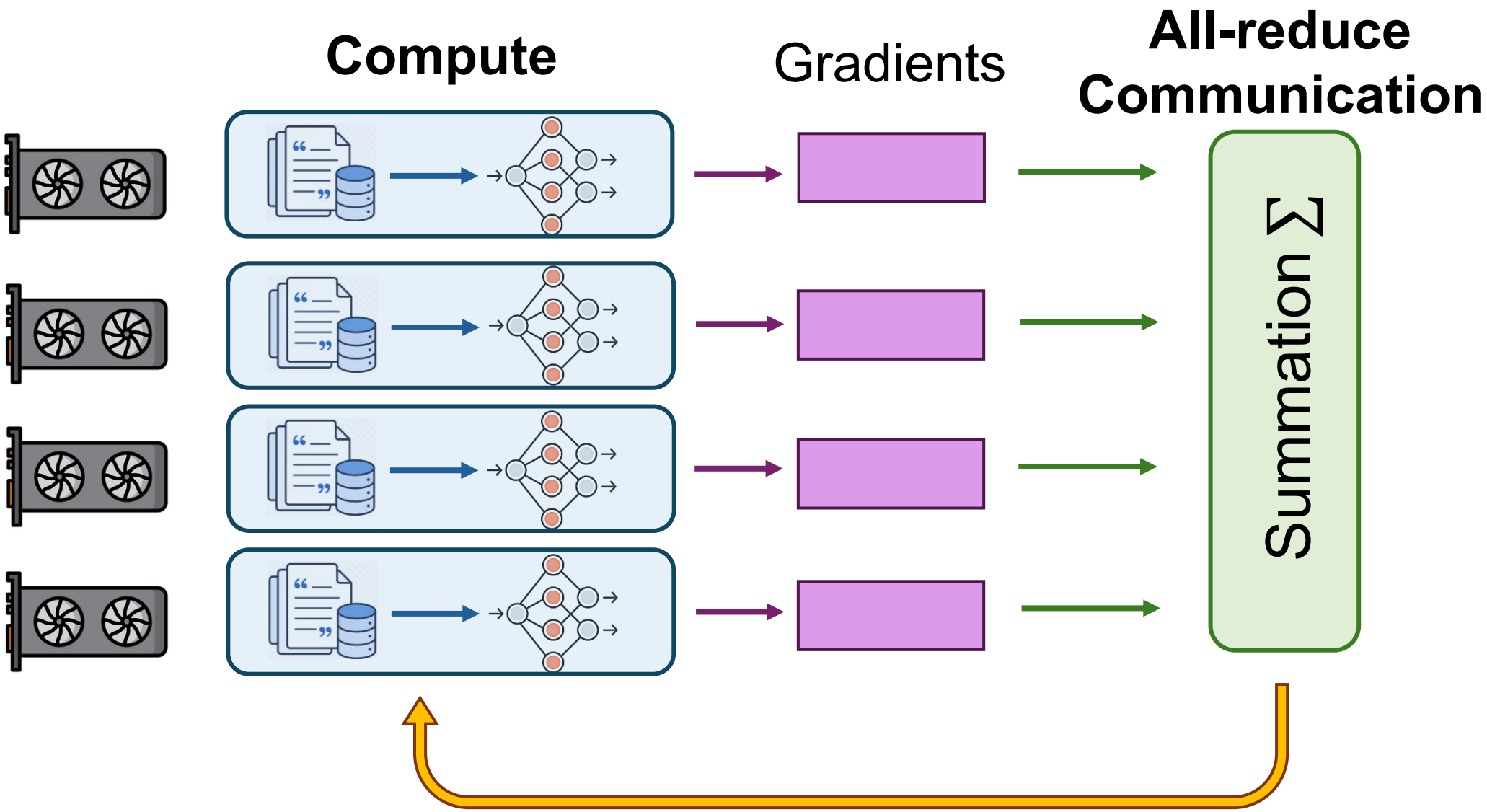
To appear at SIGCOMM ’26

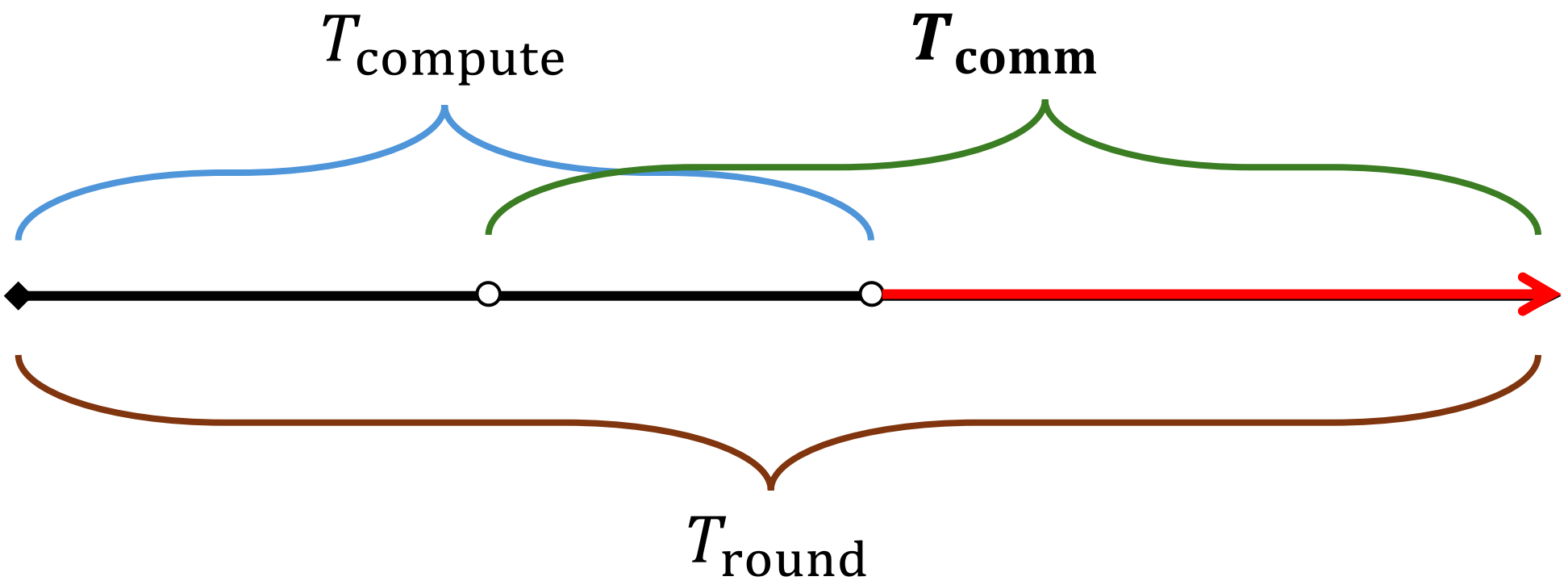
DynamiQ: Accelerating Gradient Synchronization using Compressed Multi-hop All-reduce

Wenchen Han, Shay Vargaftik
Michael Mitzenmacher, Ran Ben Basat

Data Parallelism in Distributed LLM Training

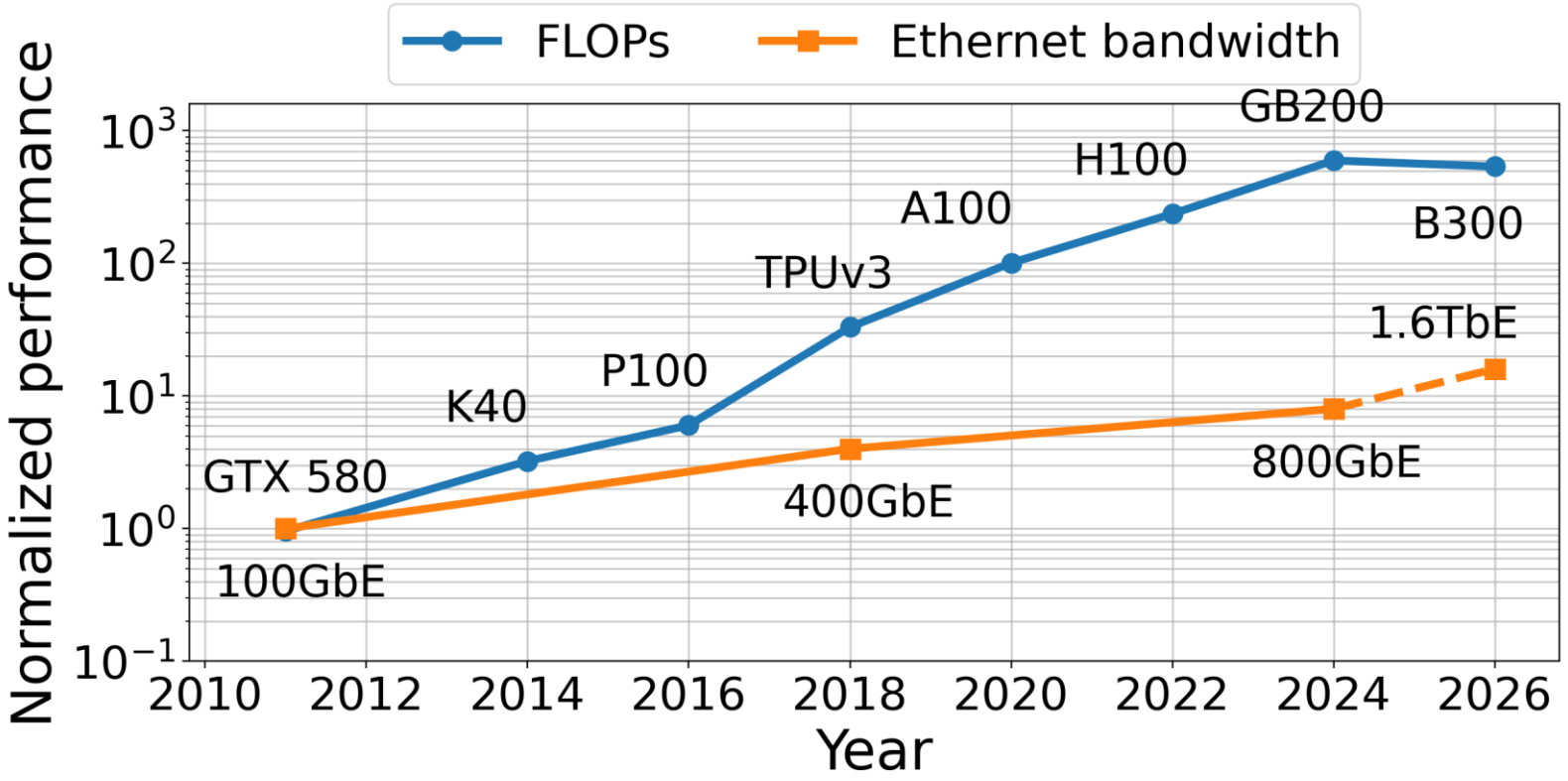


Data Parallelism in Distributed LLM Training

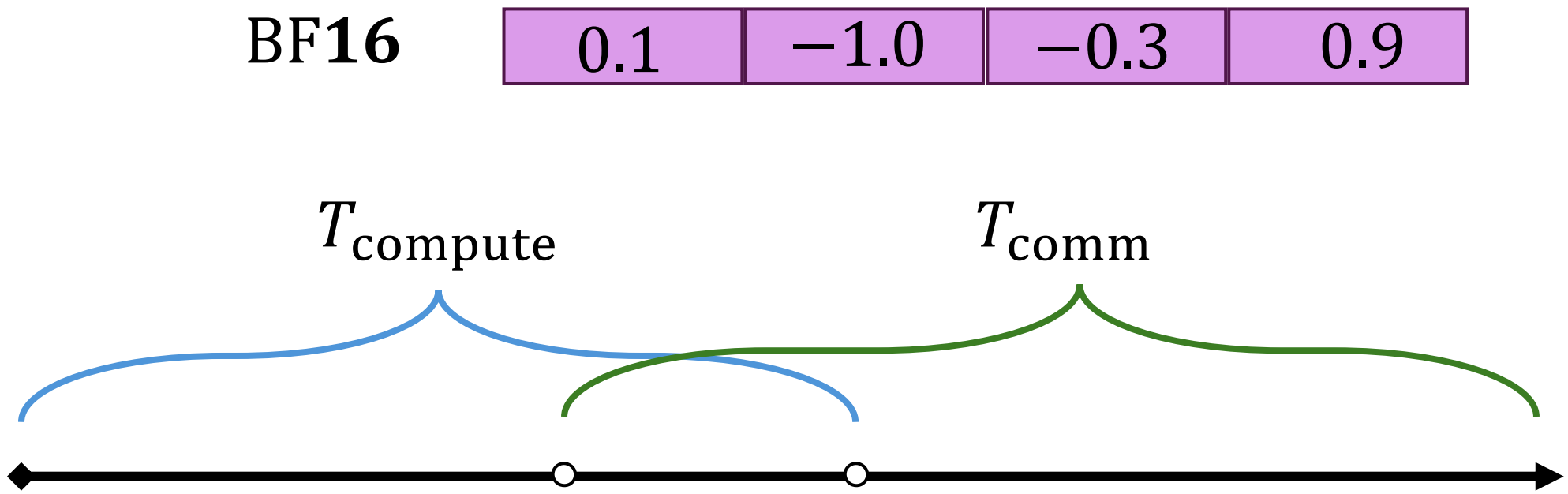


Gradient Communication Remains a Bottleneck

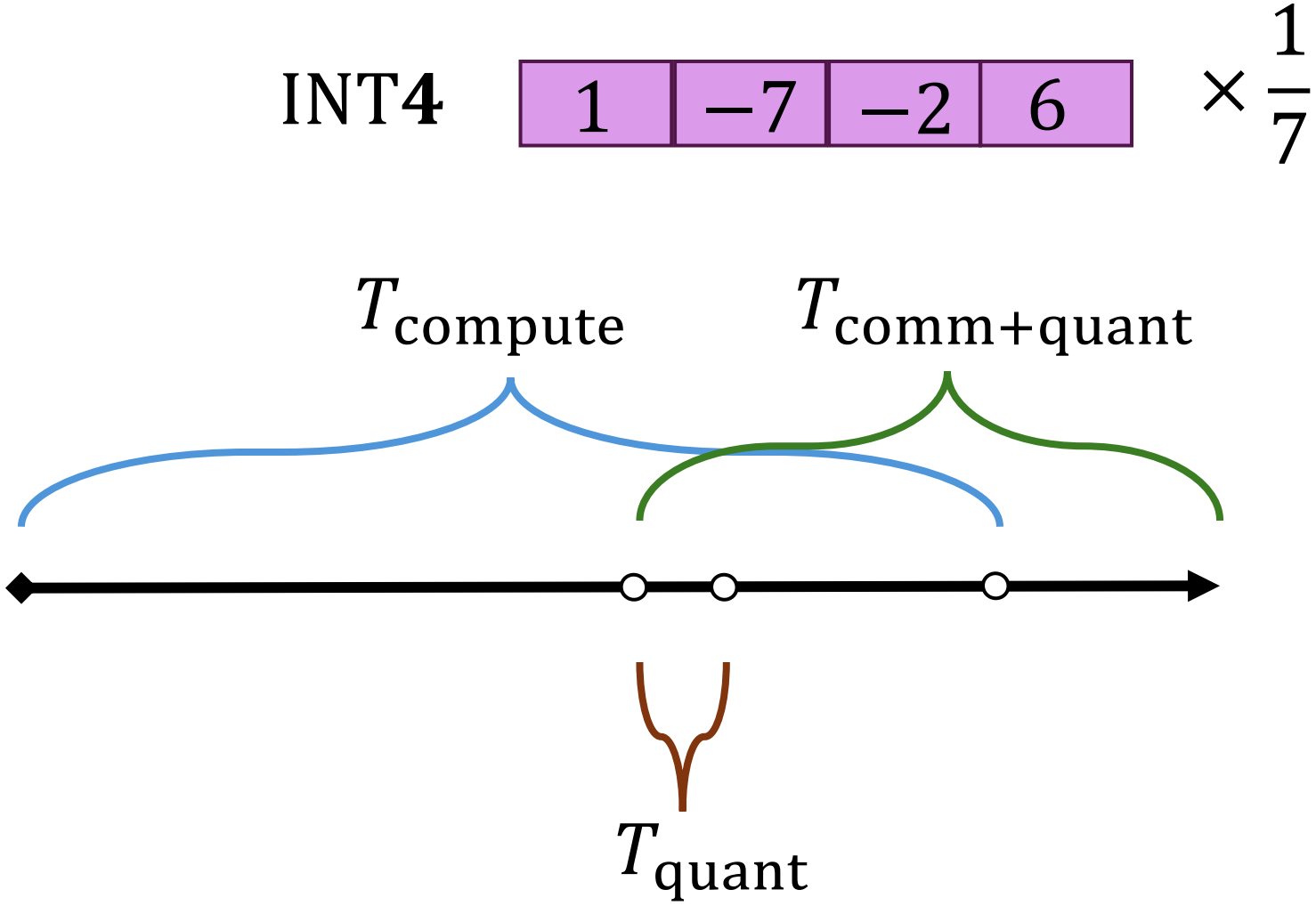
- Up to 44% in large LLM training, 800Gbps network [1].
- Compute grows faster than bandwidth.



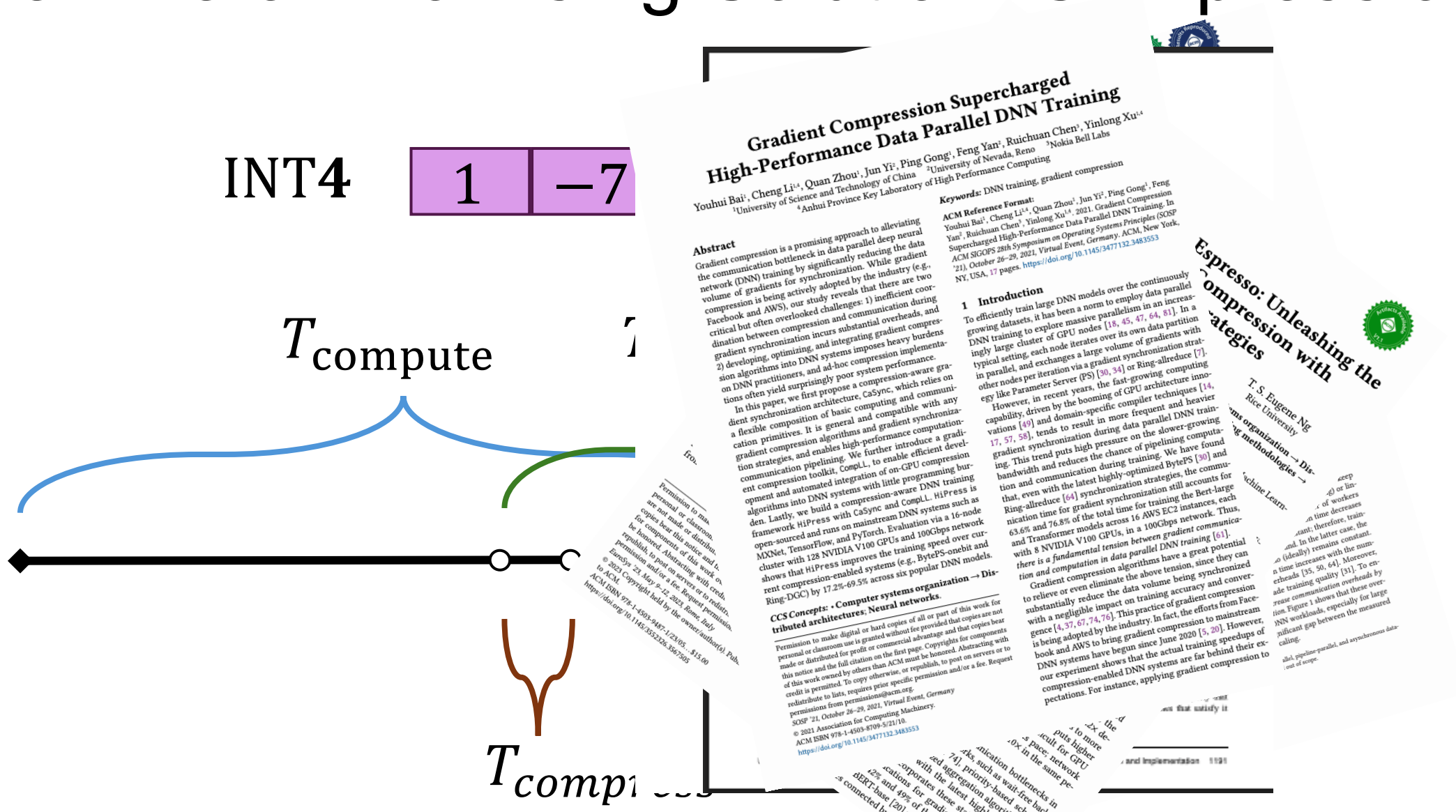
One Line of Promising Solutions: Compression



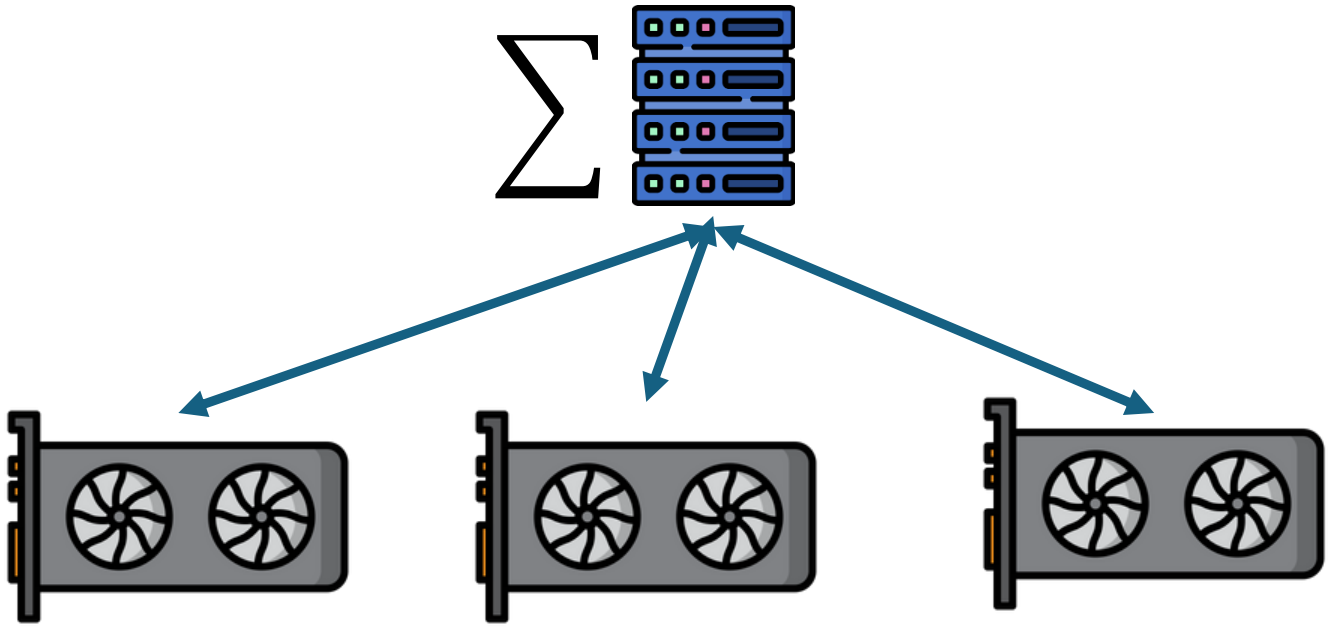
One Line of Promising Solutions: Compression



One Line of Promising Solution: Compression

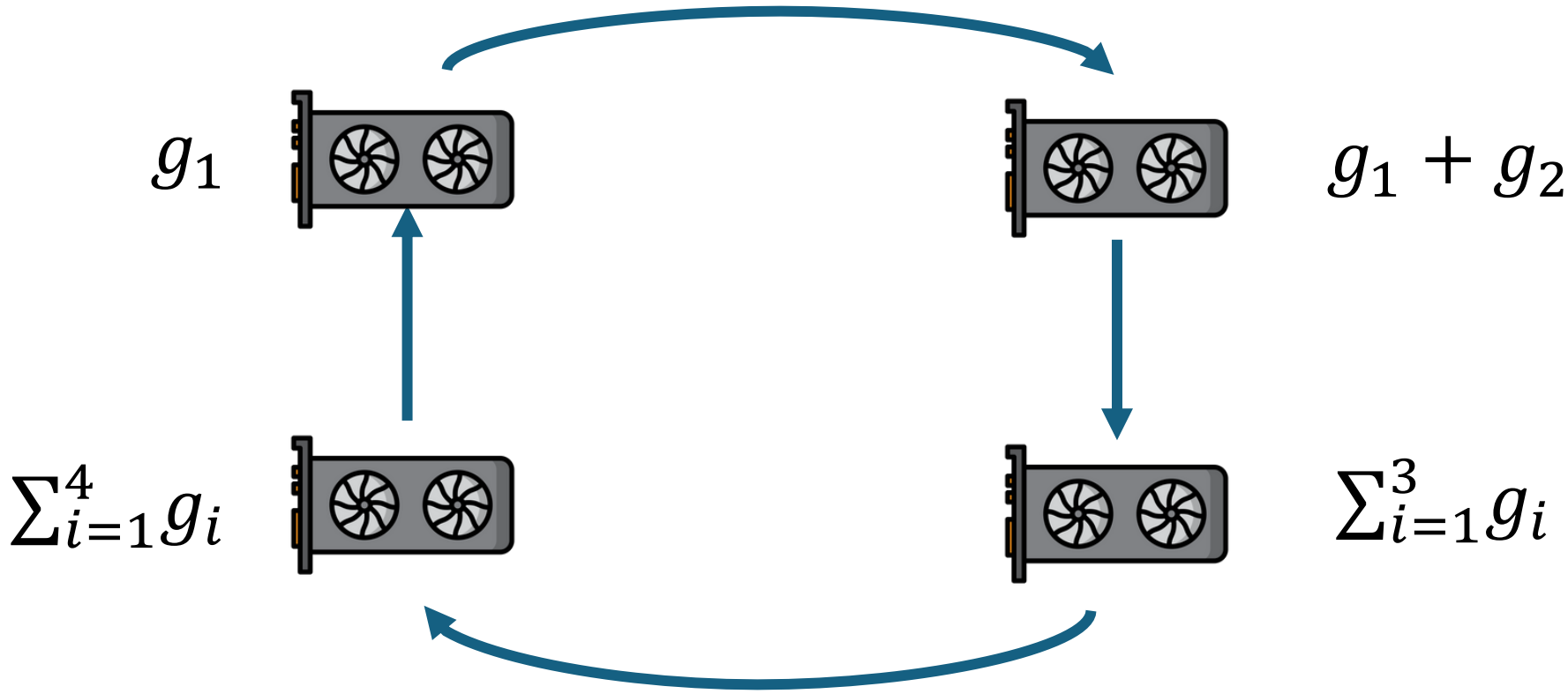


Most Compression Schemes Are Designed for PS[2]



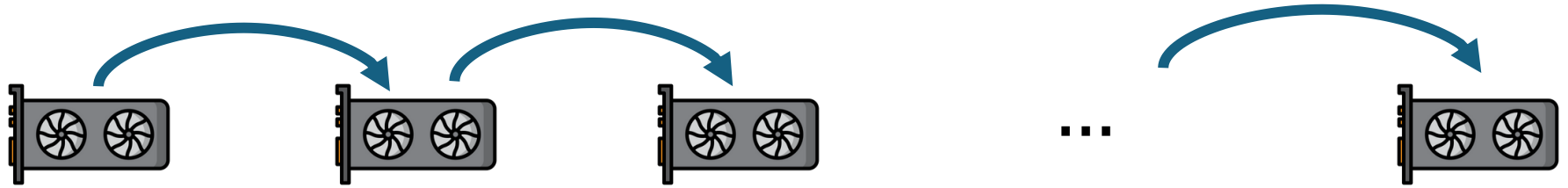
- Parameter Server (PS): Single-hop, centralized

None Are Optimized for **Multihop All-reduce**



- Multi-hop All-reduce, decentralized.
- Some variants are scalable, de-facto standard.

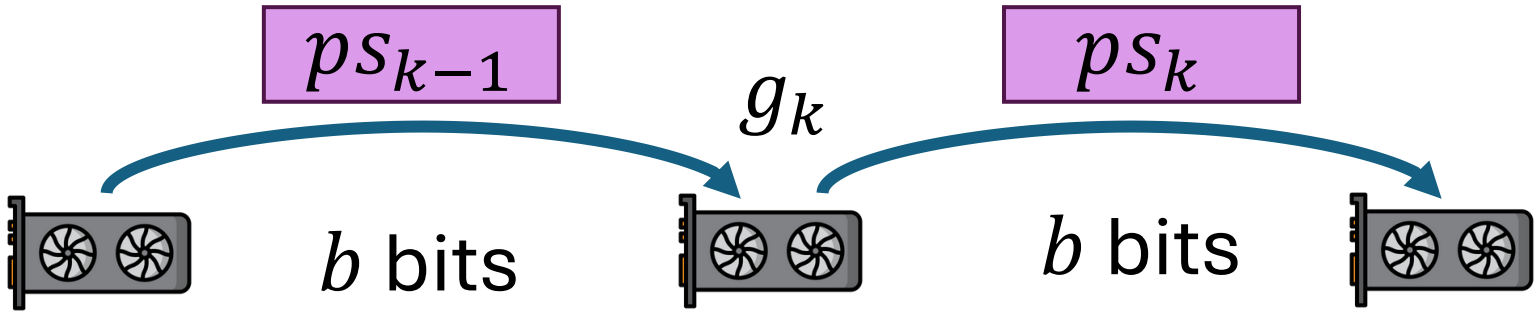
The Main Challenge: Overflows.



Partial Sum	1	-6	-8		$O(n)$
Compressed	1	-7	-2		$\lvert\hat{g}_n\rvert$
Original	0.1	-1.0	-0.3		g_n

Decompress Accumulate Recompress

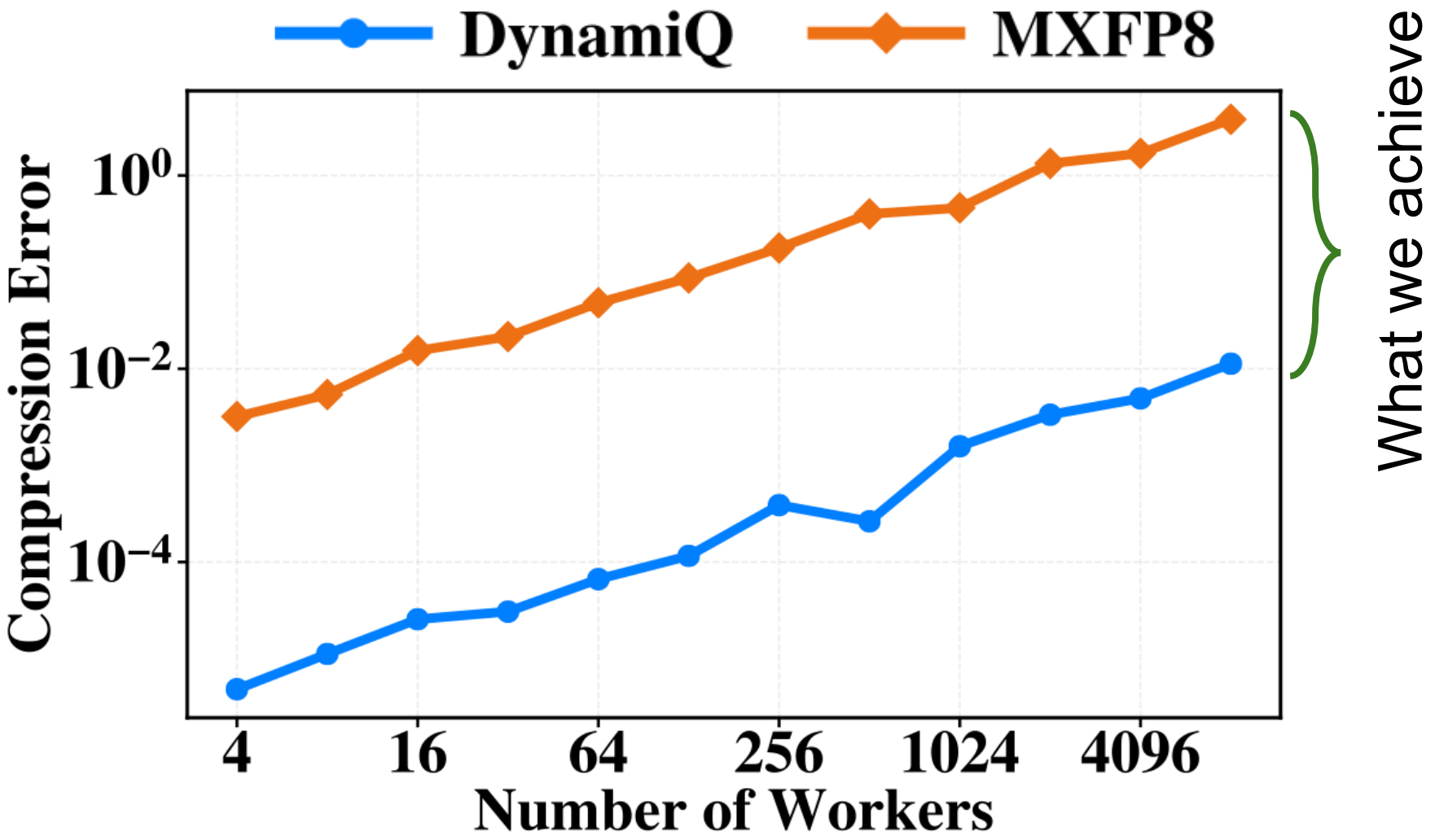
- $ps_k = \text{recompress}(\text{decompress}(ps_{k-1}) + g_k)$



Challenges

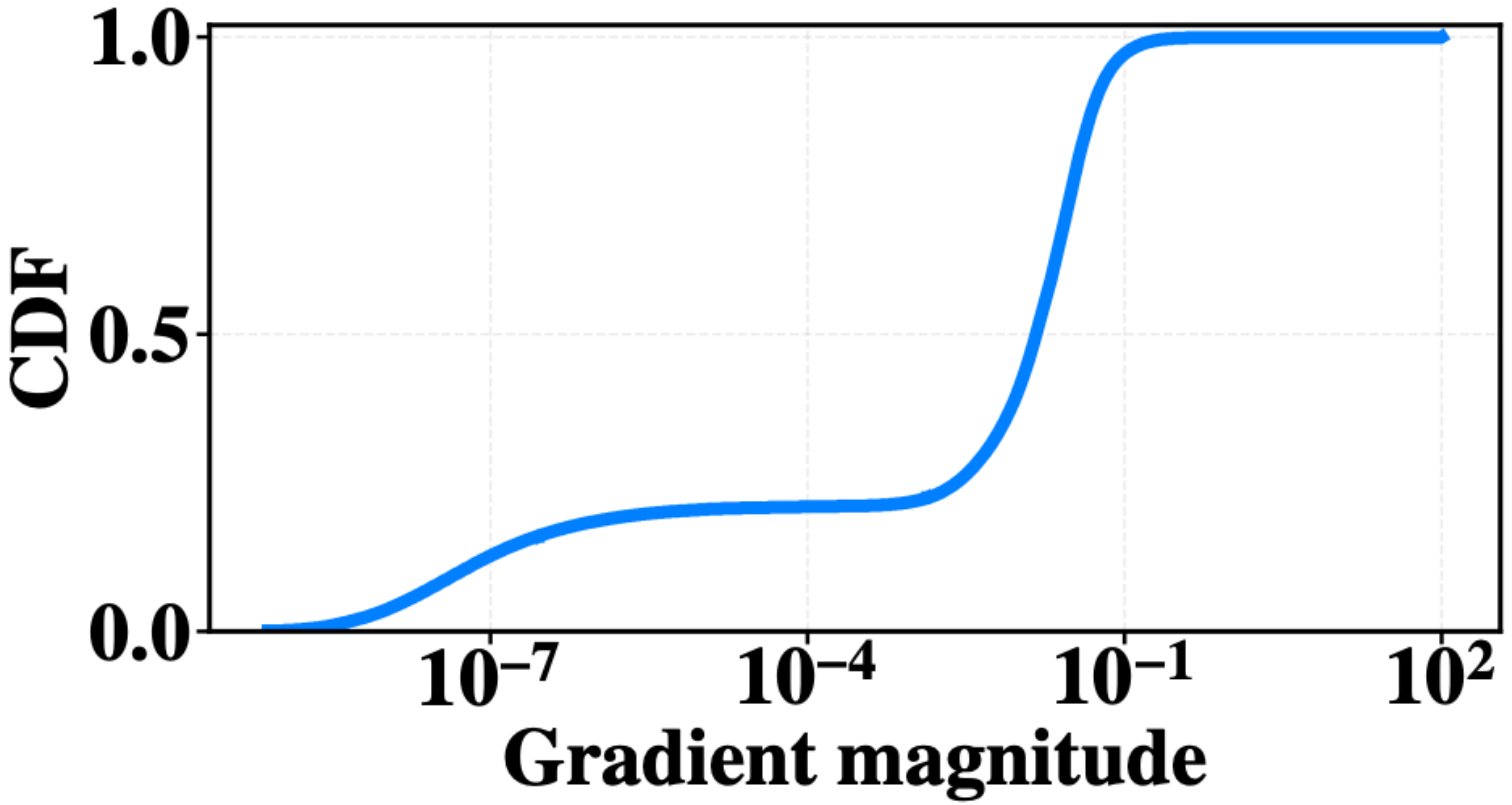
- Compression error that accumulates over hops
- Computational overhead for de/recompress

Relative Compression Error



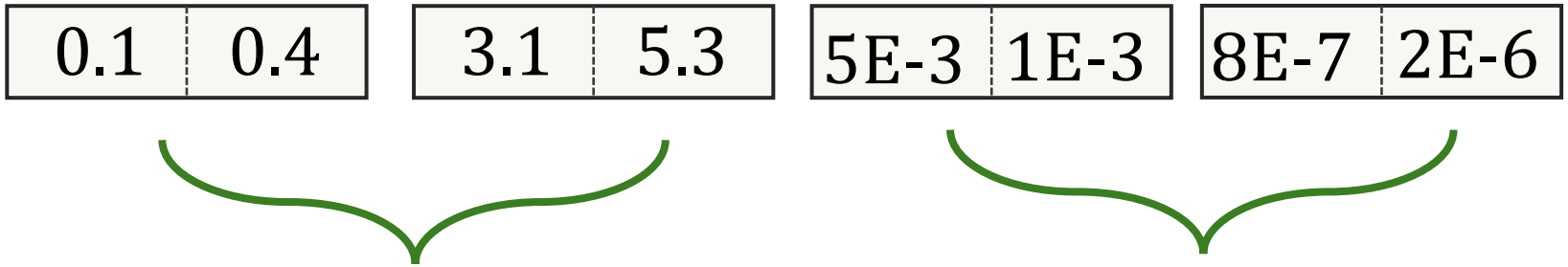
Minimizing Compression Error: Multi-bitrates

0.1	0.4	3.1	5.3	5E-3	1E-3	8E-7	2E-6



Highly Skewed Gradient

Minimizing Compression Error: Multi-bitrates



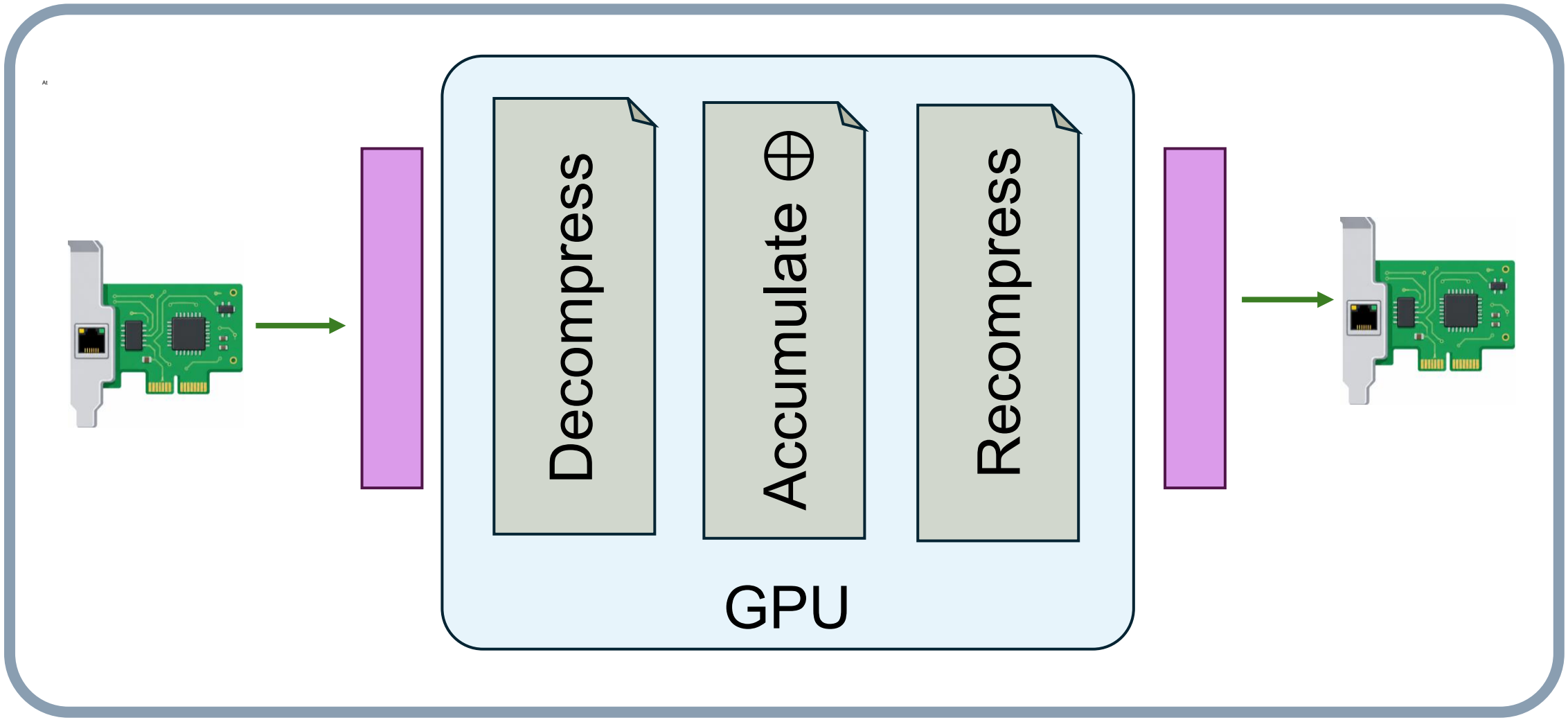
Idea: allocate more bits to more important entries

Challenges

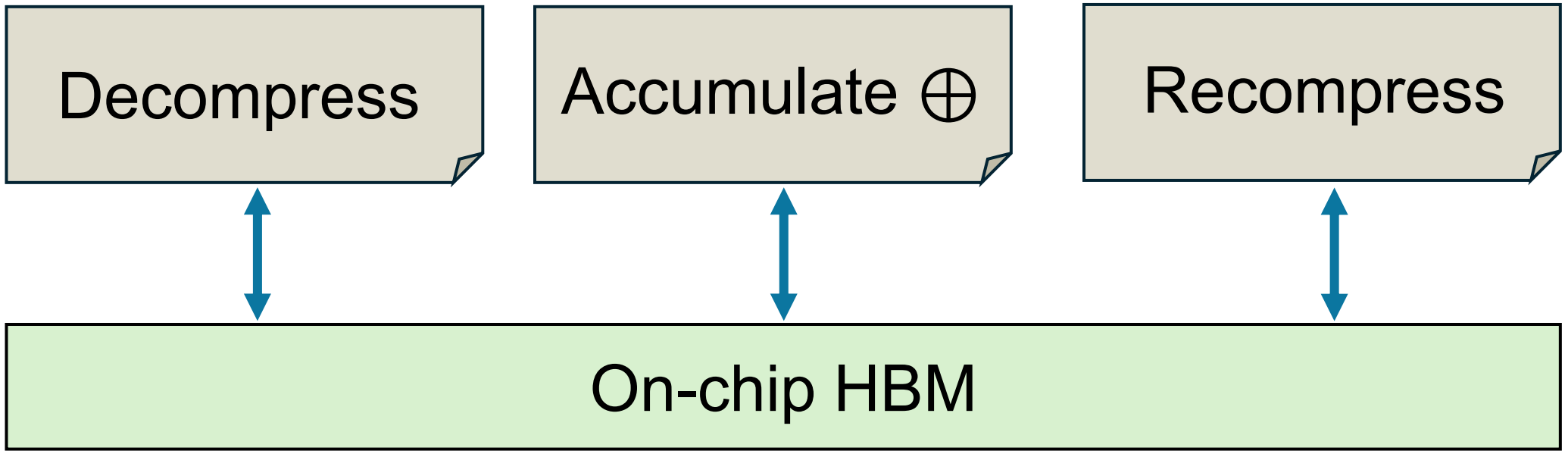
At

- Compression error that accumulates over hops
- Computational overhead for de/recompress

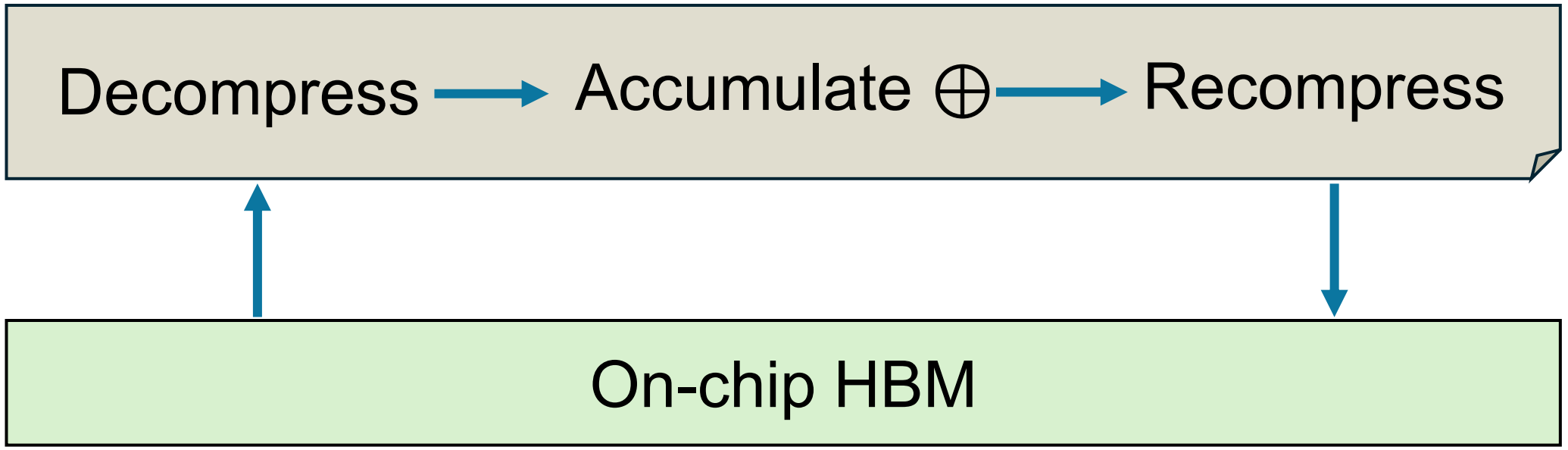
Computational Overhead



Optimization 1: Fused Kernels.

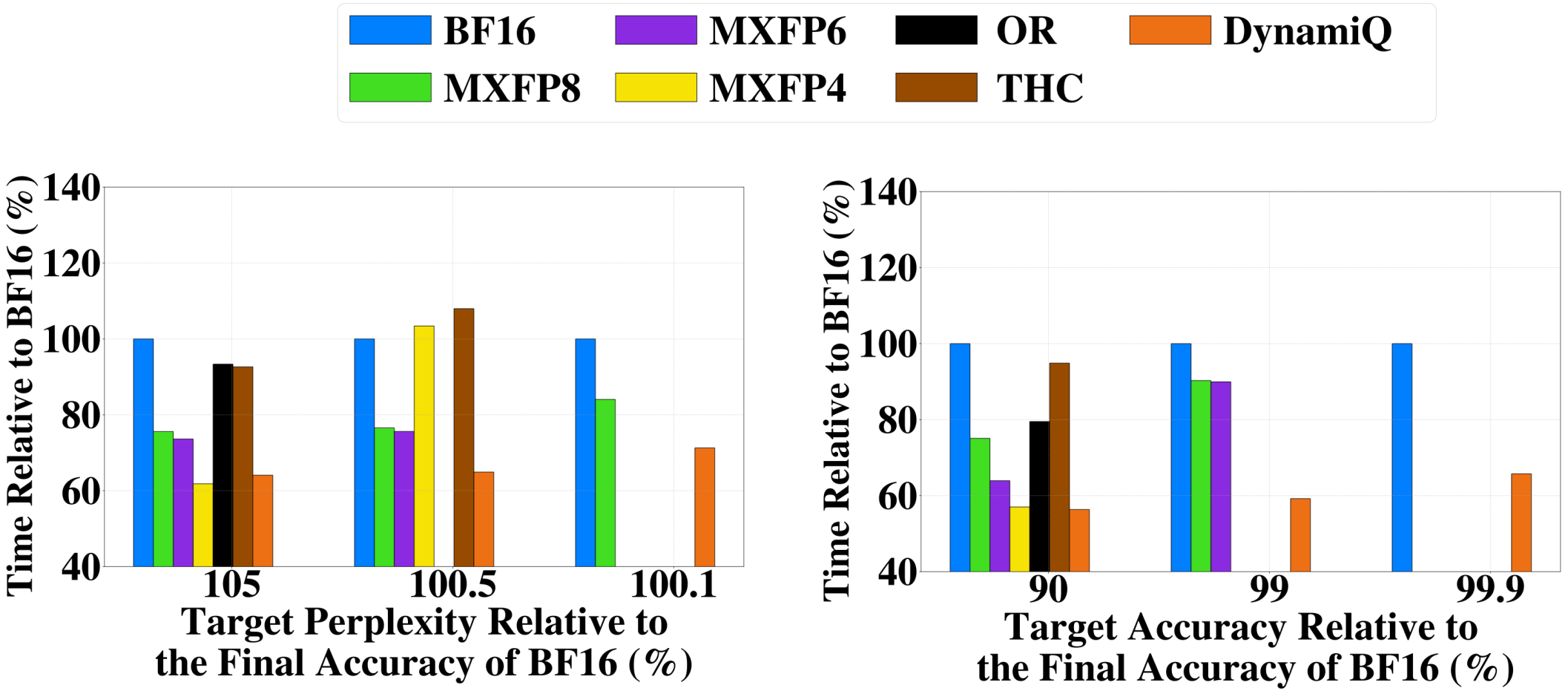


Optimization 1: Fused Kernels.



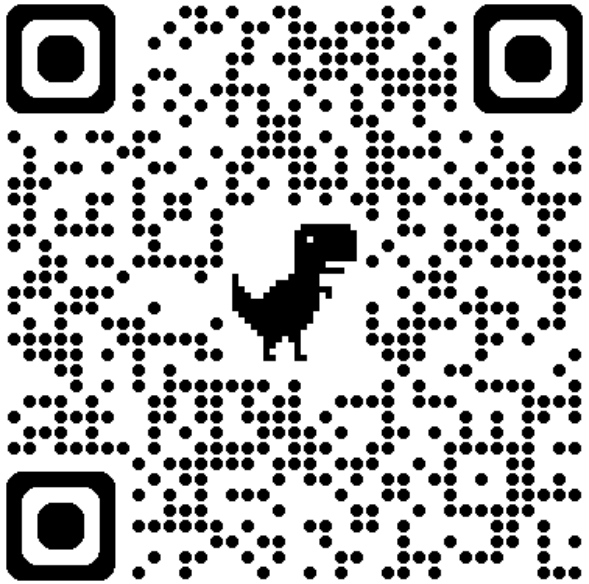
2-3x speedup

Main Evaluation Results

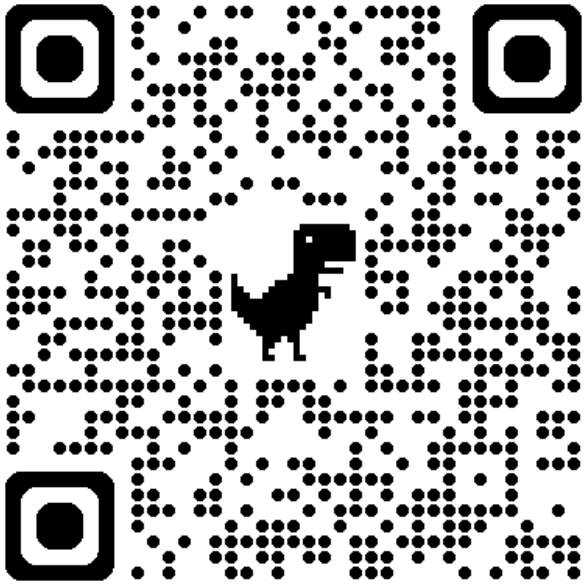


Thank you!

- Multi-hop all-reduce with compression is now practical with DynamiQ.



Our Paper



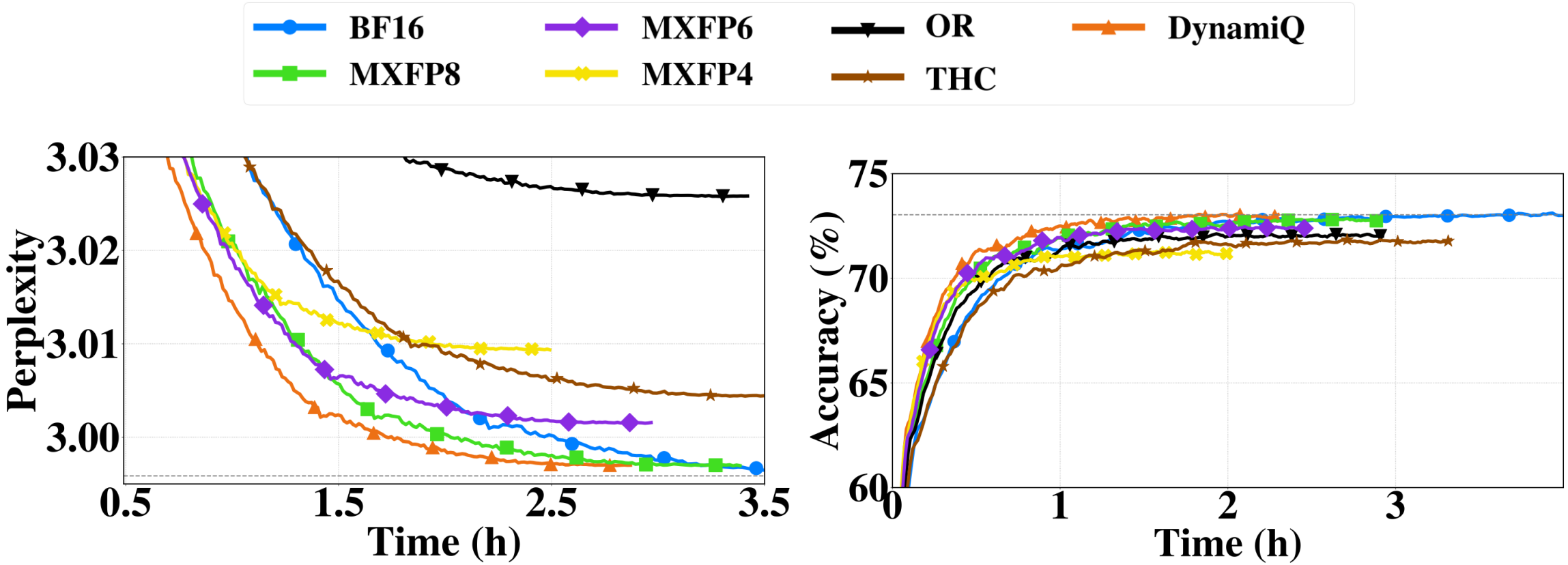
Our Code

References

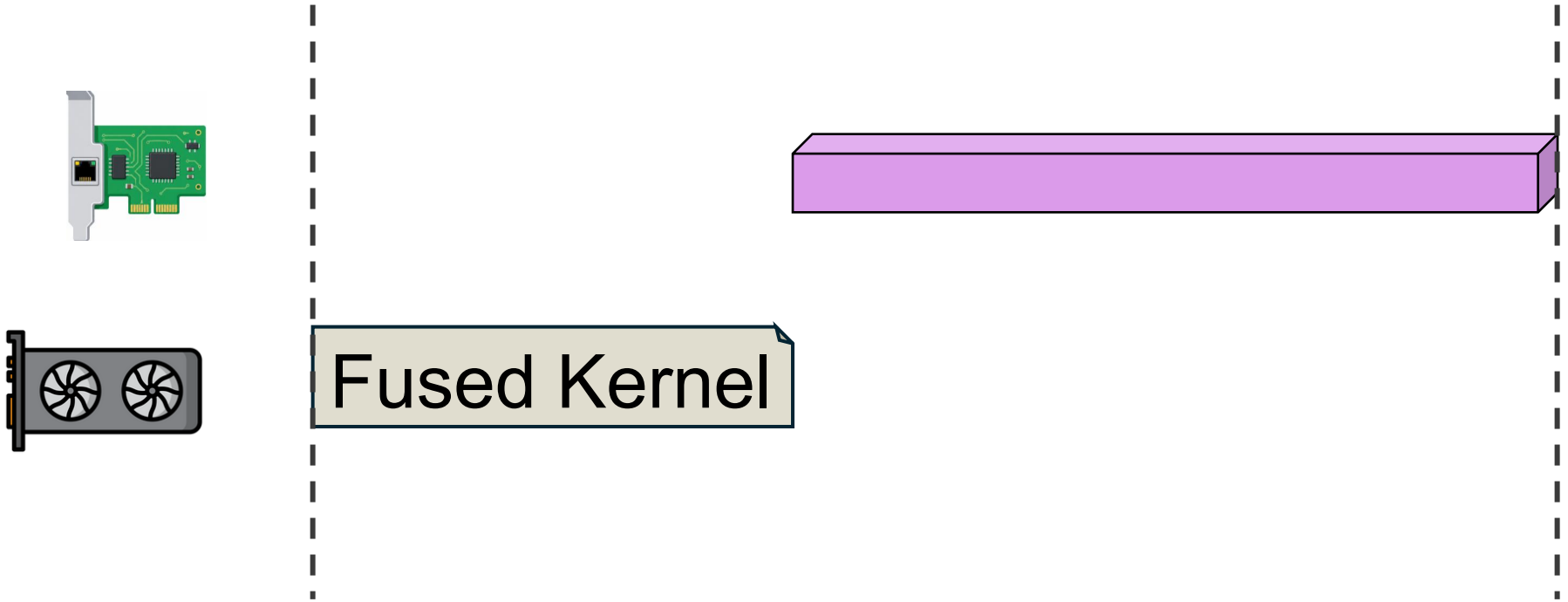
[1] Wang et al. ZeRO++: Extremely Efficient Collective Communication for Giant Model Training. Microsoft Research.

[2]. Han et al. Beyond Throughput and Compression Ratios: Towards High End-to-end Utility of Gradient Compression. In HotNets ’24.

Main Evaluation Results

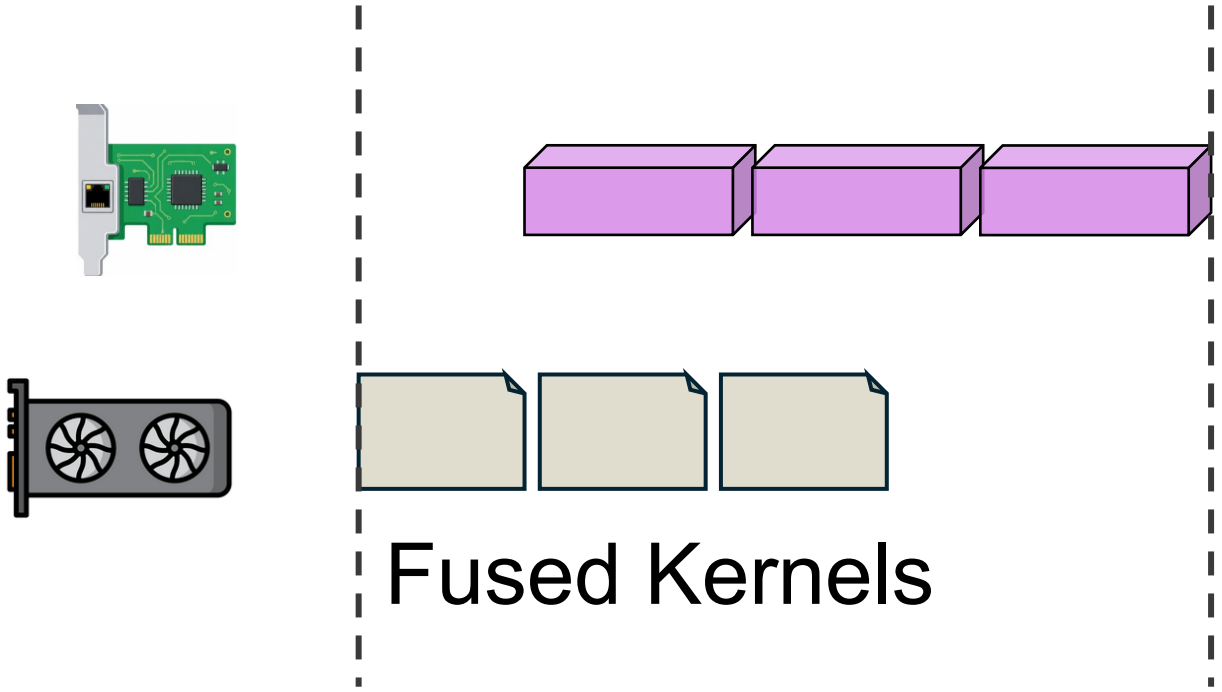


Optimization 2: Pipelined Communication



Serialized computation and communication

Optimization 2: Pipelined Communication



Pipelined, blockwise computation and communication

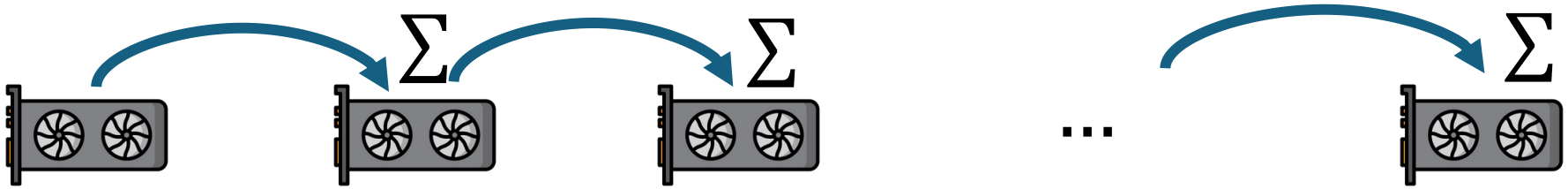
Why This Works: Intuition

- Each group i contributes to $e_i = \frac{|g_i|^2}{4^{b_i}}$.
- We want to minimize $\sum e_i$ subject to $\frac{1}{\mathrm{n}}\sum b_i = \bar{b}$
- Intuitively, allocating bits to groups with larger values contributes to more error reduction.

Summary of Additional Results

- Running background jobs over shared network.
- Experiments with butterfly all-reduce.
- Scalability analysis up to 8192 workers.

The Main Challenge: Overflows.



- Partial-sum range grows linearly along the AR topo.
- With limited bit-width overflows happen.