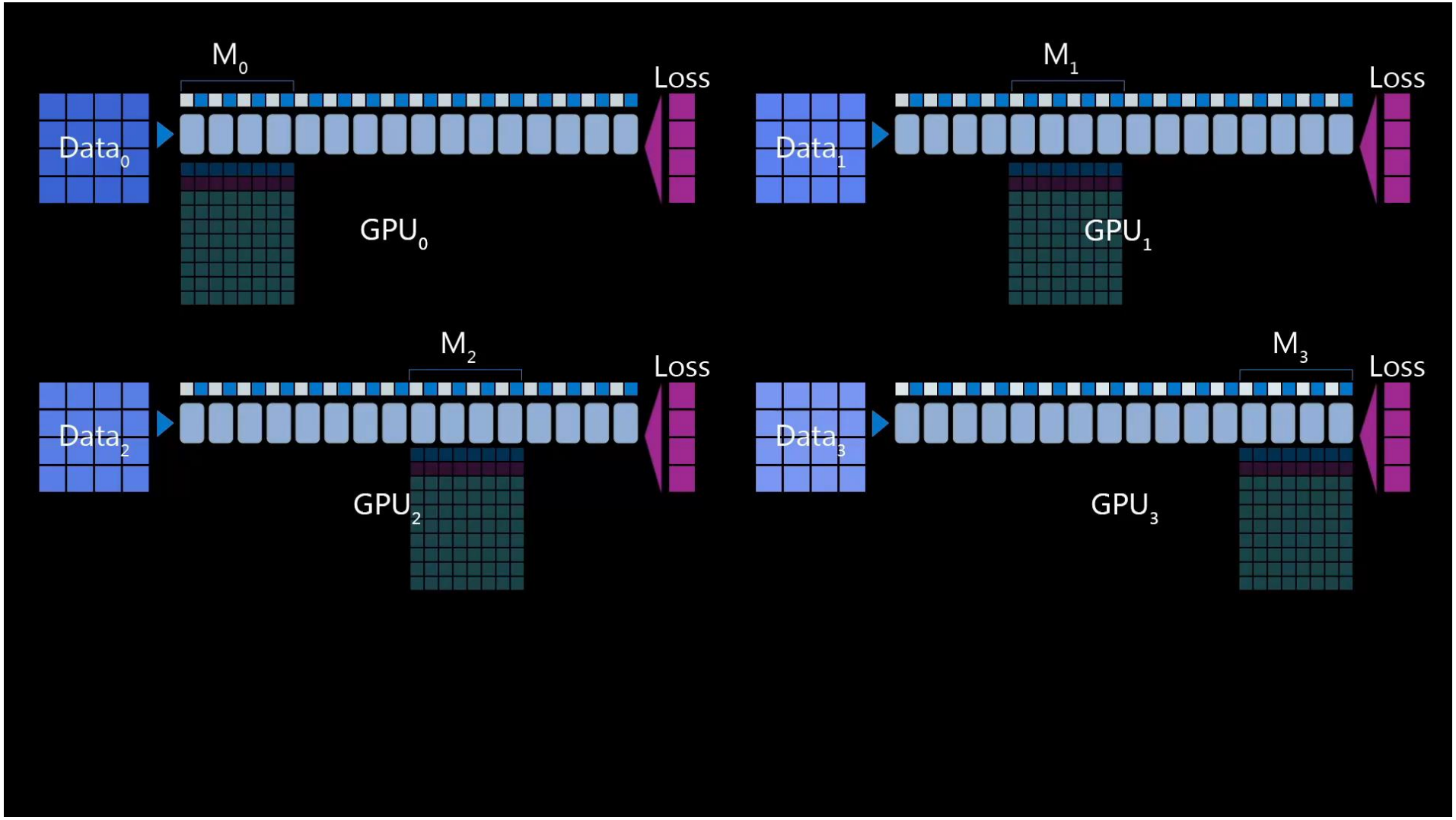
# ZeRO-DP : Stage 3 Forward Propagation

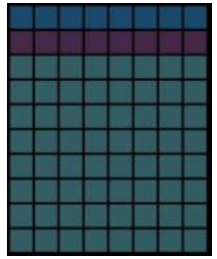


fp16 params
fp16 grads
fp32 grads
fp32 momentum
fp32 variance
fp32 params

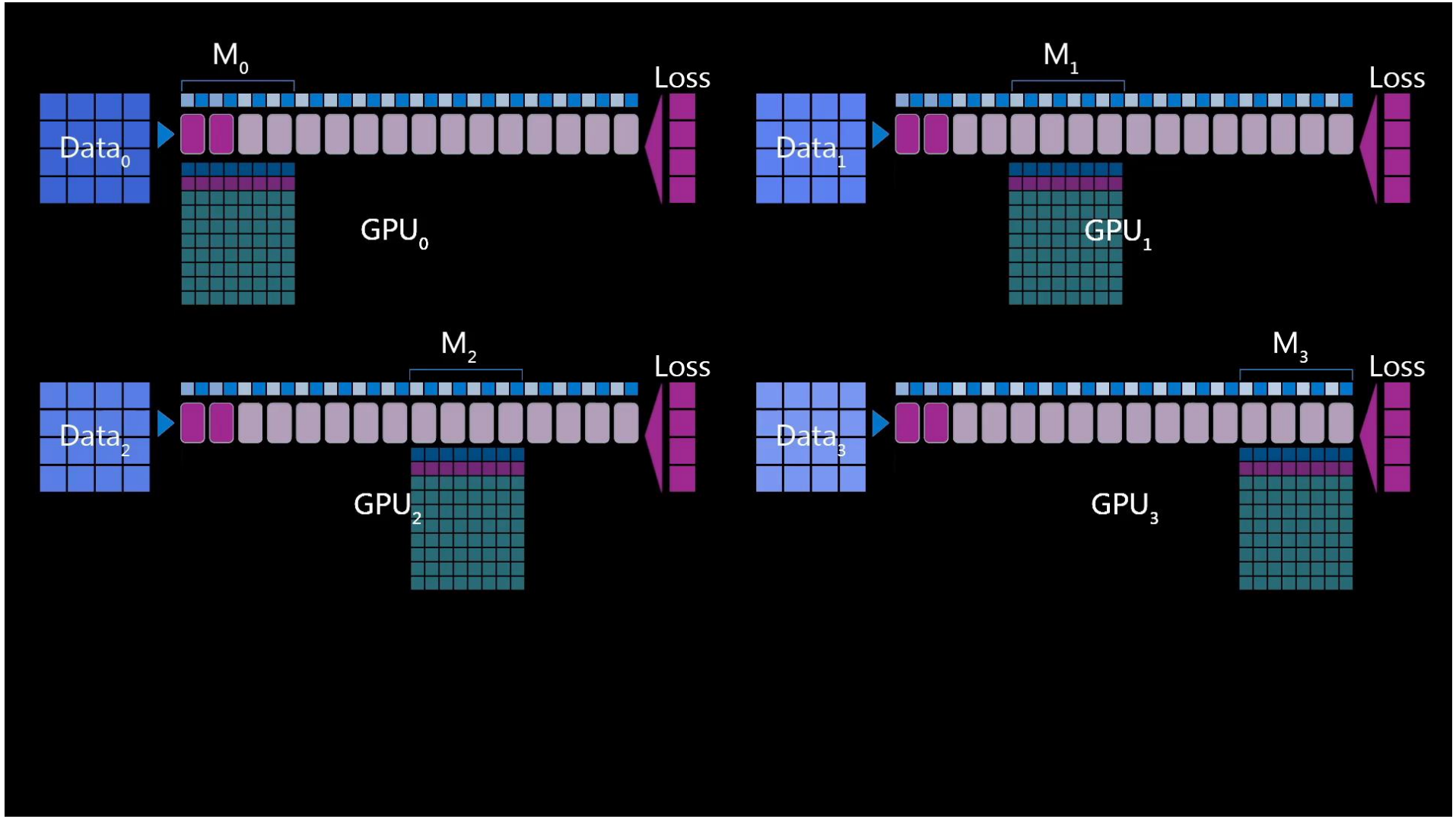
# ZeRO-DP : Stage 3 Backward Propagation

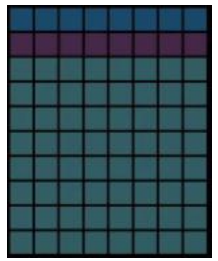


fp16 params
fp16 grads
fp32 grads
fp32 momentum
fp32 variance
fp32 params

# ZeRO-DP : Stage 3 Optimizer Step



fp16 params
fp16 grads
fp32 grads
fp32 momentum
fp32 variance
fp32 params

# ZeRO-DP: Memory Savings

- Progressive memory savings and Communication Volume

| | Memory Reduction with N GPUs |
|---|---|
| Data Parallel | 1x |
| ZeRO Stage 1 ($P_{os}$) | 4x |
| ZeRO Stage 2 ($P_{os+g}$) | 8x |
| ZeRO Stage 3 ($P_{os+g+p}$) | Nx |

*Mixed precision Adam on Cluster of DGX-2 with NVIDIA 32 GB V100 GPUs

# ZeRO-DP: Memory Savings

- Progressive memory savings and Communication Volume

| | Memory Reduction with N GPUs | Max params with ZeRO only (in billions) |
|---|---|---|
| Data Parallel | 1x | 1.2 |
| ZeRO Stage 1 ($P_{os}$) | 4x | 7 |
| ZeRO Stage 2 ($P_{os+g}$) | 8x | 14 |
| ZeRO Stage 3 ($P_{os+g+p}$) | Nx | >1000 |

*Mixed precision Adam on Cluster of DGX-2 with NVIDIA 32 GB V100 GPUs

# ZeRO-DP: Memory Savings

- Progressive memory savings and Communication Volume

| | Memory Reduction with N GPUs | Max params with ZeRO only (in billions) | Max params with ZeRO and model parallelism (in billions) |
|---|---|---|---|
| Data Parallel | 1x | 1.2 | 20 |
| ZeRO Stage 1 ($P_{os}$) | 4x | 7 | 100 |
| ZeRO Stage 2 ($P_{os+g}$) | 8x | 14 | 200 |
| ZeRO Stage 3 ($P_{os+g+p}$) | Nx | >1000 | >1000 |

*Mixed precision Adam on Cluster of DGX-2 with NVIDIA 32 GB V100 GPUs

# ZeRO-DP: Communication Volume

- Progressive memory savings and Communication Volume

| | Memory Reduction with N GPUs | Max params with ZeRO only (in billions) | Max params with ZeRO and model parallelism (in billions) | Comm Volume |
|---|---|---|---|---|
| Data Parallel | 1x | 1.2 | 20 | 1x |
| ZeRO Stage 1 ($P_{os}$) | 4x | 7 | 100 | 1x |
| ZeRO Stage 2 ($P_{os+g}$) | 8x | 14 | 200 | 1x |
| ZeRO Stage 3 ($P_{os+g+p}$) | Nx | >1000 | >1000 | 1.5x |

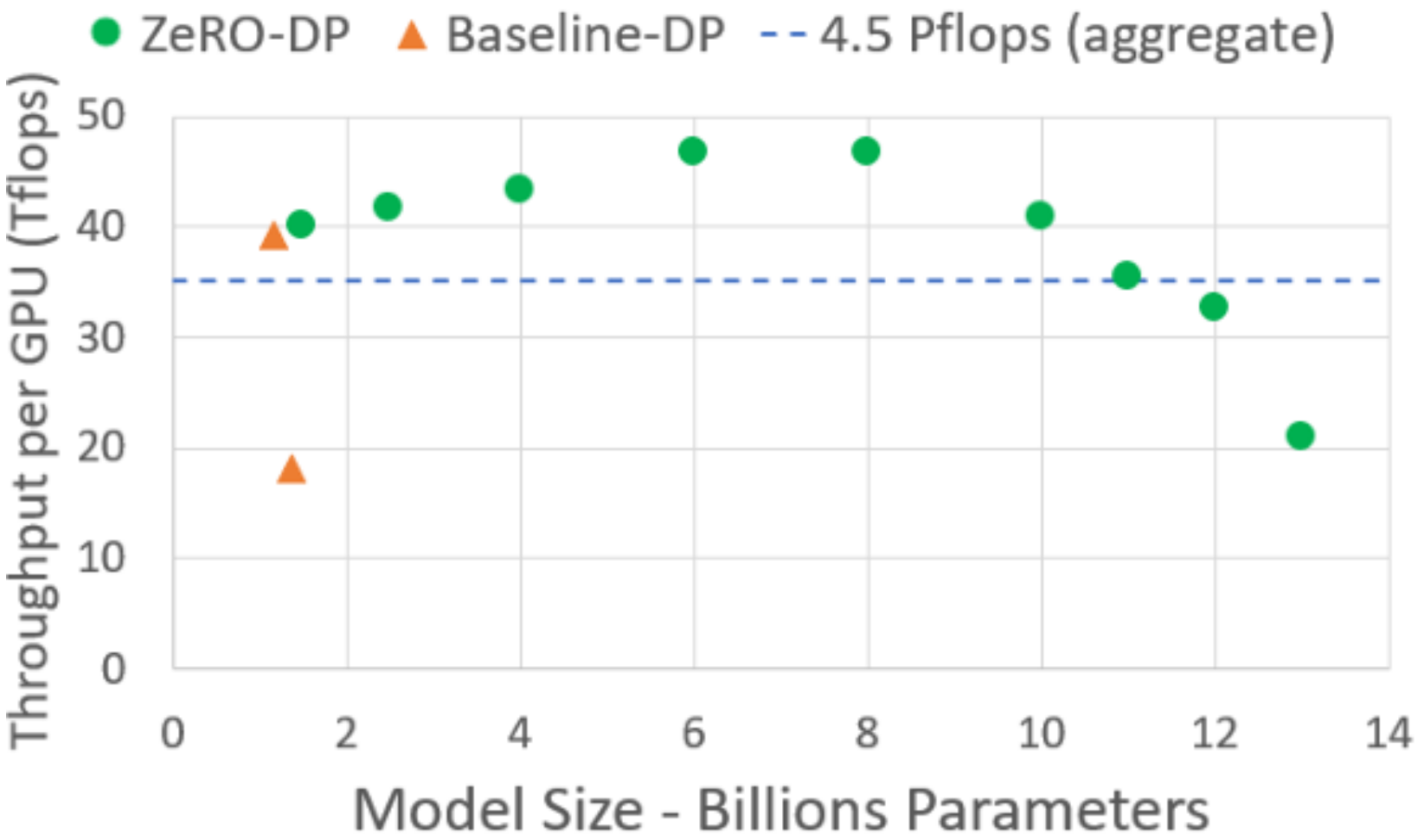
*Mixed precision Adam on Cluster of DGX-2 with NVIDIA 32 GB V100 GPUs

# ZeRO Usability

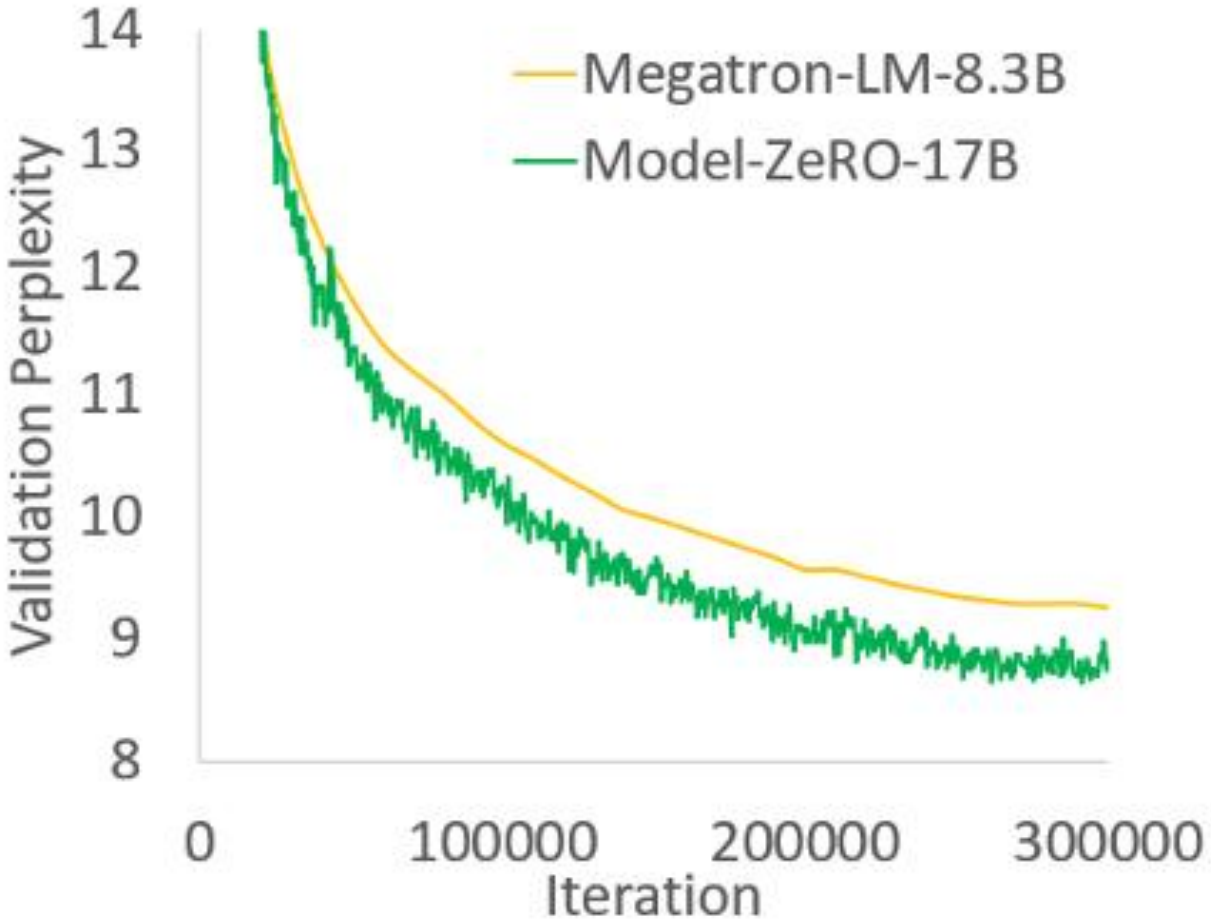
```
# construct torch.nn.Module
model = MyModel()

# wrap w. DeepSpeed engine
engine, *_ = deepspeed.initialize(
    model=model,
    config=ds_config)

# training-loop w.r.t. engine
for batch in data_loader:
  loss = engine(batch)
  engine.backward(loss)
  engine.step()
```

```
ds_config = {
  "optimizer": {
    "type": "Adam",
    "params": {"lr": 0.001}
  },
  "zero": {
    "stage": 3,
    "offload_optimizer": {
        "device": "[cpu|nvme]"
    },
    "offload_param": {
        "device": "[cpu|nvme]"
    }
  }
}
```

# ZeRO Performance: Max Model Throughput



# ZeRO Performance: Turning-NLG 17B



# Logistics

Two guest lectures from industry

- March 10, Meta, Zechun Liu (online)
- March 12, Amazon, Yida Wang (online)

# Questions?