Lancet: Accelerating MoE Training via Whole Graph Computation-Communication Overlapping

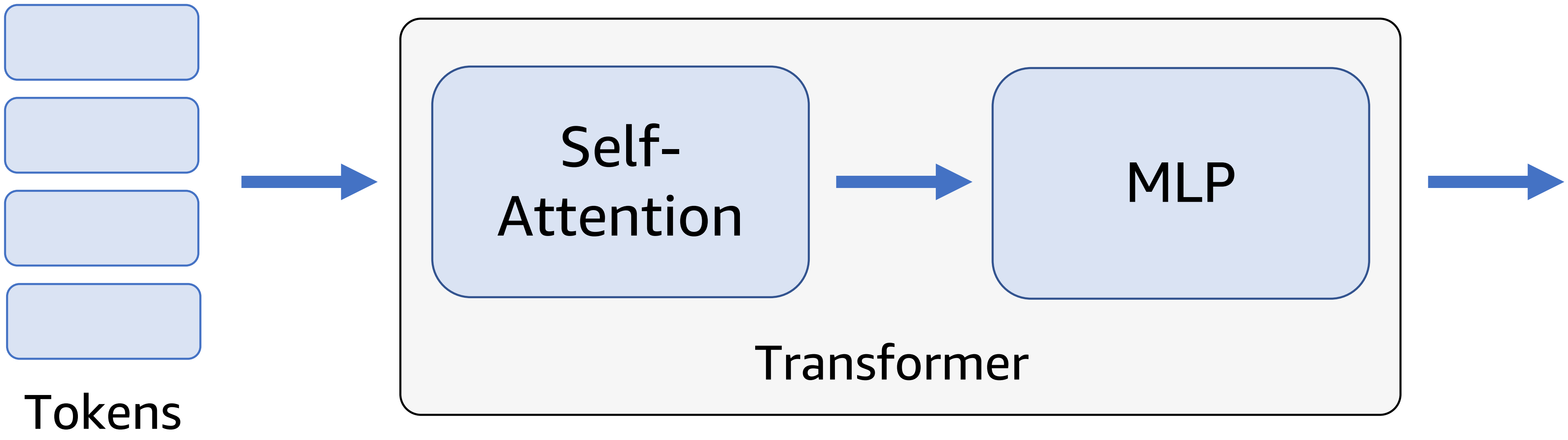
Chenyu Jiang[1*], Ye Tian[1*], Zhen Jia[2], Shuai Zheng[3^], Chuan Wu[1], Yida Wang[2]

[1]The University of Hong Kong, [2]Amazon Web Services, [3]Boson AI

*Work done while interning at AWS. ^Work done while at AWS.

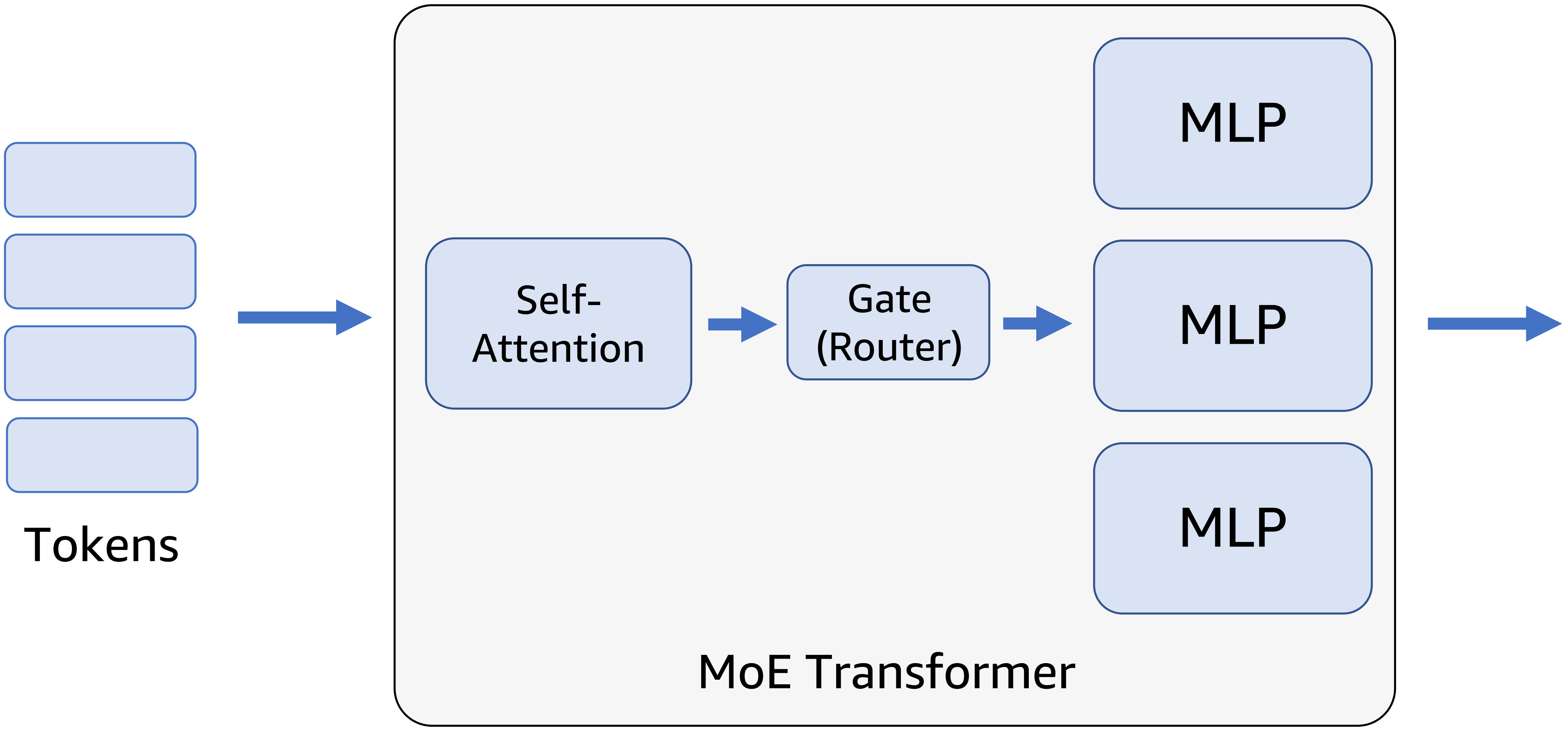
Background

Mixture-of-Experts



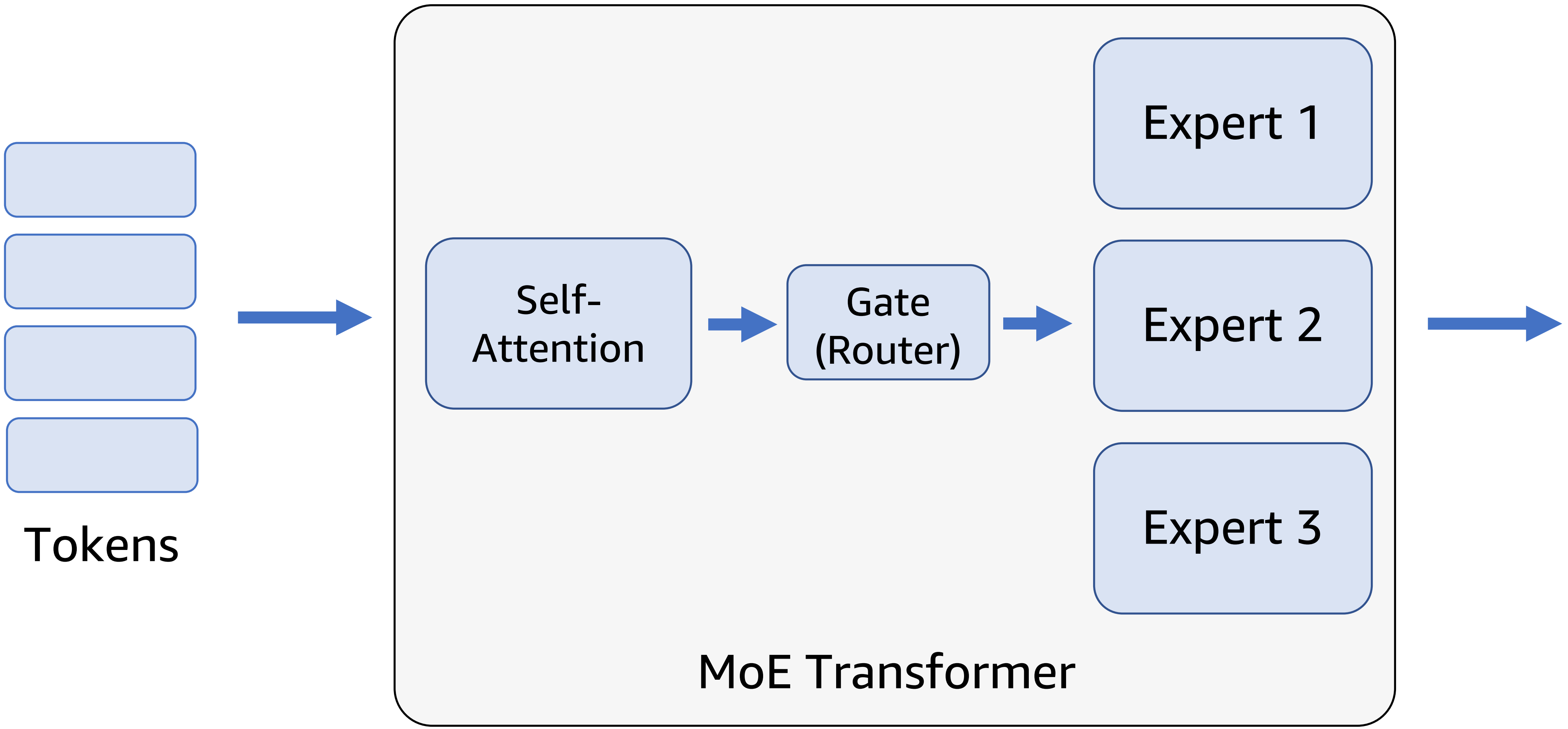
Background

Mixture-of-Experts



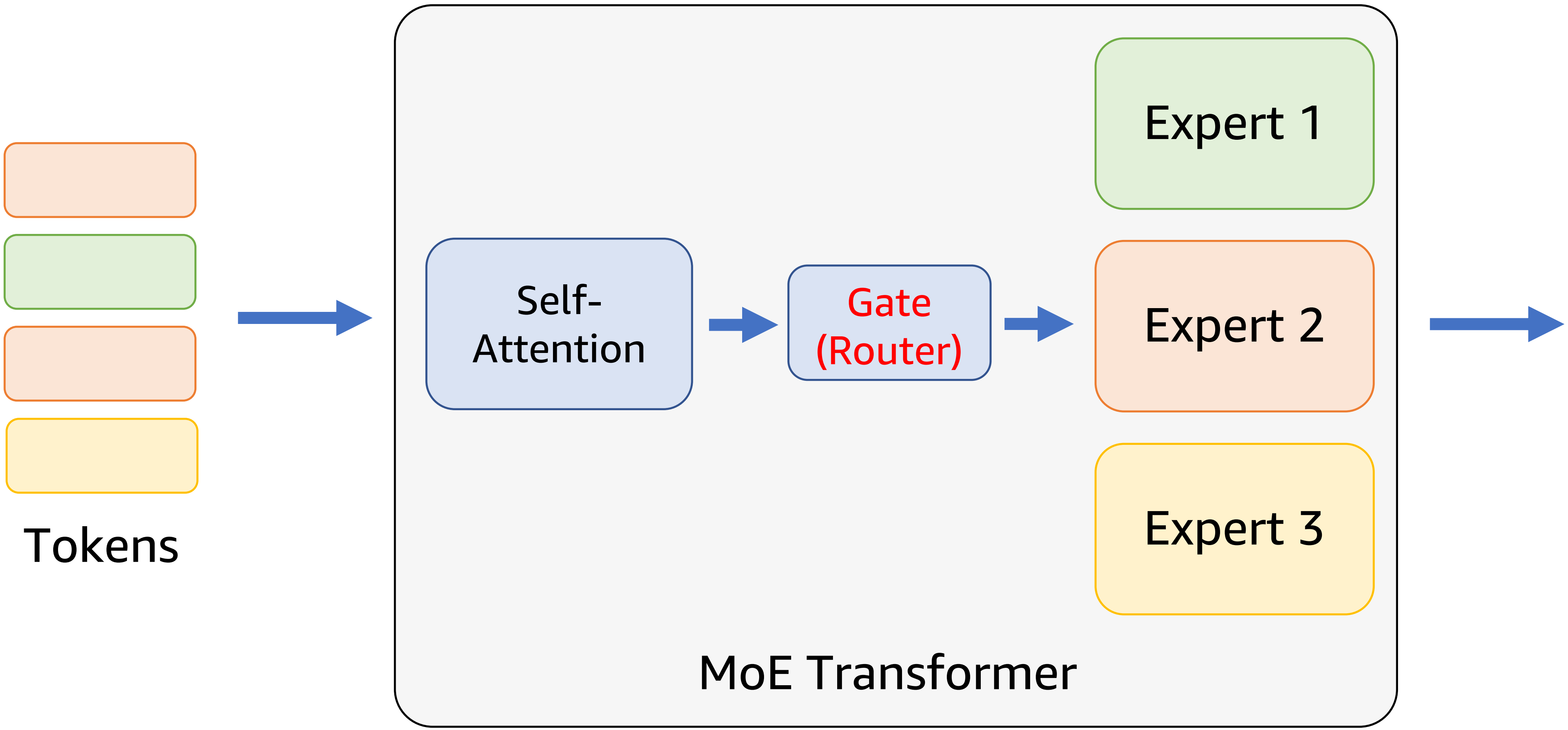
Background

Mixture-of-Experts



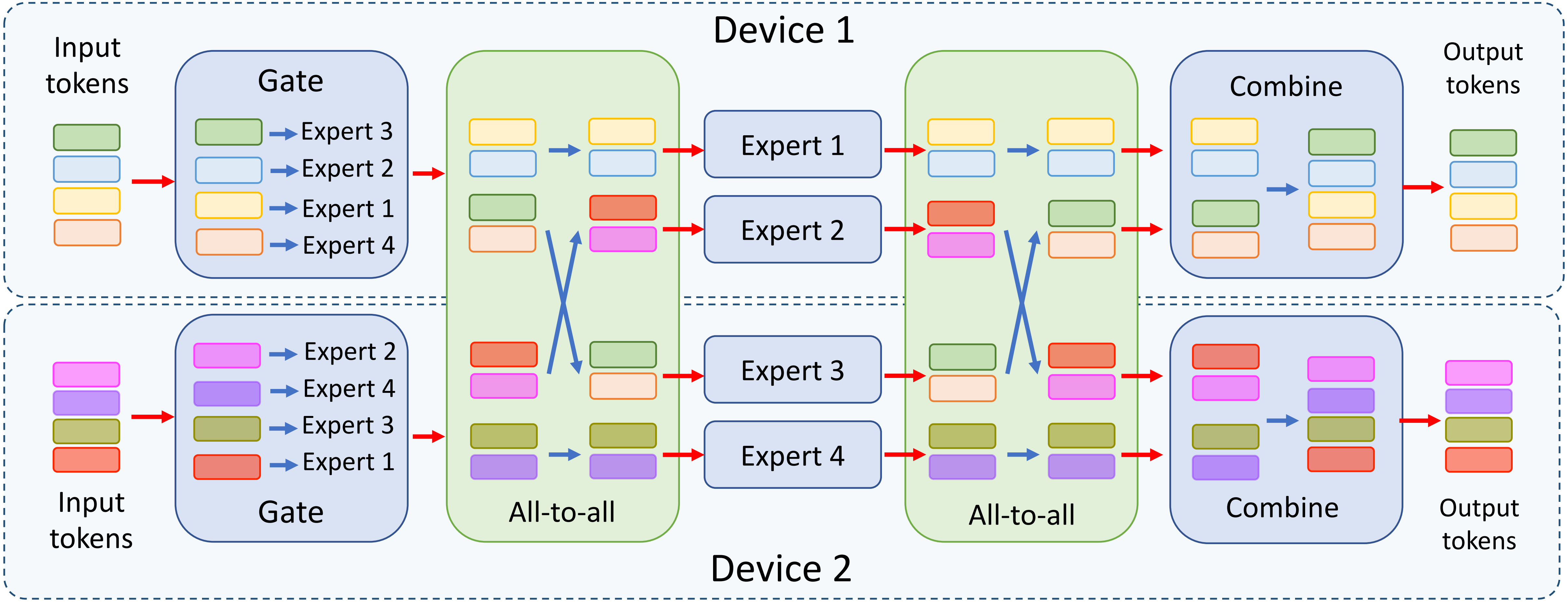
Background

Mixture-of-Experts



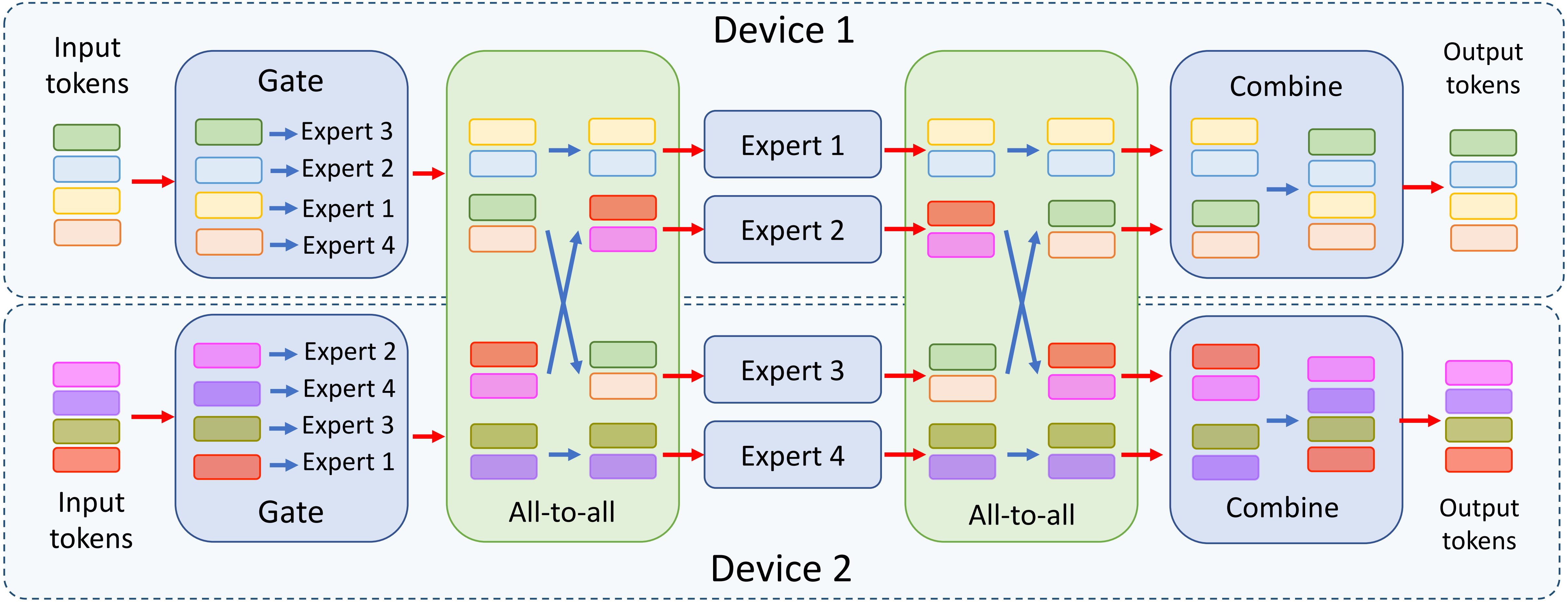
Background

All-To-All Communication in MoE Training



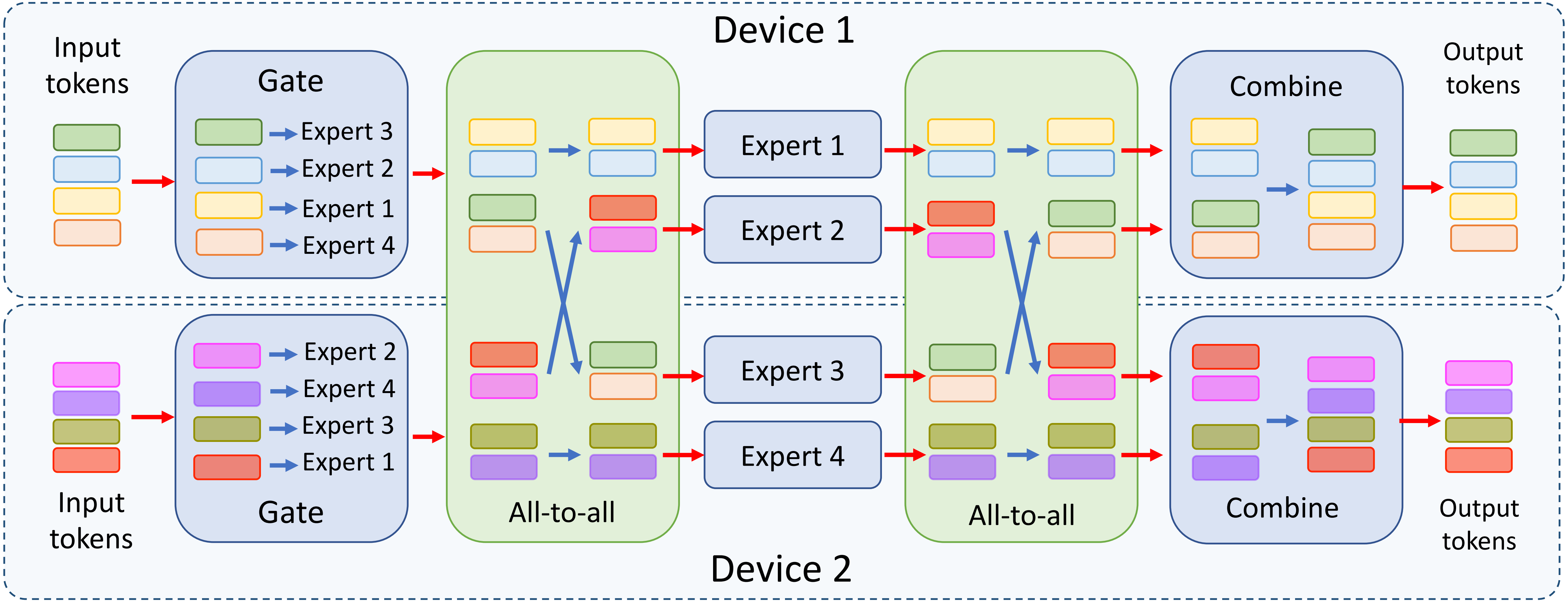
Background

All-To-All Communication in MoE Training



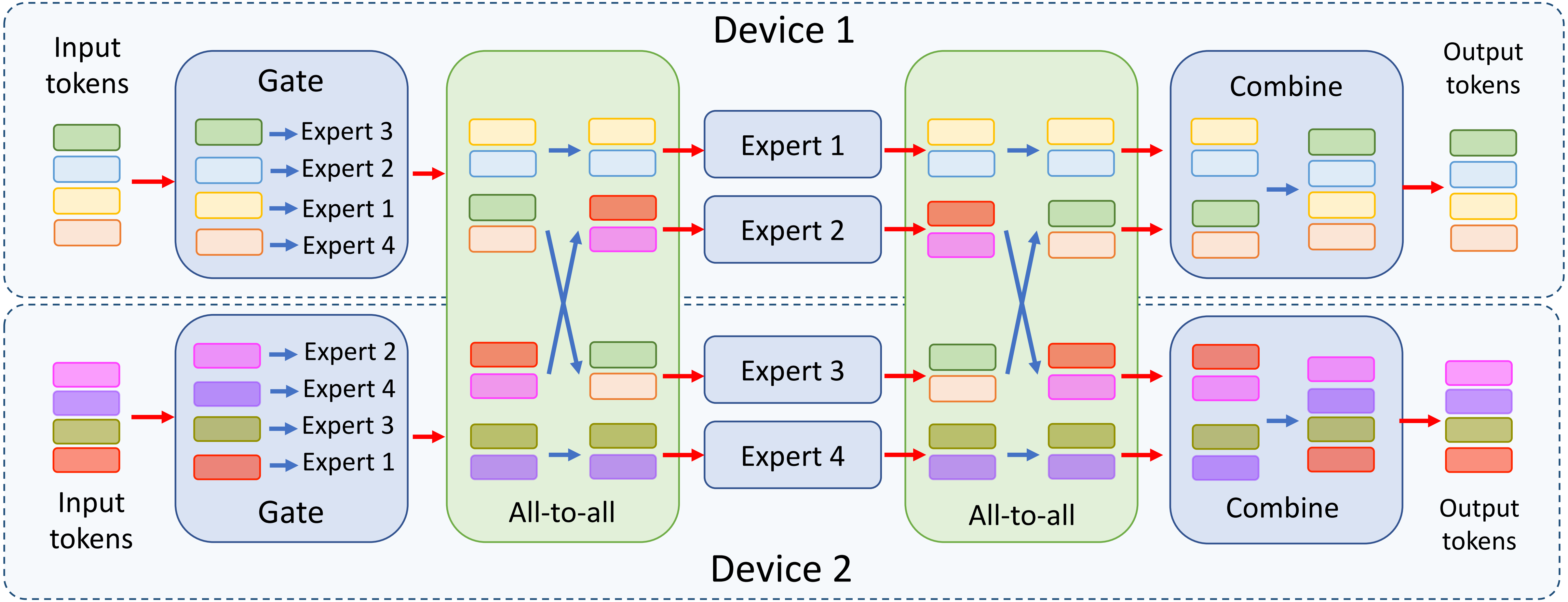
Background

All-To-All Communication in MoE Training



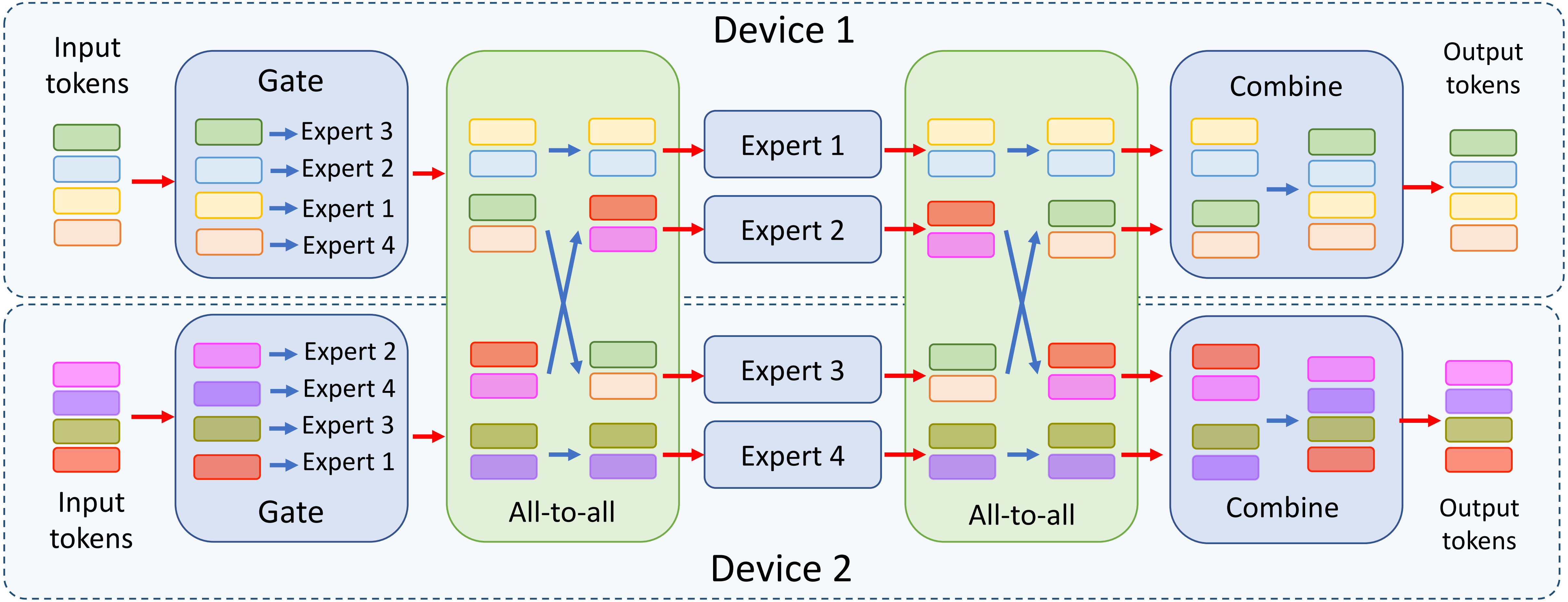
Background

All-To-All Communication in MoE Training



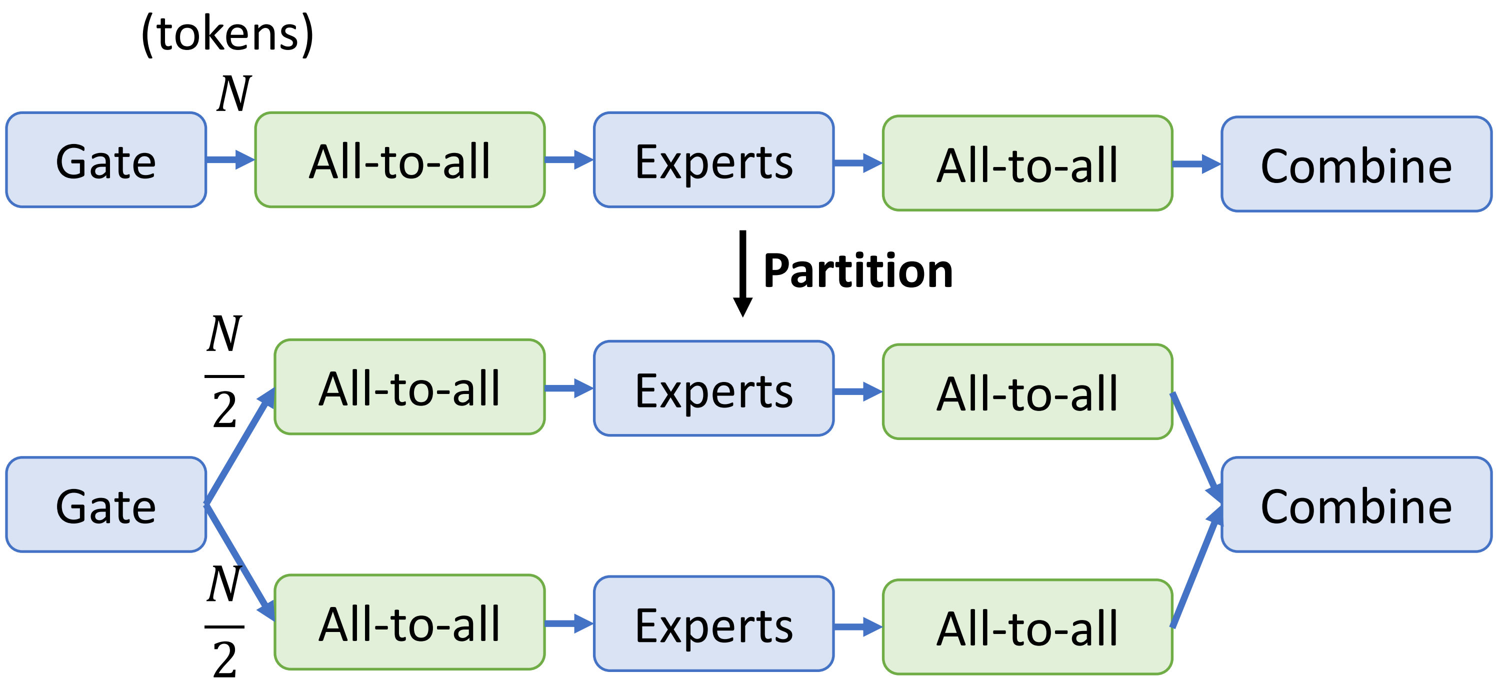
Background

All-To-All Communication in MoE Training



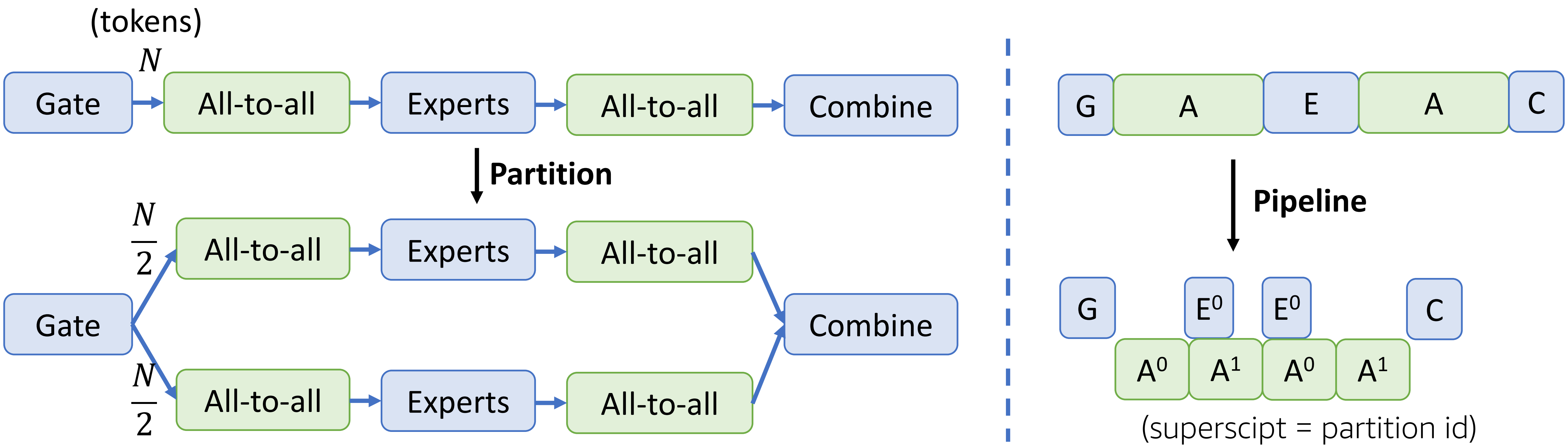
Background

SOTA solution: Overlap All-to-All and expert computation



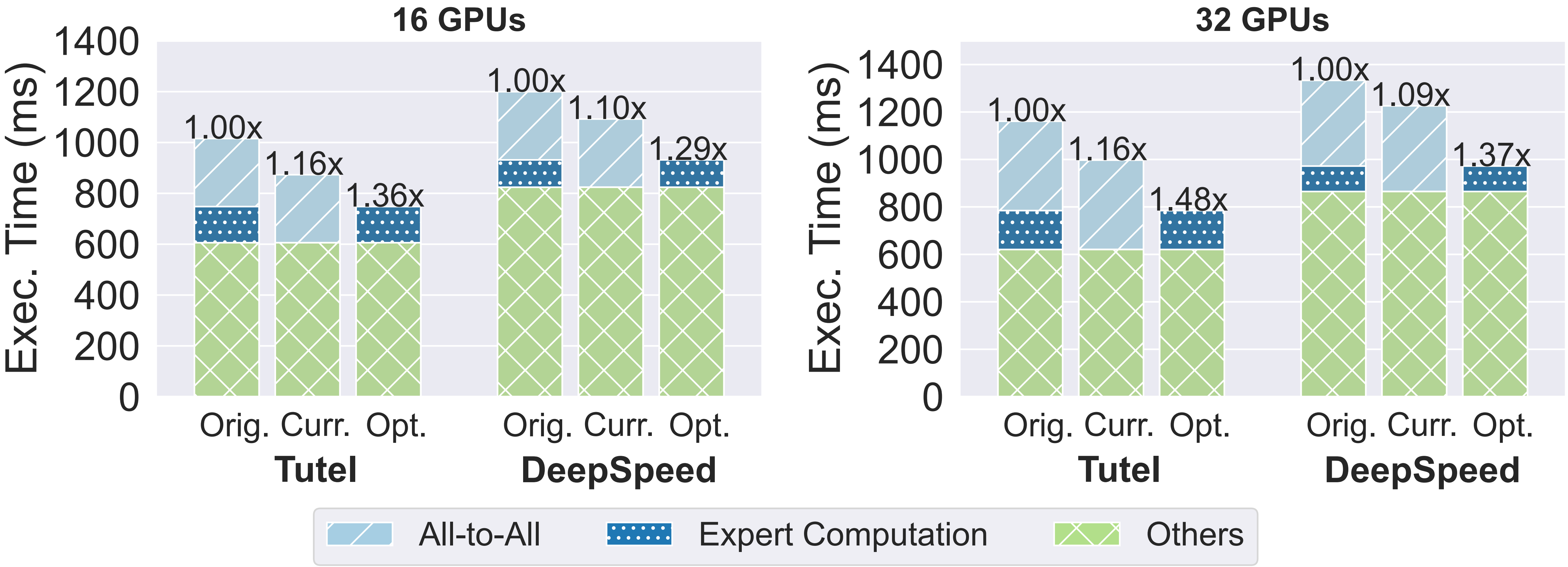
Background

SOTA solution: Overlap All-to-All and expert computation



Background

SOTA solution: Overlap All-to-All and expert computation (cont'd)

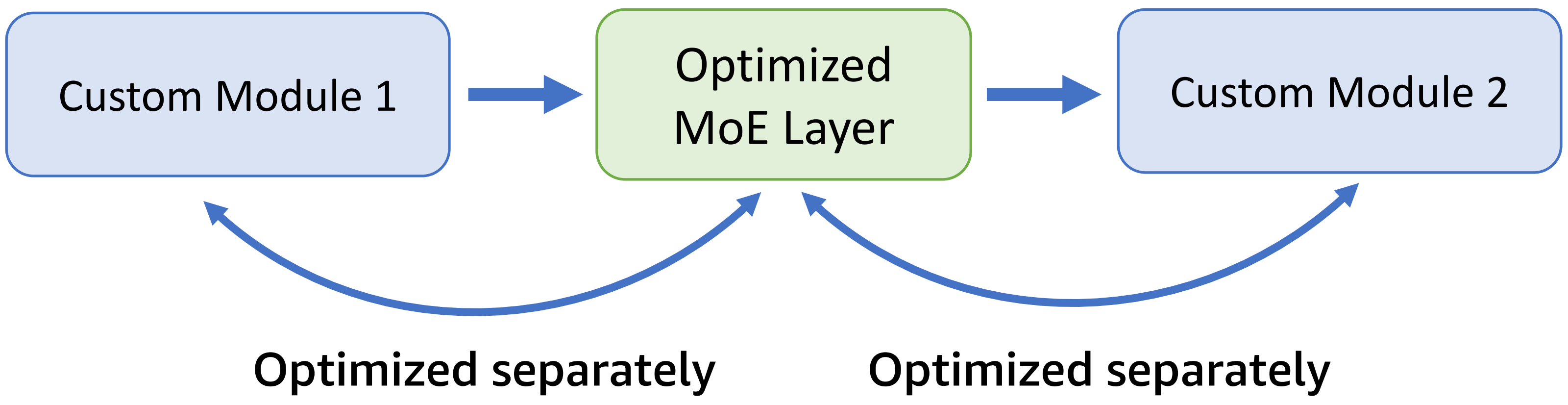


(GPT2-MoE model, two experts per GPU, running on AWS EC2 p4d instances)

Problem: Expert computation unable to fully overlap all-to-all

Our insight

Current methods constraint the scope of optimization within the MoE layer.

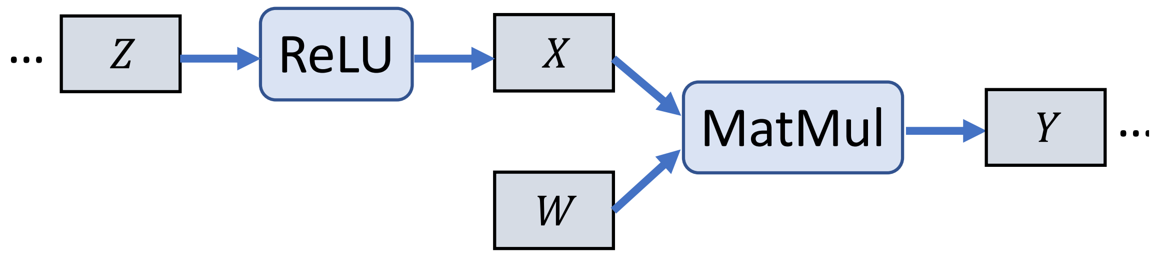


What if we consider the optimization opportunities at the whole model level?

Extend the scope of overlapping

Opportunity 1: Weight Gradient Computation

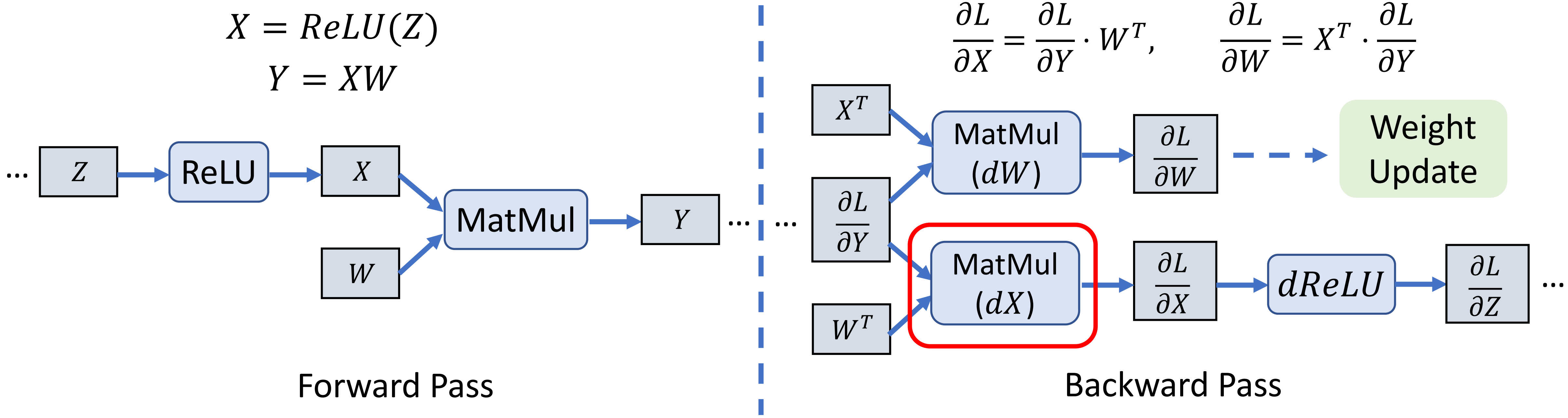
$$X = ReLU(Z)$$
$$Y = XW$$



Forward Pass

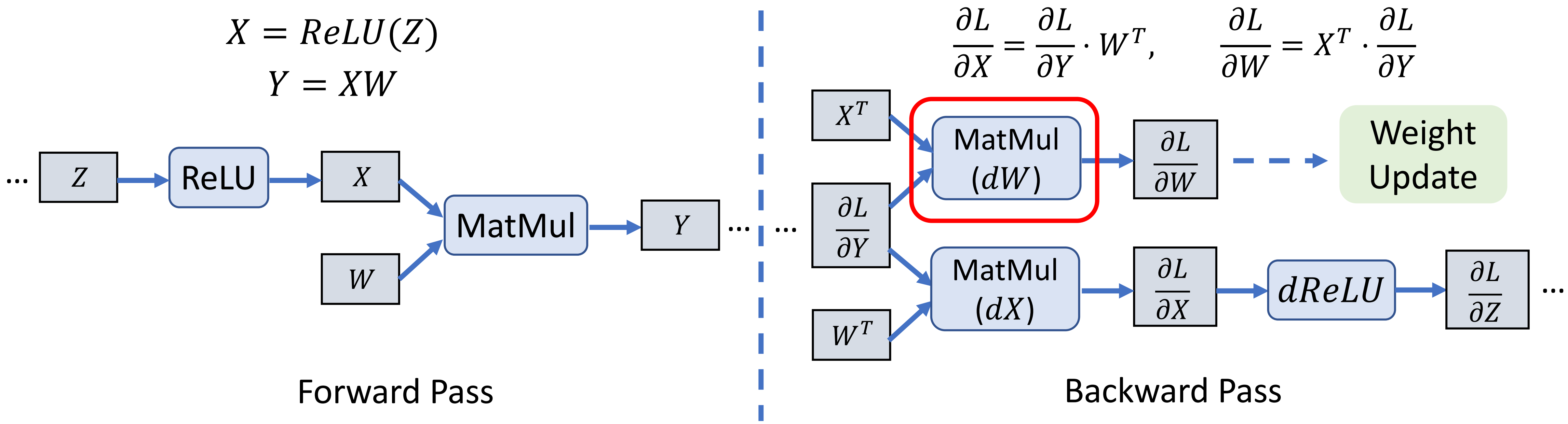
Extend the scope of overlapping

Opportunity 1: Weight Gradient Computation



Extend the scope of overlapping

Opportunity 1: Weight Gradient Computation

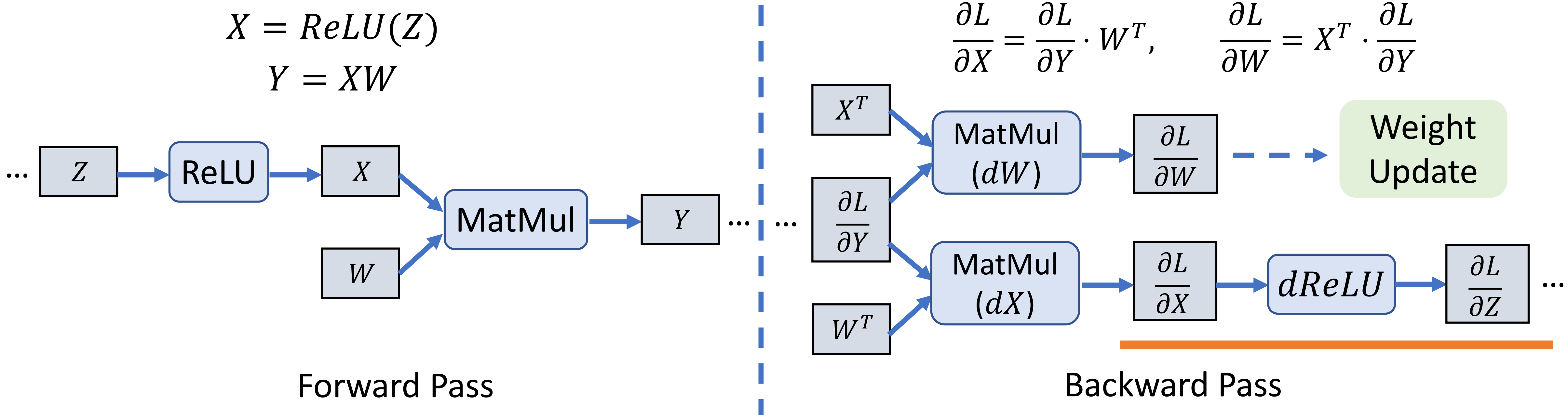


Extend the scope of overlapping

Opportunity 1: Weight Gradient Computation

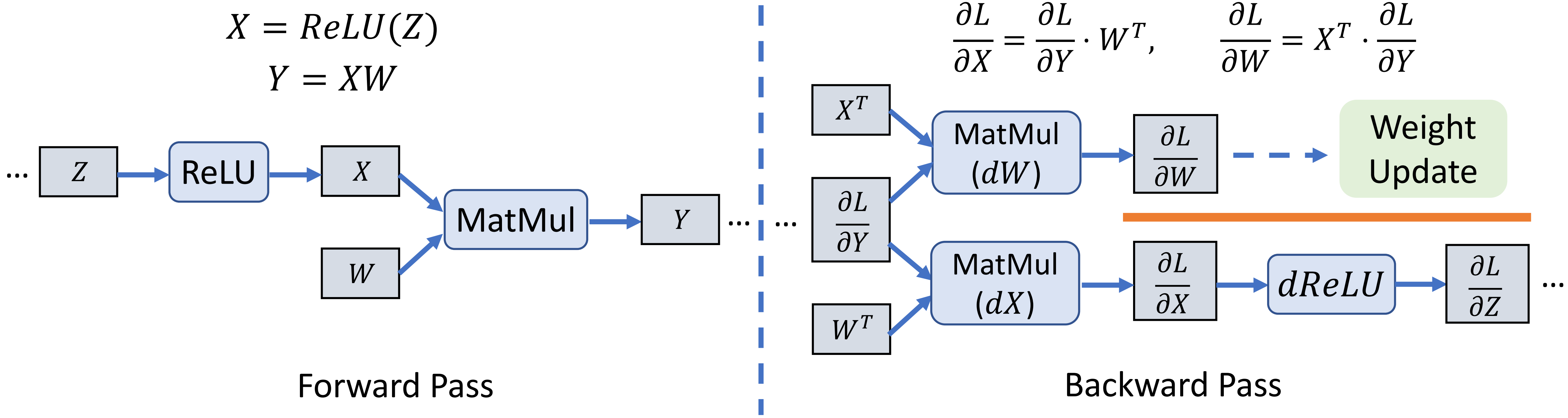
$$X = ReLU(Z)$$
$$Y = XW$$

$$\frac{\partial L}{\partial X} = \frac{\partial L}{\partial Y} \cdot W^T, \qquad \frac{\partial L}{\partial W} = X^T \cdot \frac{\partial L}{\partial Y}$$



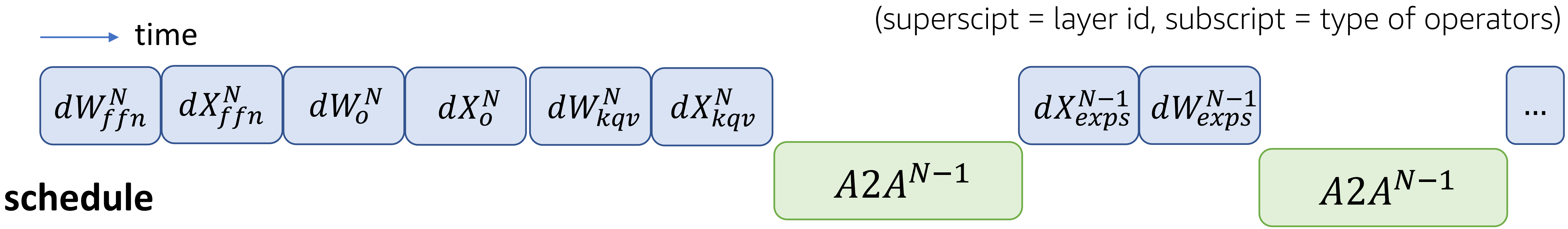
Extend the scope of overlapping

Opportunity 1: Weight Gradient Computation



Extend the scope of overlapping

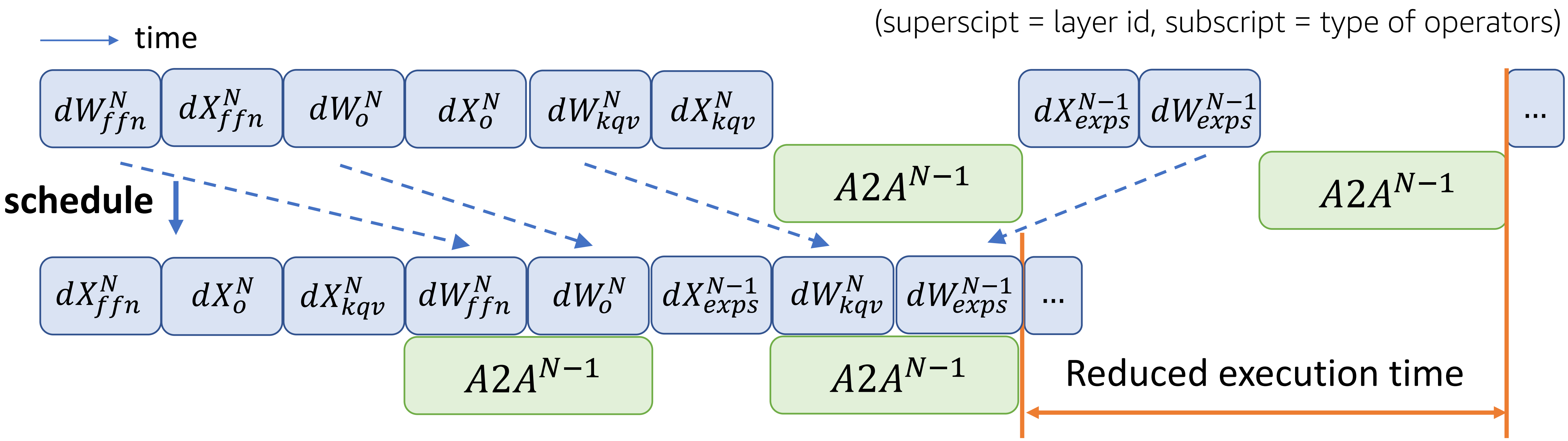
Opportunity 1: Weight Gradient Computation



Weight gradient computation can be scheduled to overlap with All-to-All during the backward pass.

Extend the scope of overlapping

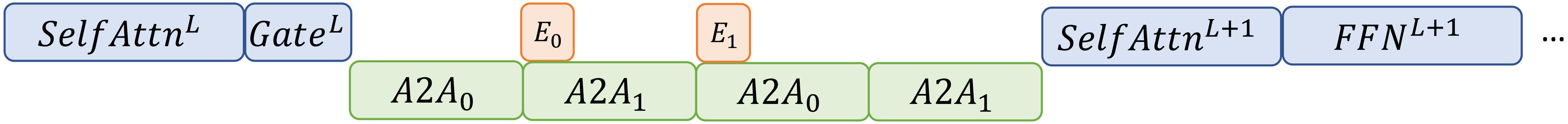
Opportunity 1: Weight Gradient Computation



Weight gradient computation can be scheduled to overlap with All-to-All during the backward pass.

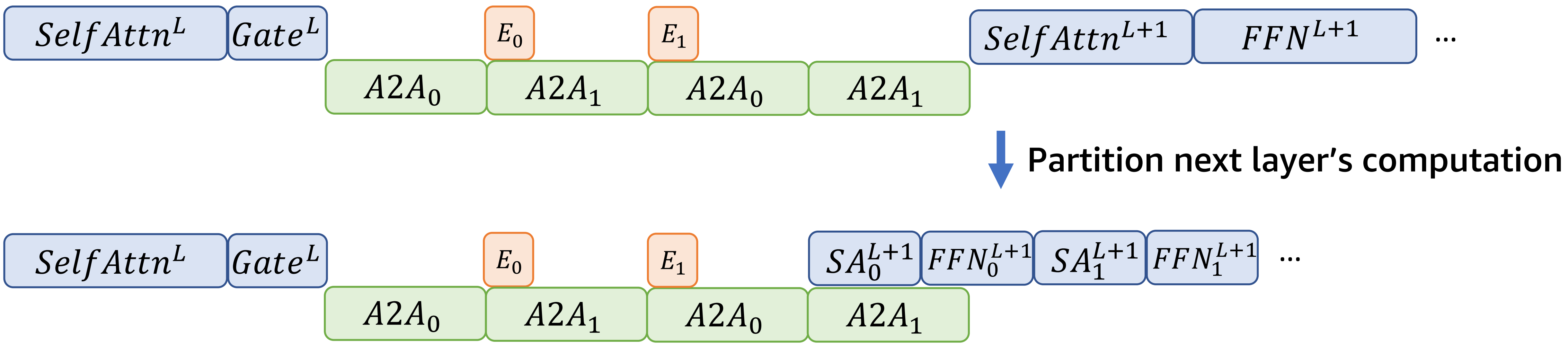
Extend the scope of overlapping

Opportunity 2: Non-expert computation



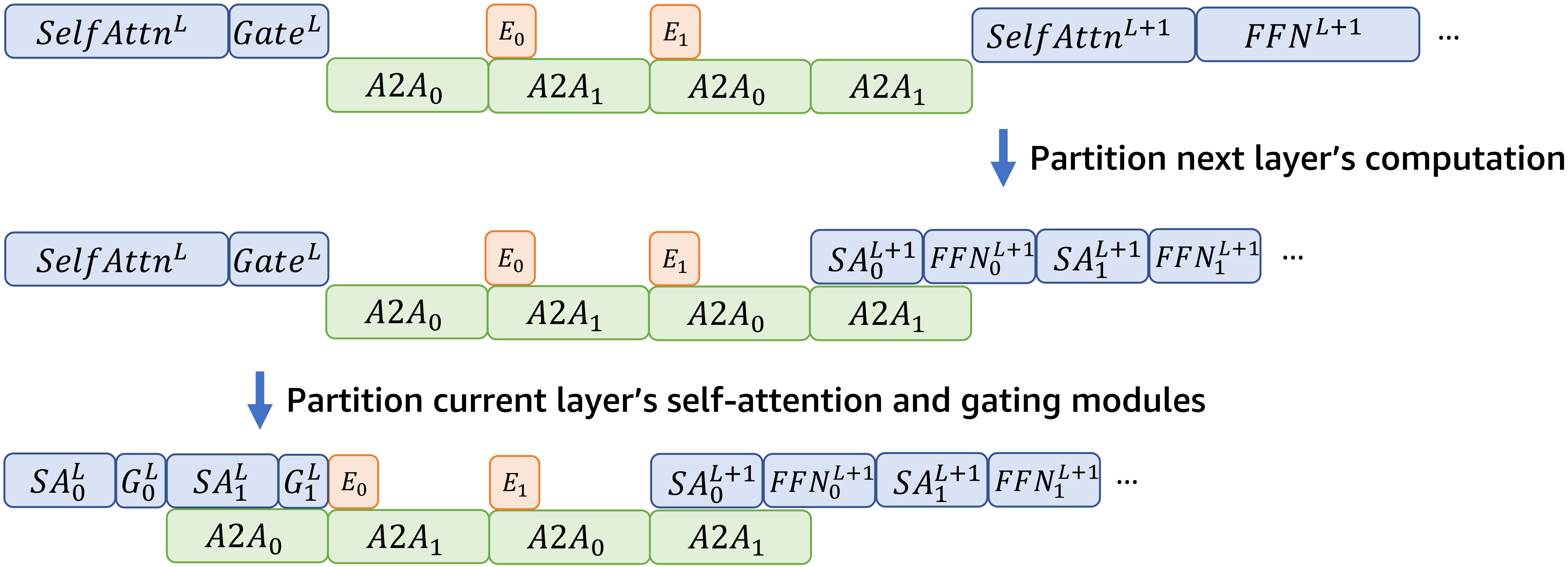
Extend the scope of overlapping

Opportunity 2: Non-expert computation



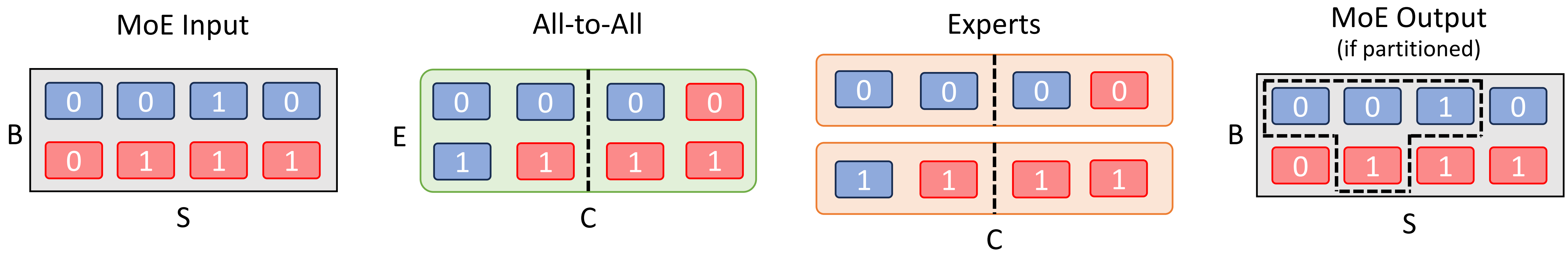
Extend the scope of overlapping

Opportunity 2: Non-expert computation



Extend the scope of overlapping

Caveat 1: Mathematically equivalent partitioning

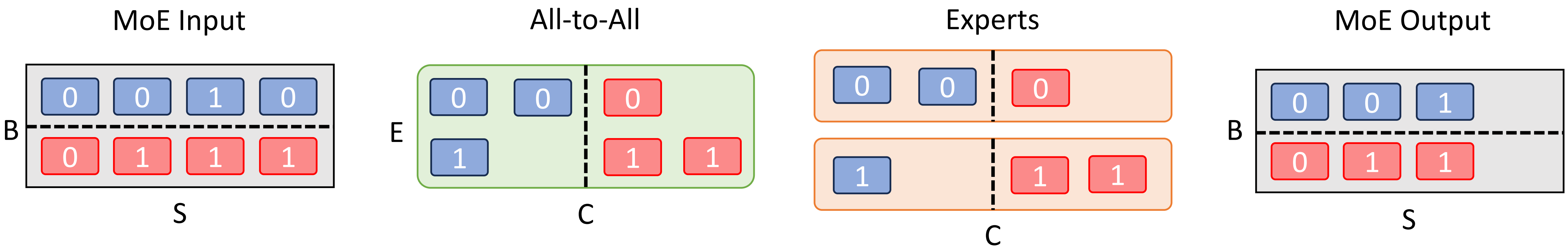


Partition at capacity dimension (current approach) limits the range of the pipeline.

Extend the scope of overlapping

Caveat 1: Mathematically equivalent partitioning

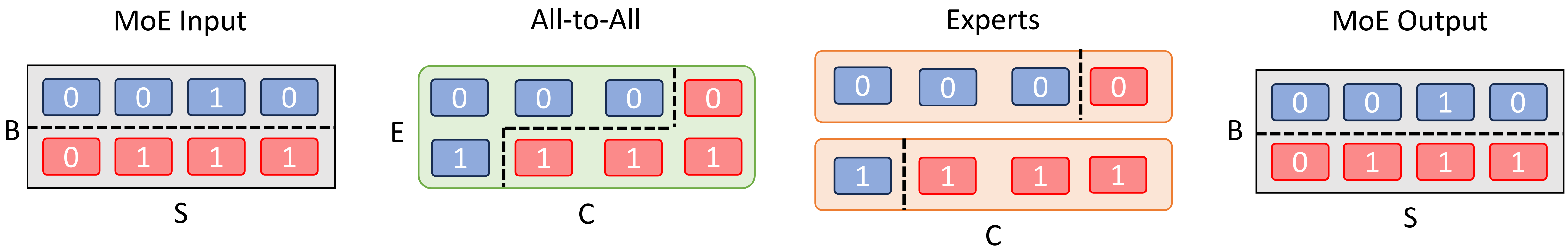
Partition at capacity dimension (current approach) limits the range of the pipeline.



Direct micro-batching may result in different token dropping patterns.

Extend the scope of overlapping

Caveat 1: Mathematically equivalent partitioning (cont'd)

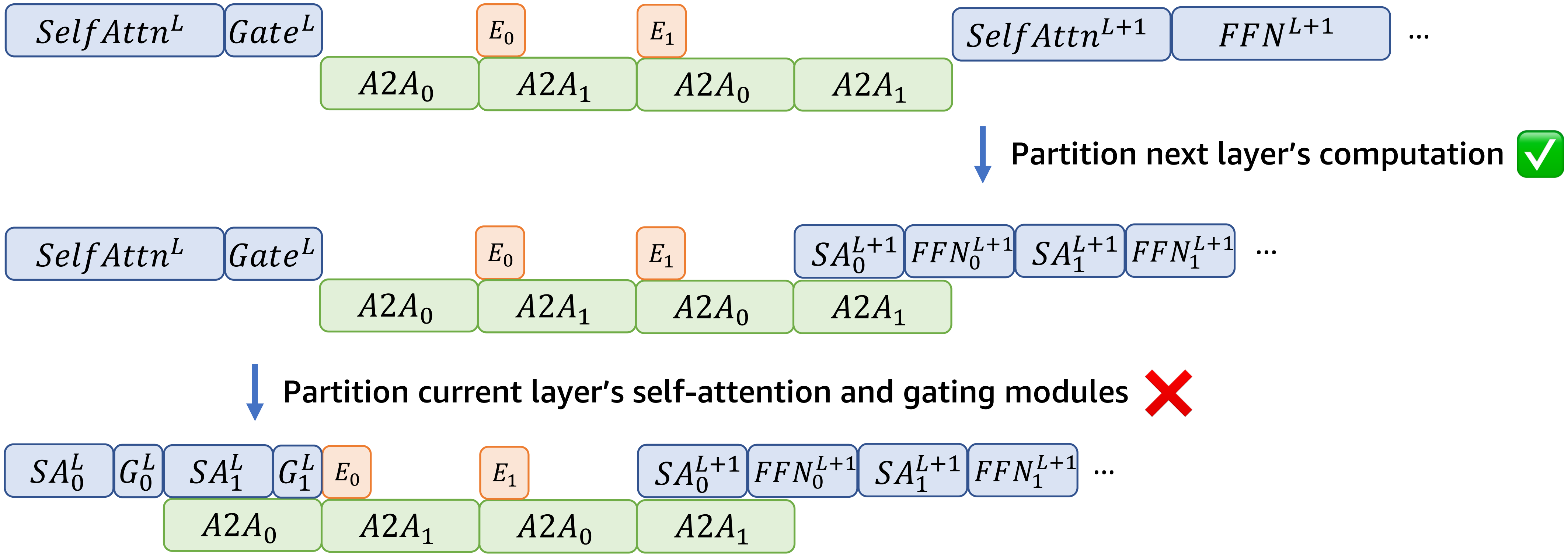


Irregular partitioning needed to ensure mathematic equivalence.

Extend the scope of overlapping

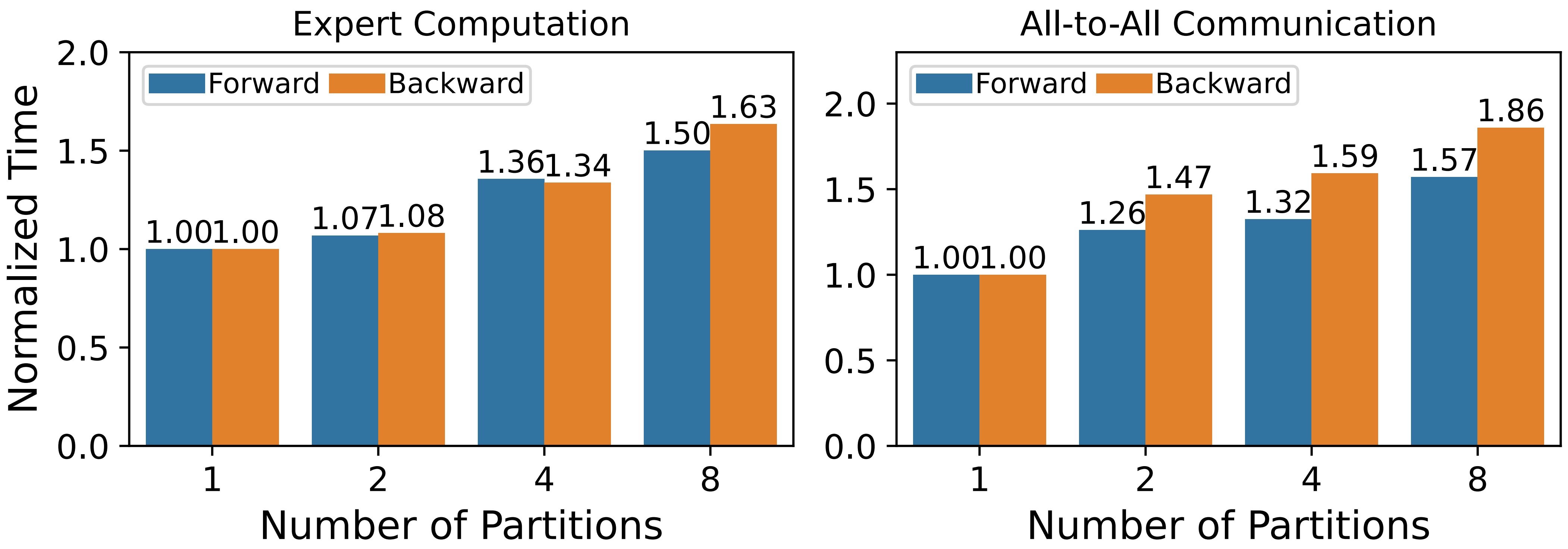
Caveat 1: Mathematically equivalent partitioning (cont'd)

Some routing methods (e.g., Batch Priority Gating, Expert Choice Gating) requires information of the whole batch, thus the pipeline cannot be extended before the gating operator.



Extend the scope of overlapping

Caveat 2: Determine the range of pipelines

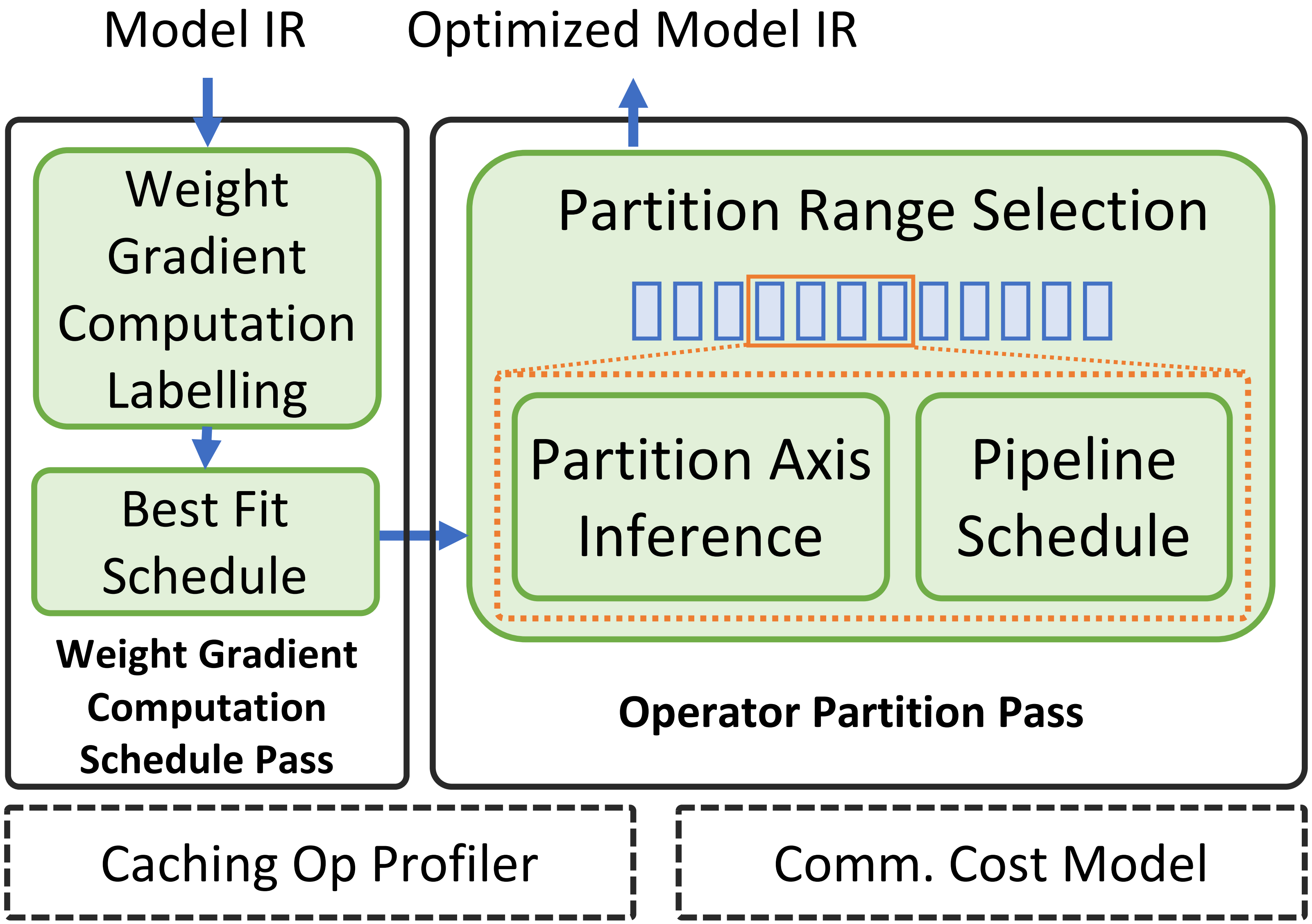


Partition overhead in Tutel, running a GPT2-MoE model with 32 experts on 2 p4d nodes (16 GPUs)

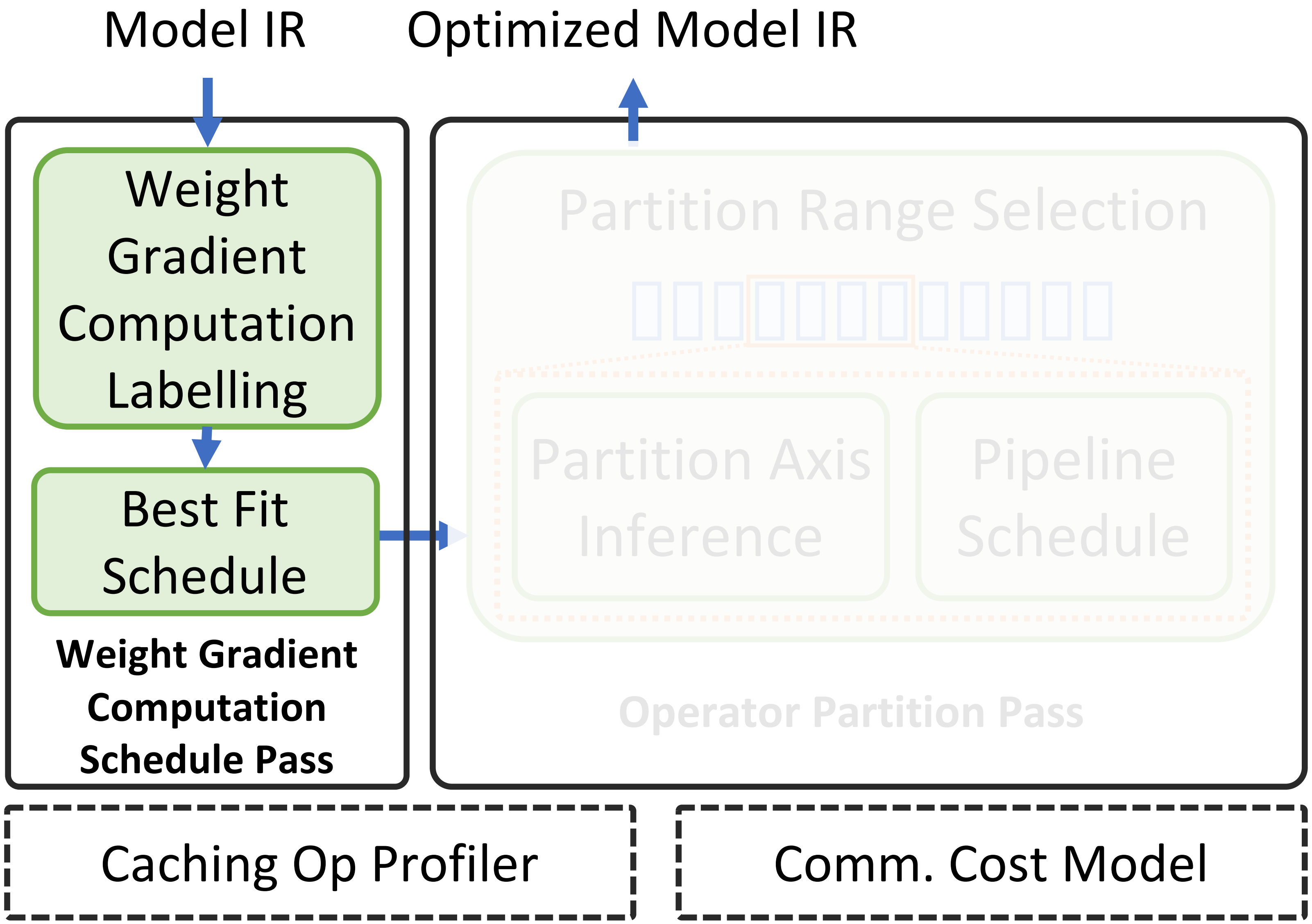
Pipeline too **short** → insufficient overlapping

Pipeline too **long** → high partition overhead

Lancet: compiler based optimizations



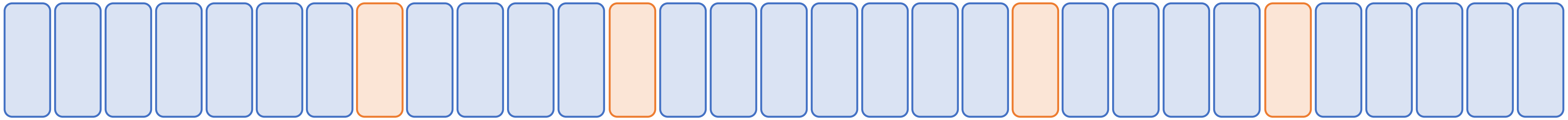
Lancet: compiler based optimizations



Weight Gradient Computation Schedule Pass

1. Dependency Analysis.

Instruction Sequence

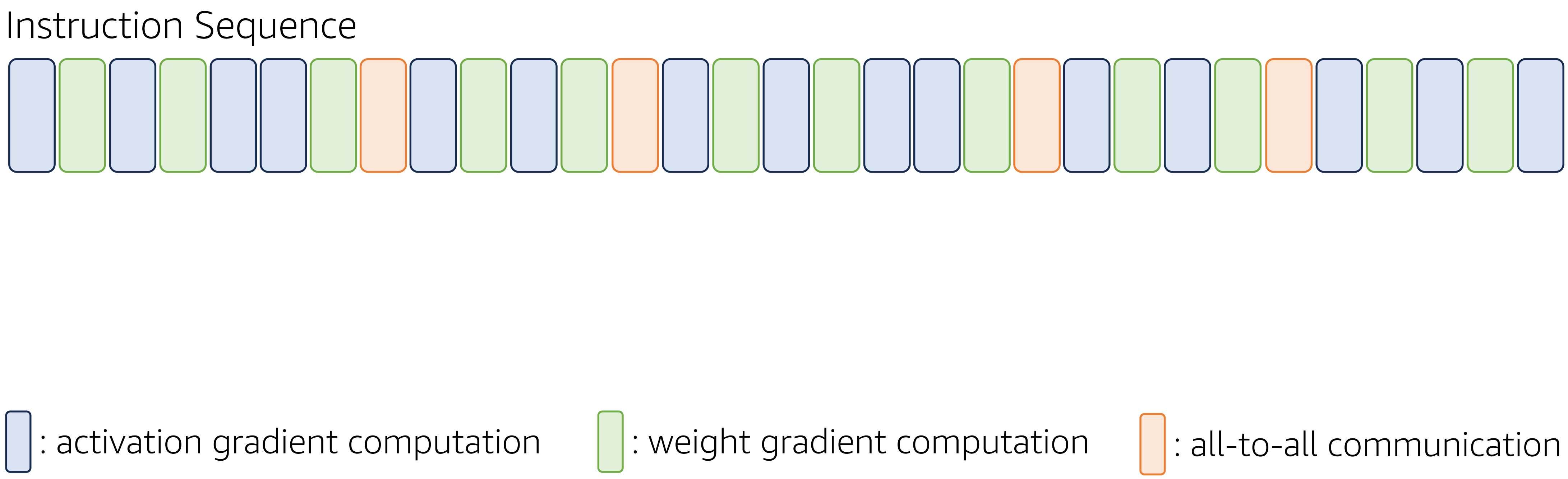


: computation instructions : weight gradient computation : all-to-all communication

Identify gradient computation operations,

Weight Gradient Computation Schedule Pass

1. Dependency Analysis.

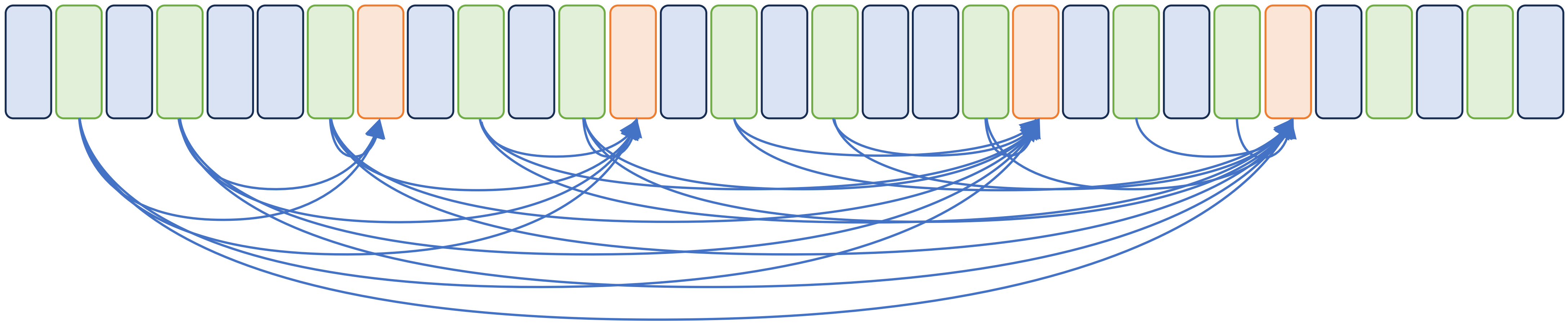


Identify gradient computation operations,

Weight Gradient Computation Schedule Pass

1. Dependency Analysis.

Instruction Sequence

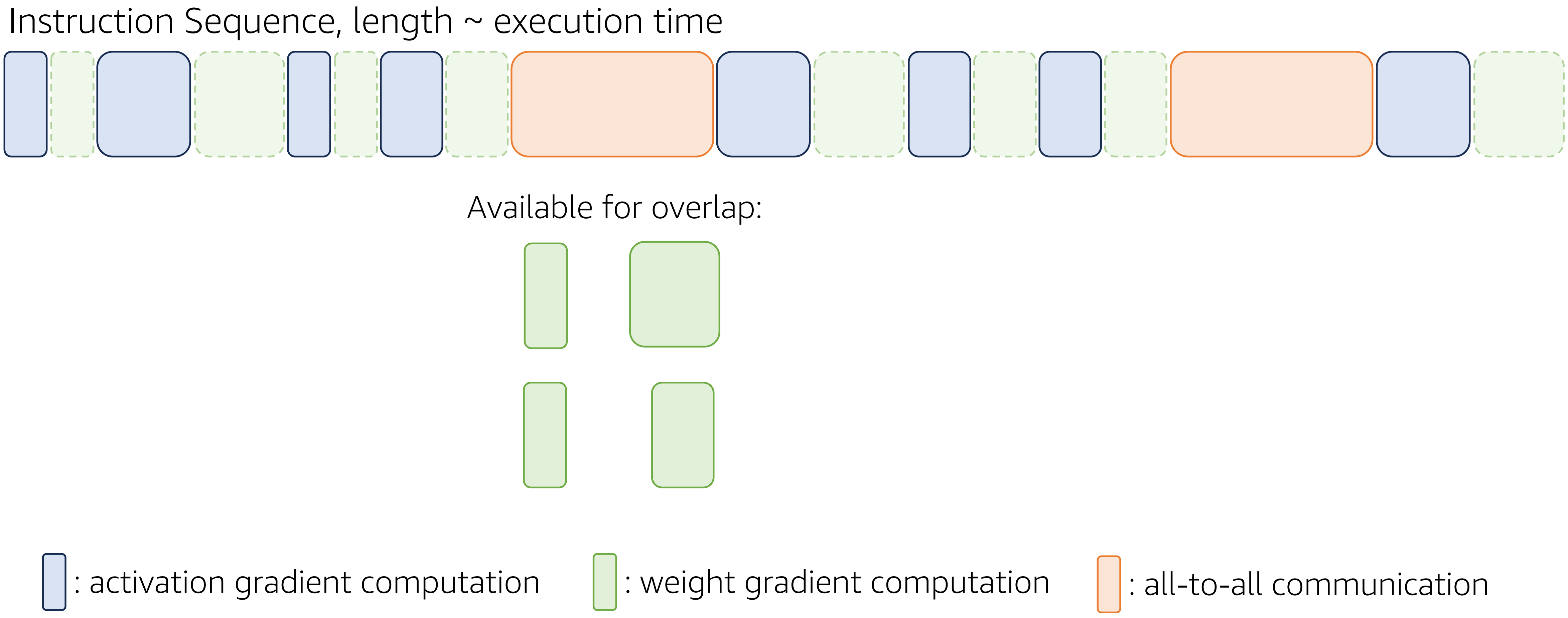


: activation gradient computation   : weight gradient computation   : all-to-all communication

Identify gradient computation operations, and the all-to-alls that can be overlapped with each.

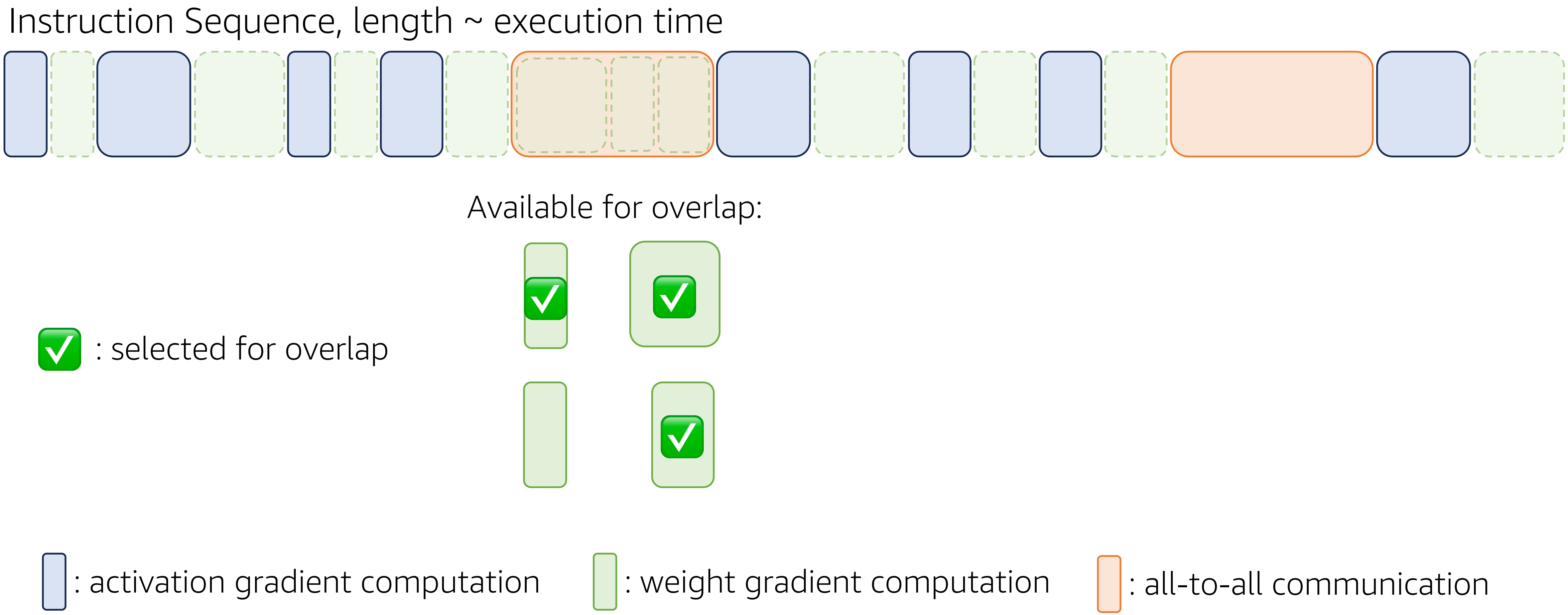
Weight Gradient Computation Schedule Pass

2. Greedy best fit schedule



Weight Gradient Computation Schedule Pass

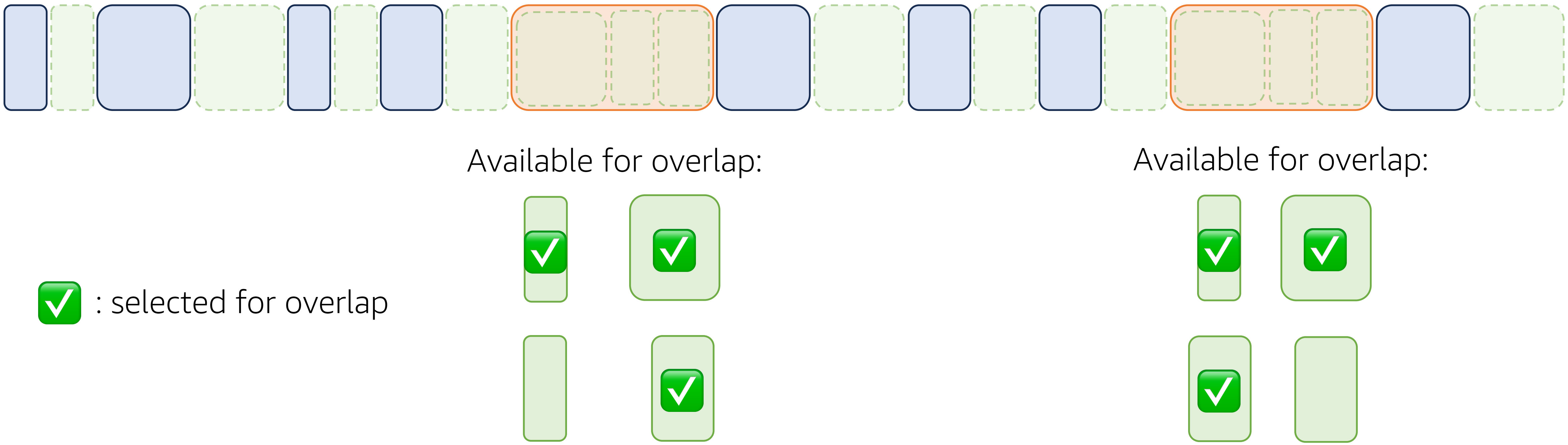
2. Greedy best fit schedule



Weight Gradient Computation Schedule Pass

2. Greedy best fit schedule

Instruction Sequence, length ~ execution time

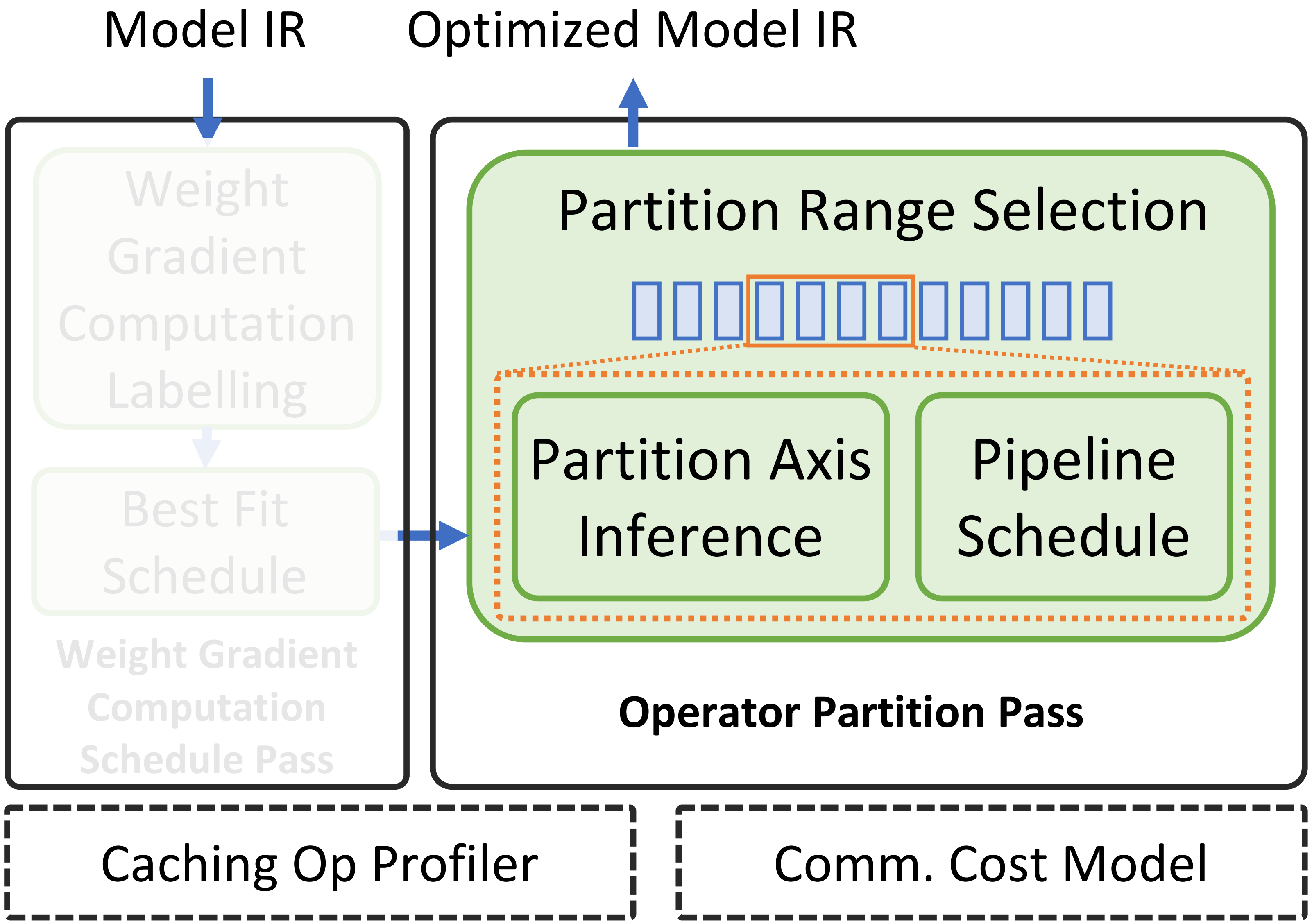


: activation gradient computation

: weight gradient computation

: all-to-all communication

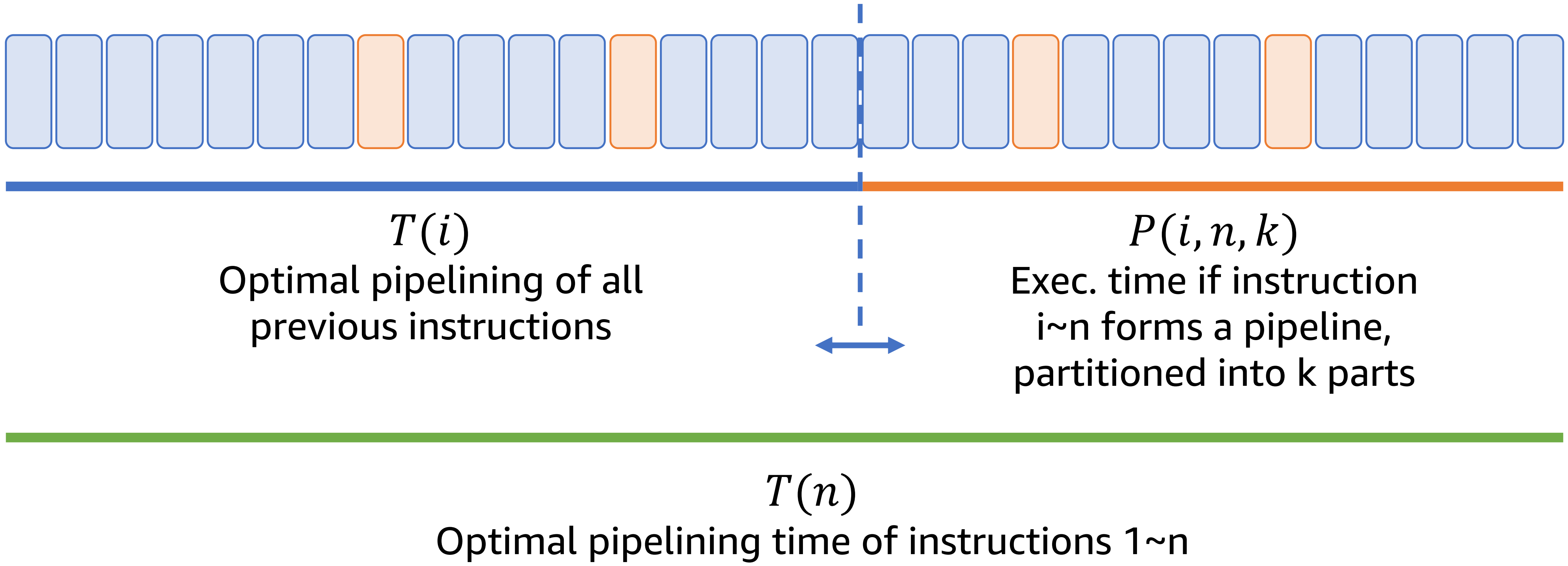
Lancet: compiler based optimizations



Operator Partition Pass

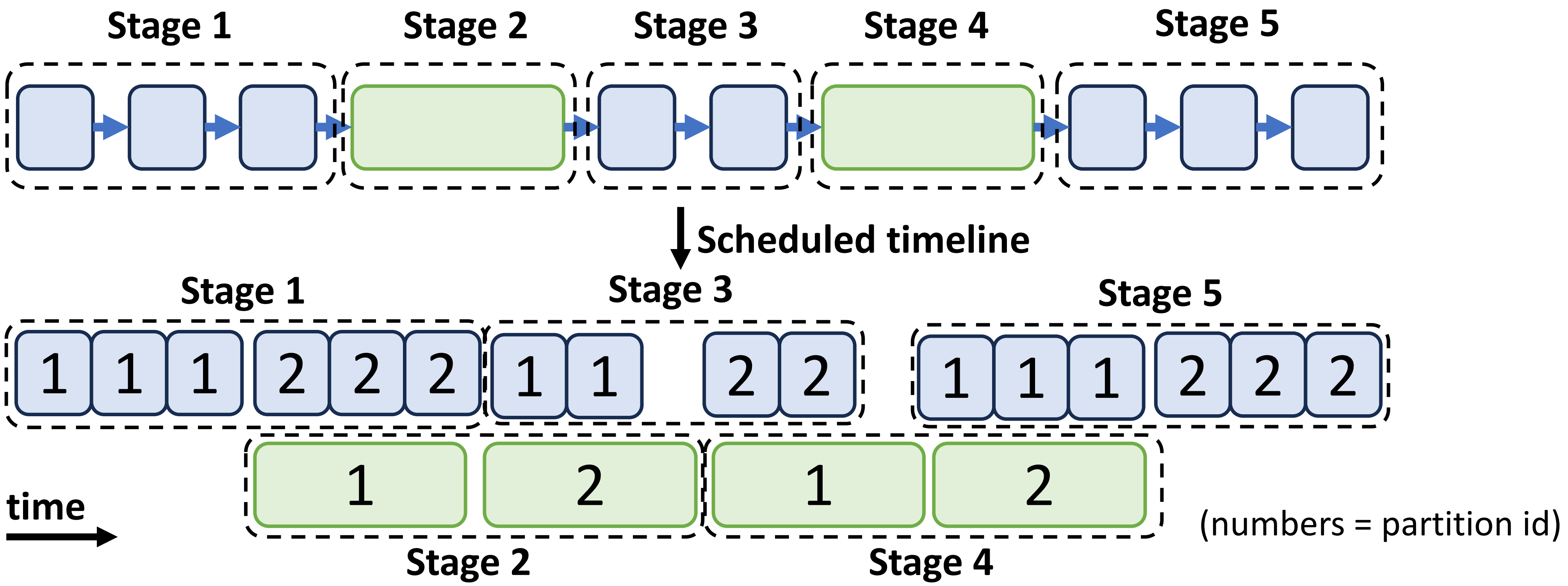
Solve for the optimal partition range with dynamic programming

$$T(n) = \min_{1 \leq i \leq n-1} \{T(i) + \min_{1 \leq k \leq K} P(i,n,k)\}$$



Operator Partition Pass

Pipeline scheduling by stages



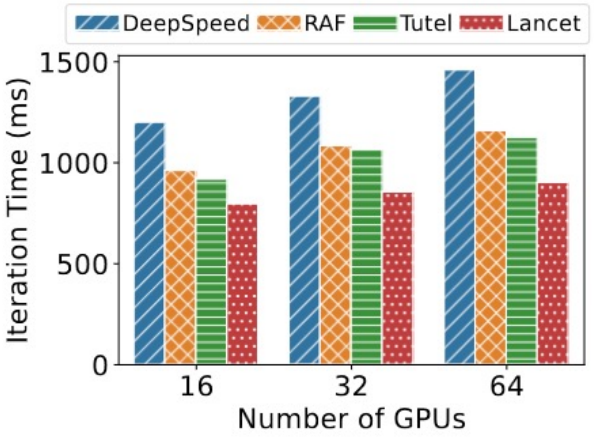
Evaluation

Testbed: Up to 8x AWS EC2 p4de (A100) and p3dn (V100) nodes (8xGPUs each node, 64 GPUs in total)

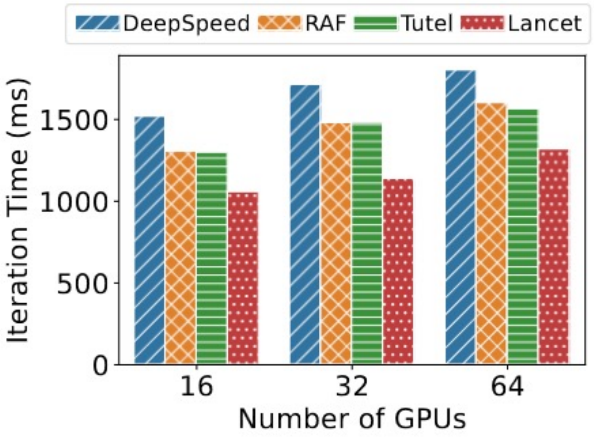
Dataset: WikiText **Models**: GPT2+MoE with two different model sizes, 2 experts per GPU.

Baseline: RAF (without optimization), Tutel, DeepSpeed.

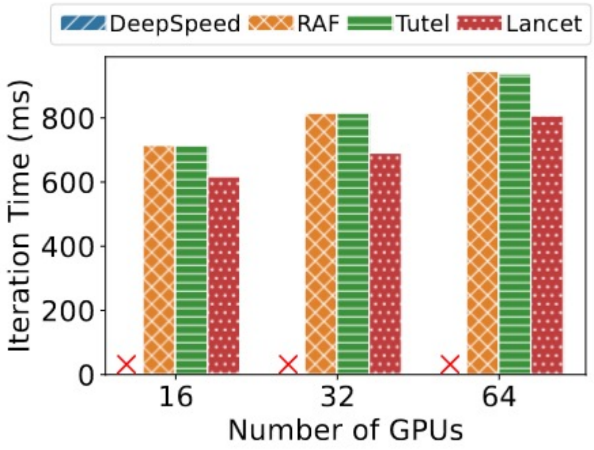
Iteration time comparison



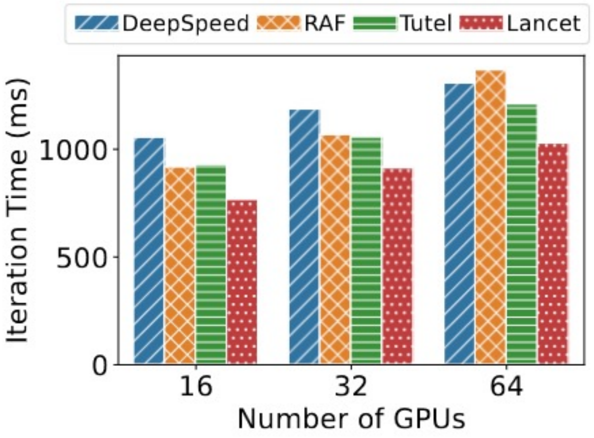
(a) GPT2-S-MoE, V100.



(b) GPT2-L-MoE, V100.



(c) GPT2-S-MoE, A100.

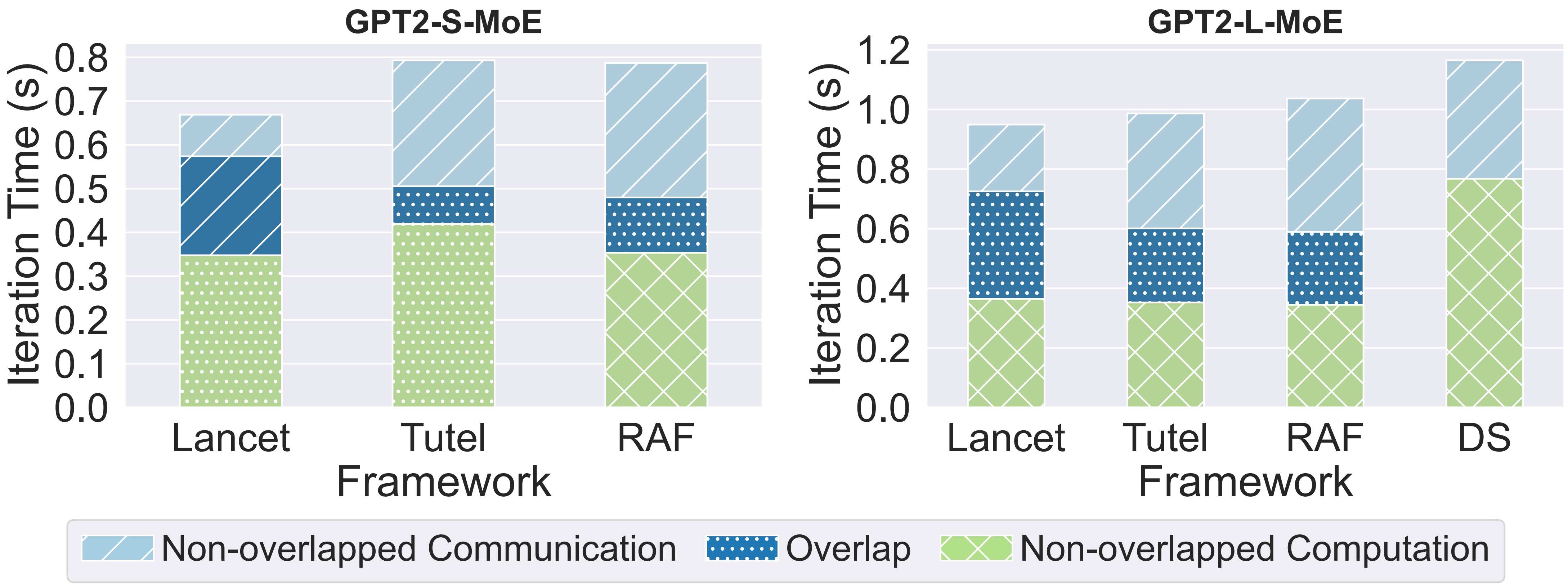


(d) GPT2-L-MoE, A100.

Up to **1.3x** speed up.

Evaluation

Iteration time decomposition

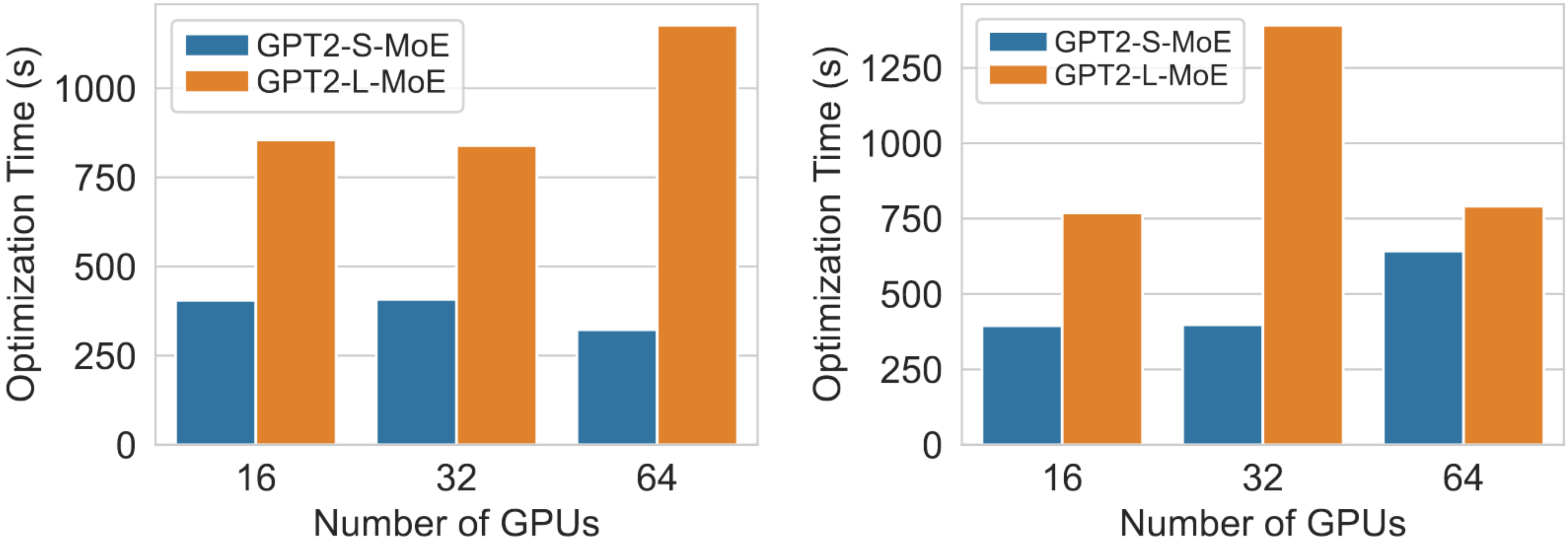


Results on 4x p4de (A100) nodes.

Reducing non-overlapped communication time by up to **77%.**

Evaluation

Optimization time



(a) V100 cluster (b) A100 cluster

(when using the Switch Gate)

The optimization can finish in a reasonable amount of time (e.g., 20 mins).

Summary

Extending optimization scope to the whole model enables more computation-communication overlapping opportunities:

- Weight gradient computation
- Non-MoE computations (self-attention, non-MoE FFNs)

Up to 1.3x speed up is observed after applying these optimizations.

Checkout the paper here: