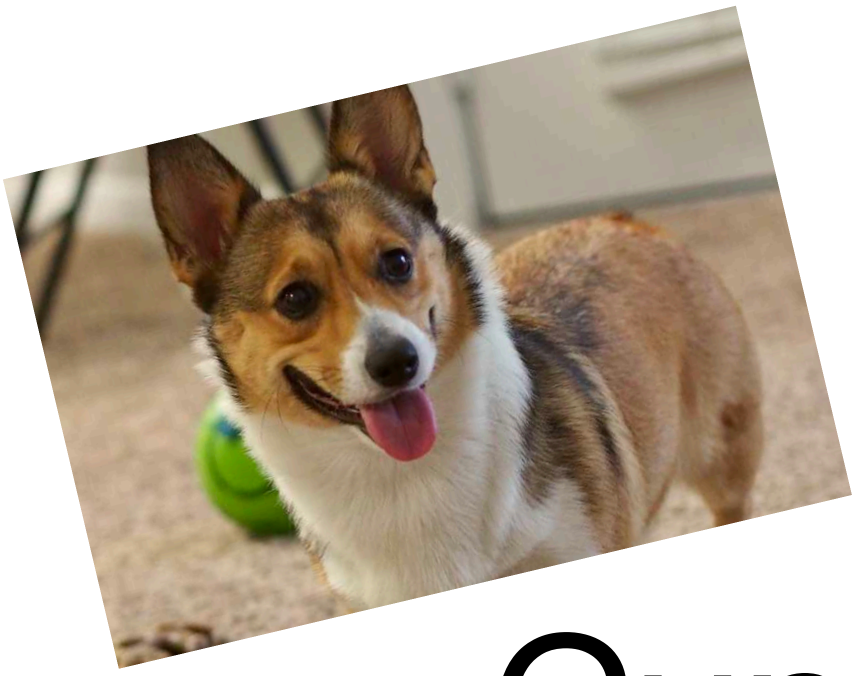


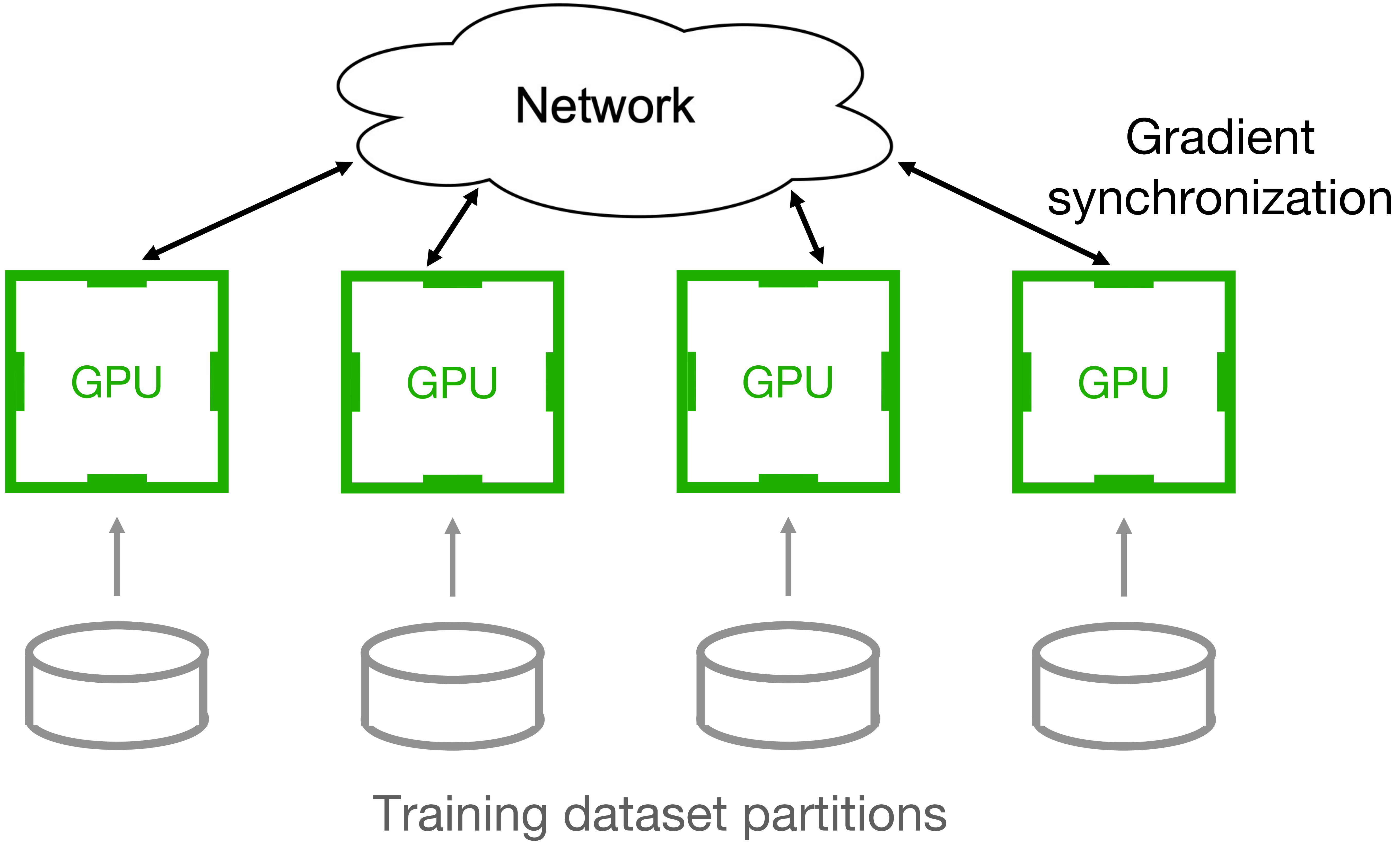
Cupcake: A compression optimizer for scalable communication-efficient distributed training

Zhuang Wang, **Xinyu Crystal Wu**, Zhaozhuo Xu and T. S. Eugene Ng

Big Data and Optical Lightpaths Driven Lab

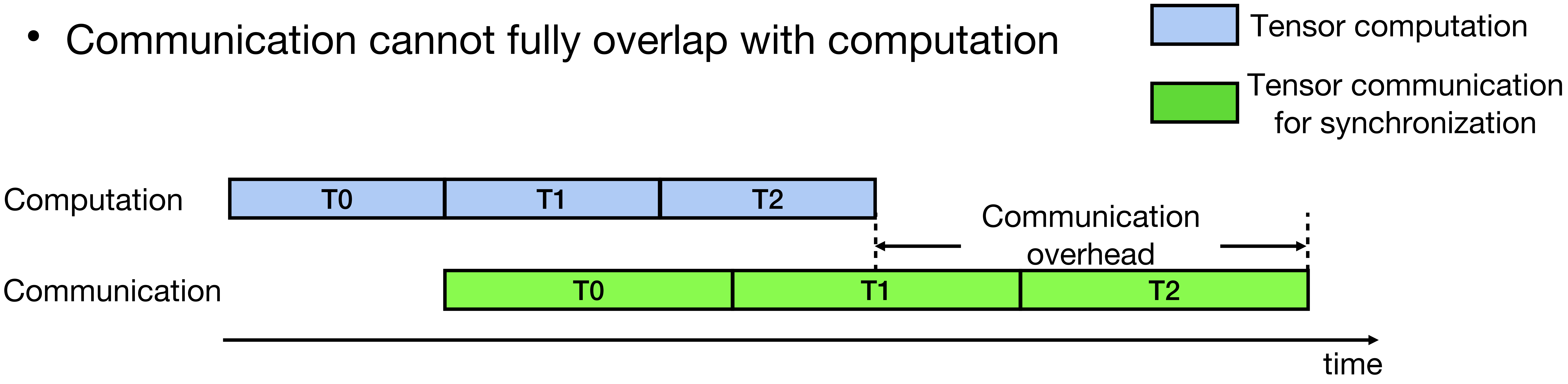
Distributed deep learning

- Gradient synchronization among GPUs



Communication is bottleneck

- Communication cannot fully overlap with computation



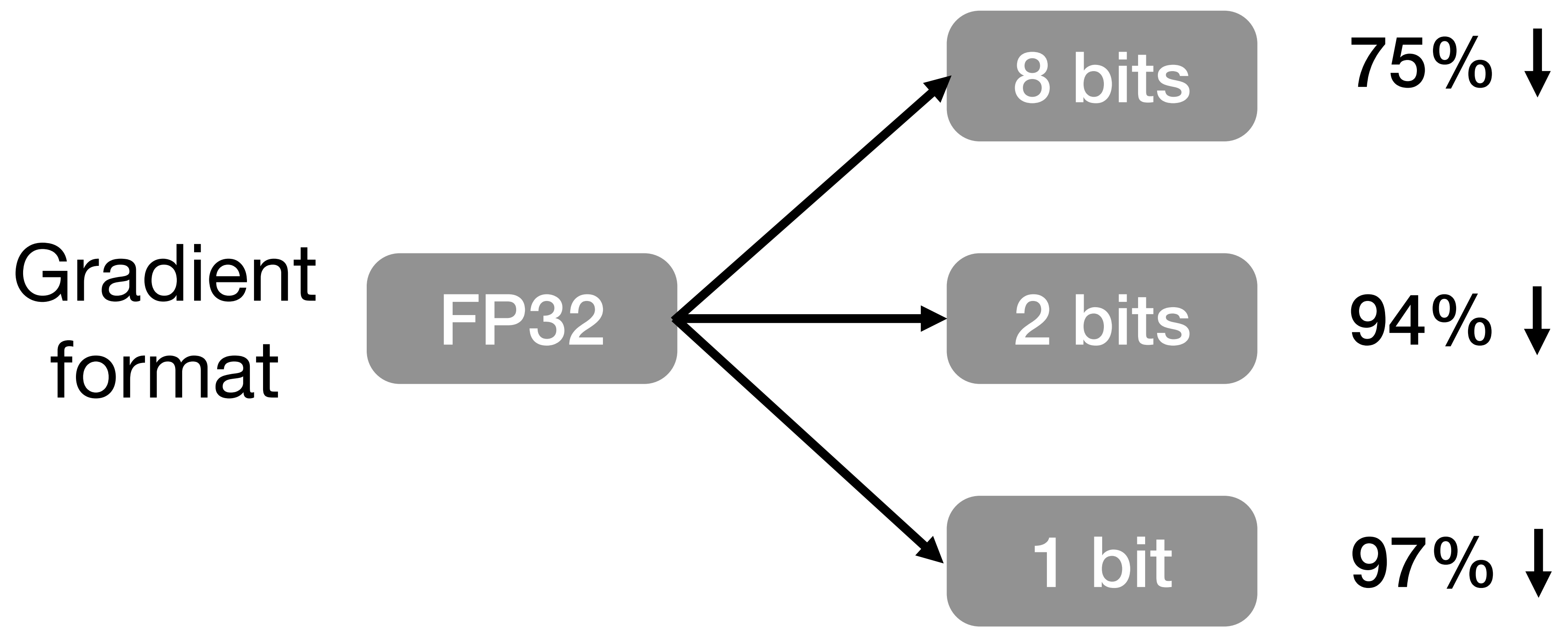
- Communication overhead can account for more than **50%** of training time [1]

[1] Gradient Compression Supercharged High-Performance Data Parallel DNN Training, SOSP ’21

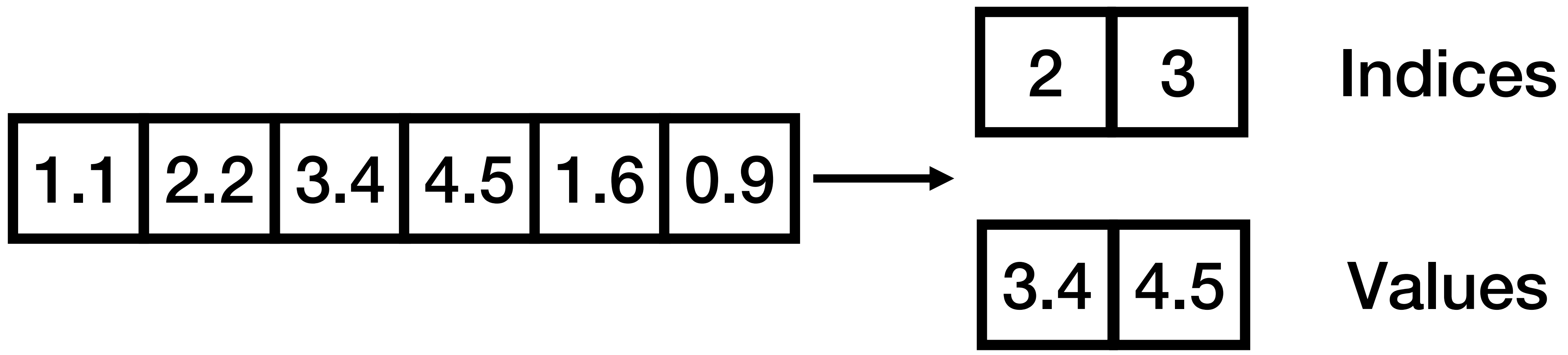
Gradient compression (GC)

- GC shrinks communicated traffic volume
  - has negligible impact on model accuracy [1]

- Quantization



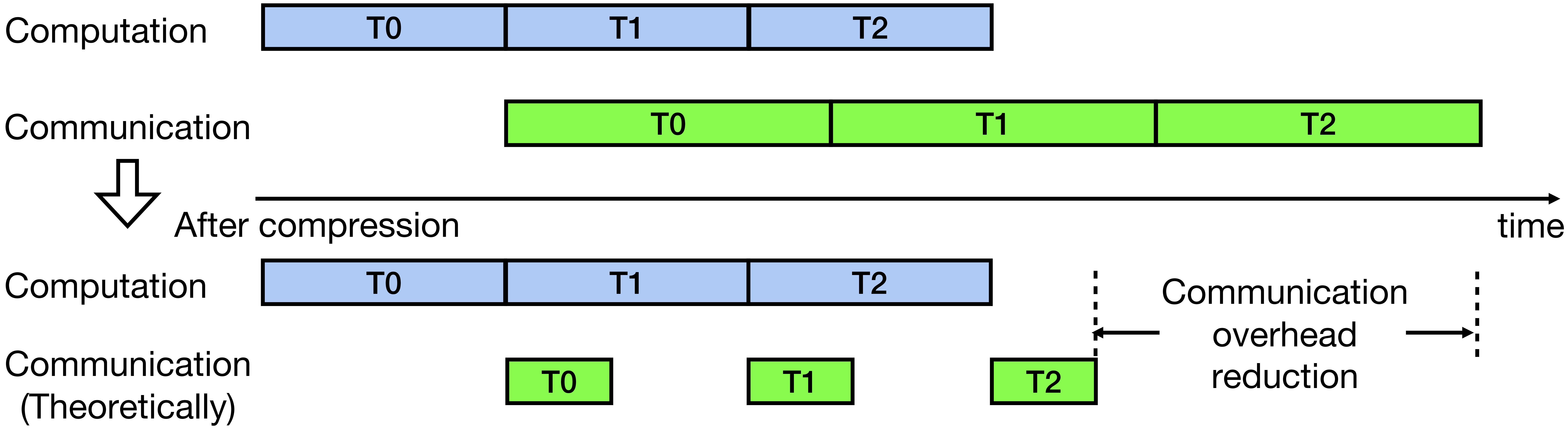
- Sparsification
  - A subset of gradients
  - Save > 99% traffic volume [2]



[1] GRACE: A compressed communication framework for distributed machine learning, ICDCS ‘21
[2] DRAGONN: Distributed Randomized Approximate Gradients of Neural Networks, ICML ‘22

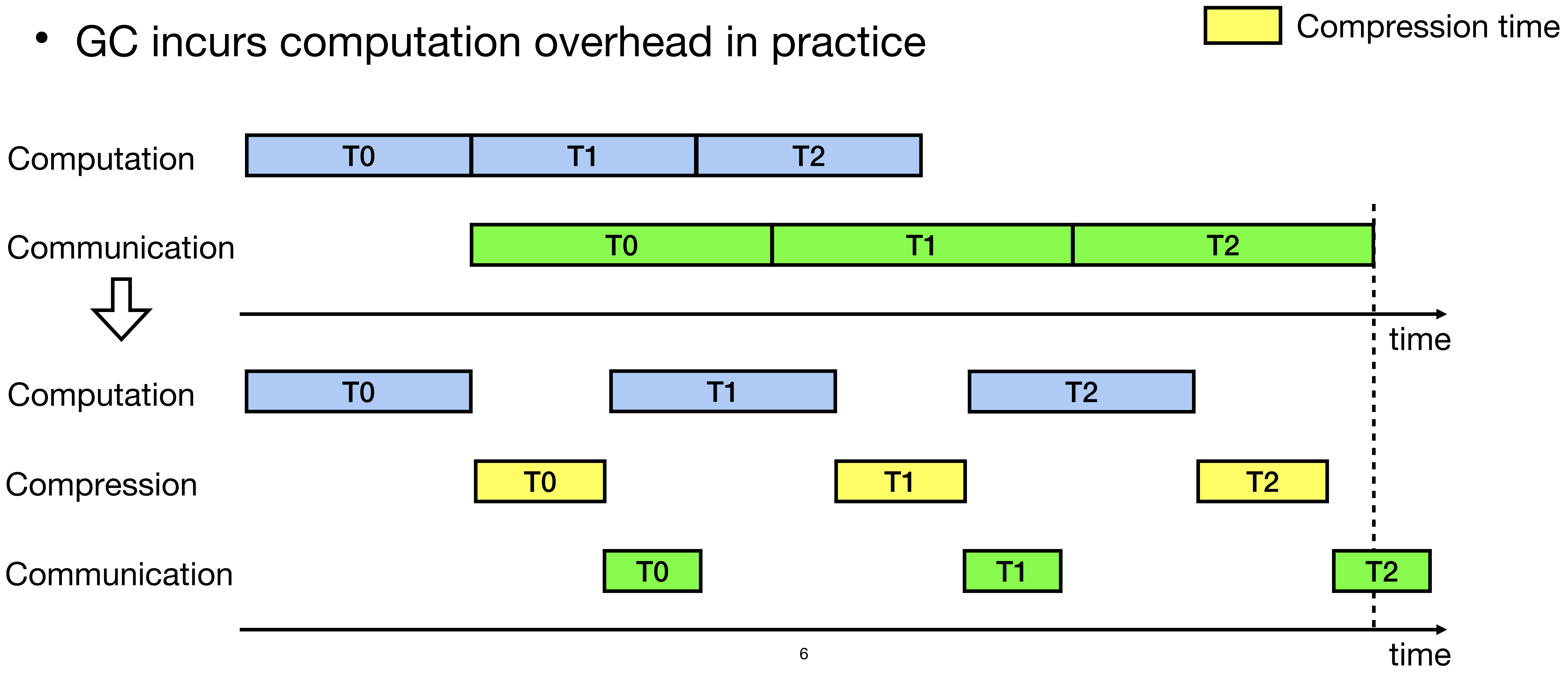
Gradient compression (GC) in theory

- GC reduces communication overhead



Gradient compression (GC) in reality

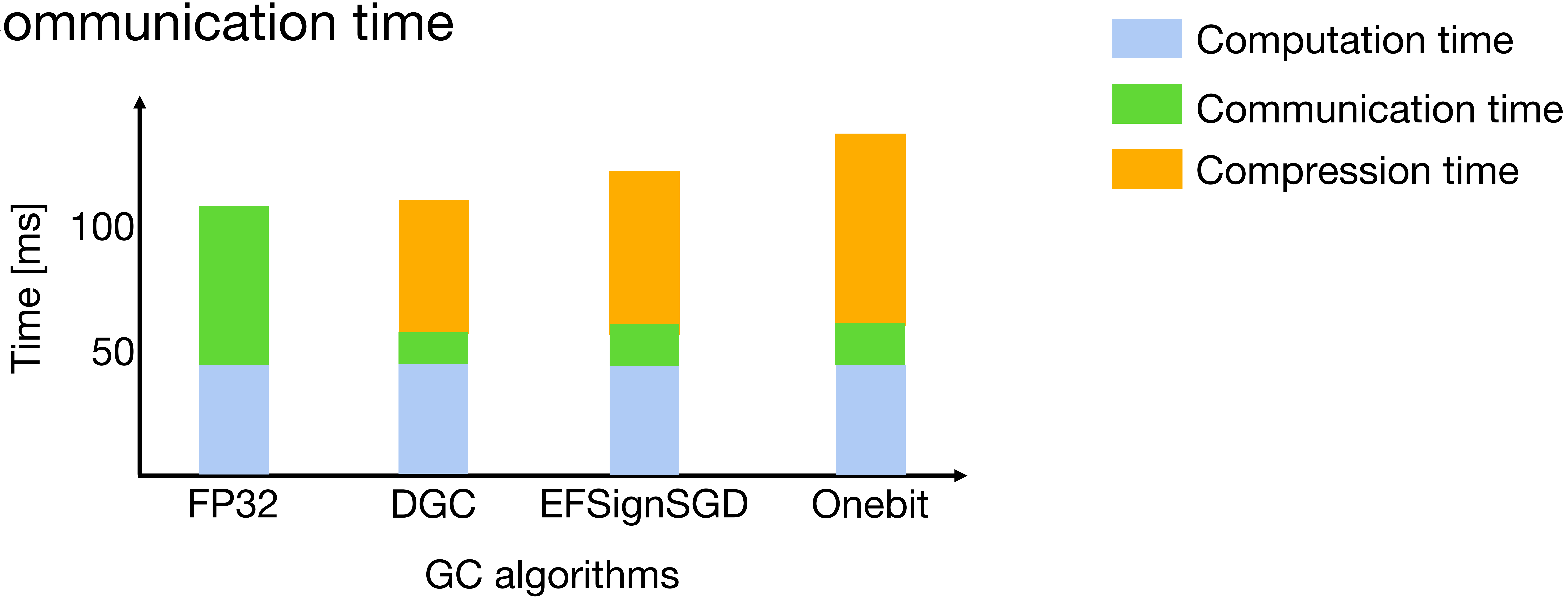
- GC incurs computation overhead in practice



Compression is costly

Iteration time breakdown

- GC reduces communication time



- GC incurs significant compression overhead

Why is compression costly?

- Two additional operations

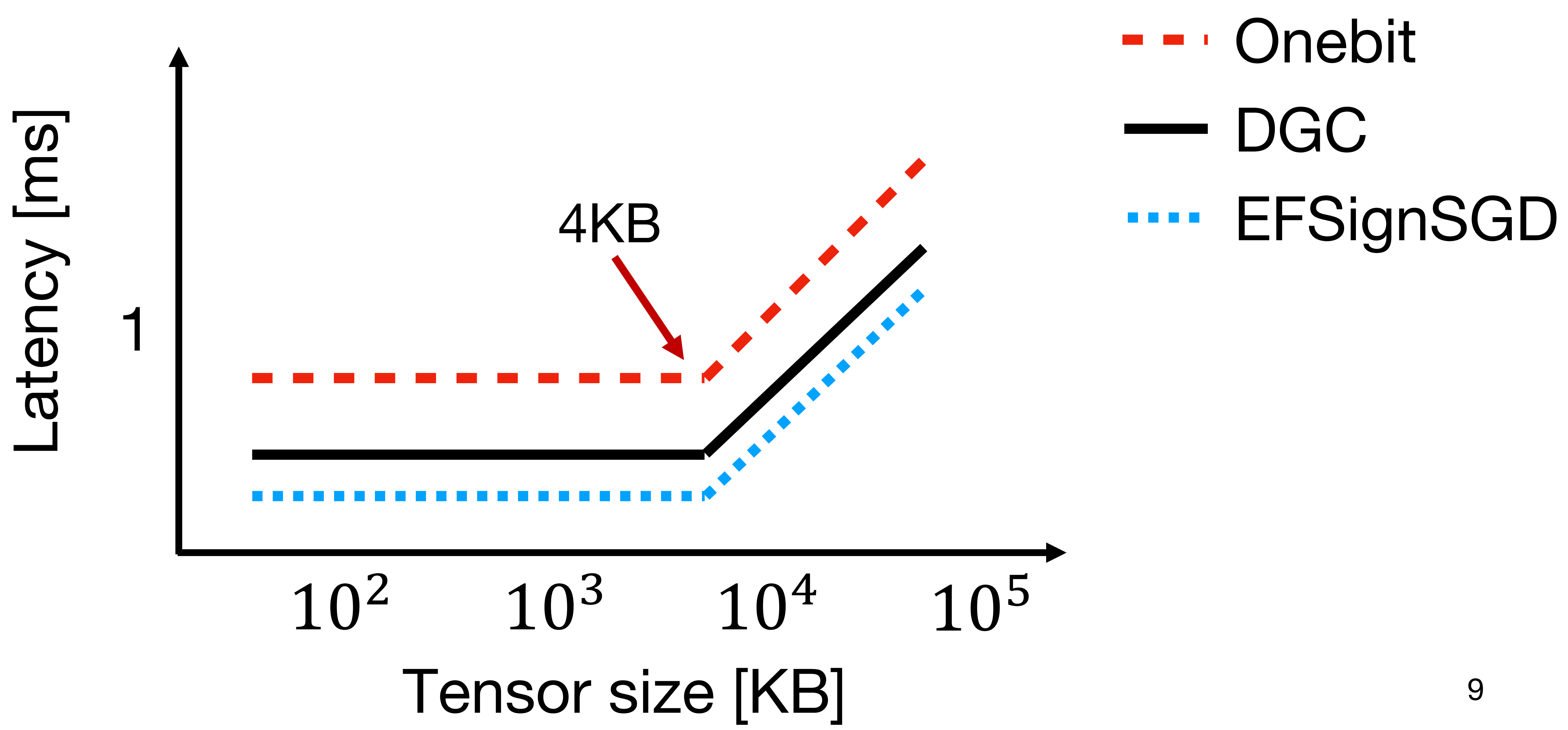
Tensor → Communication

Why is compression costly?

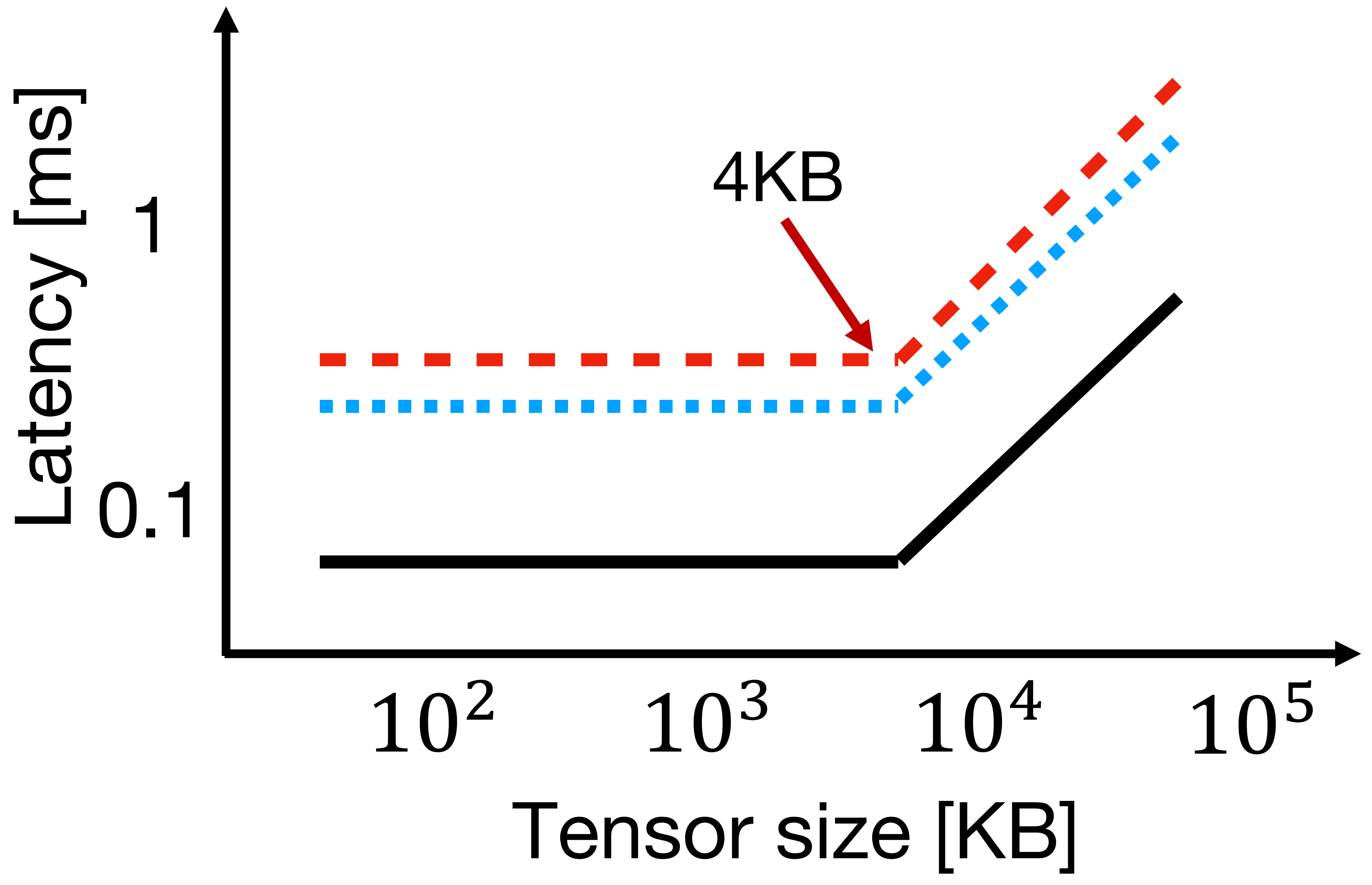
- Two additional operations

Tensor → Compress → Communication → Decompress

- Compress overhead

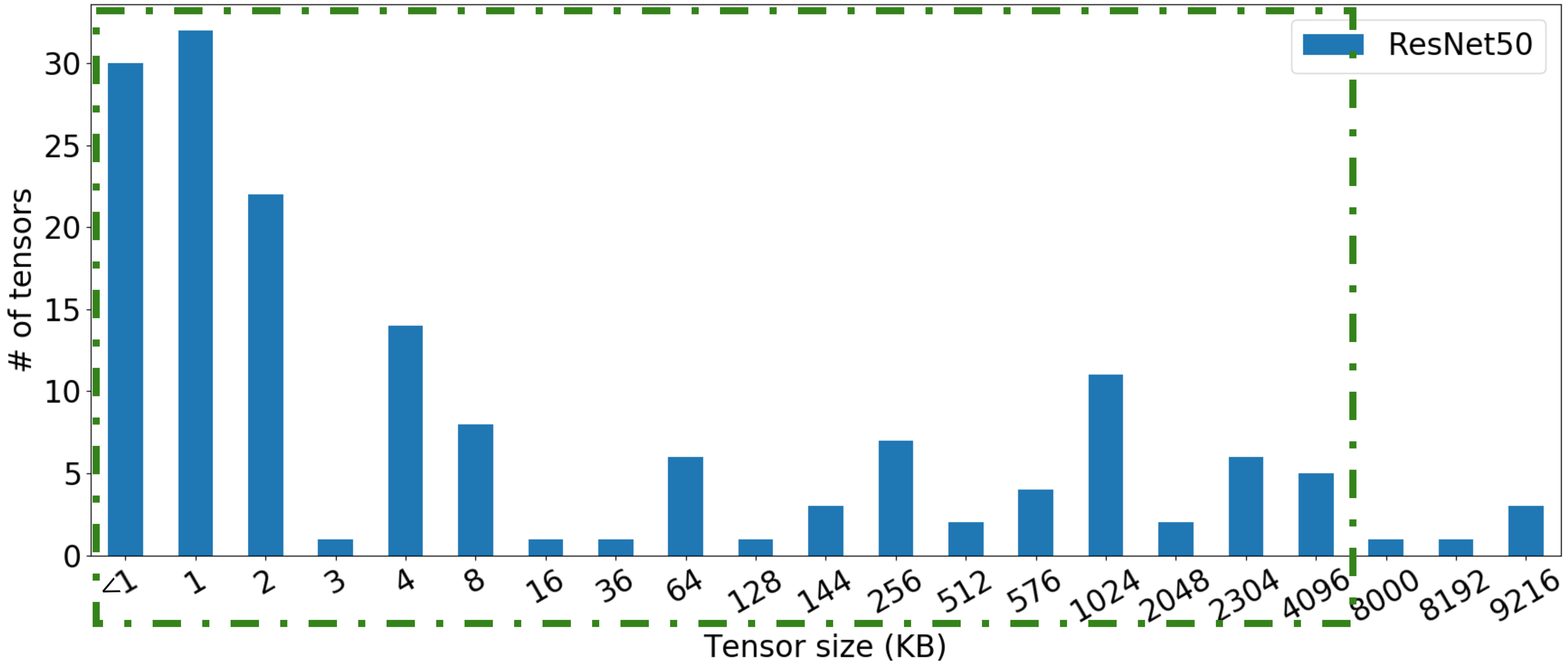


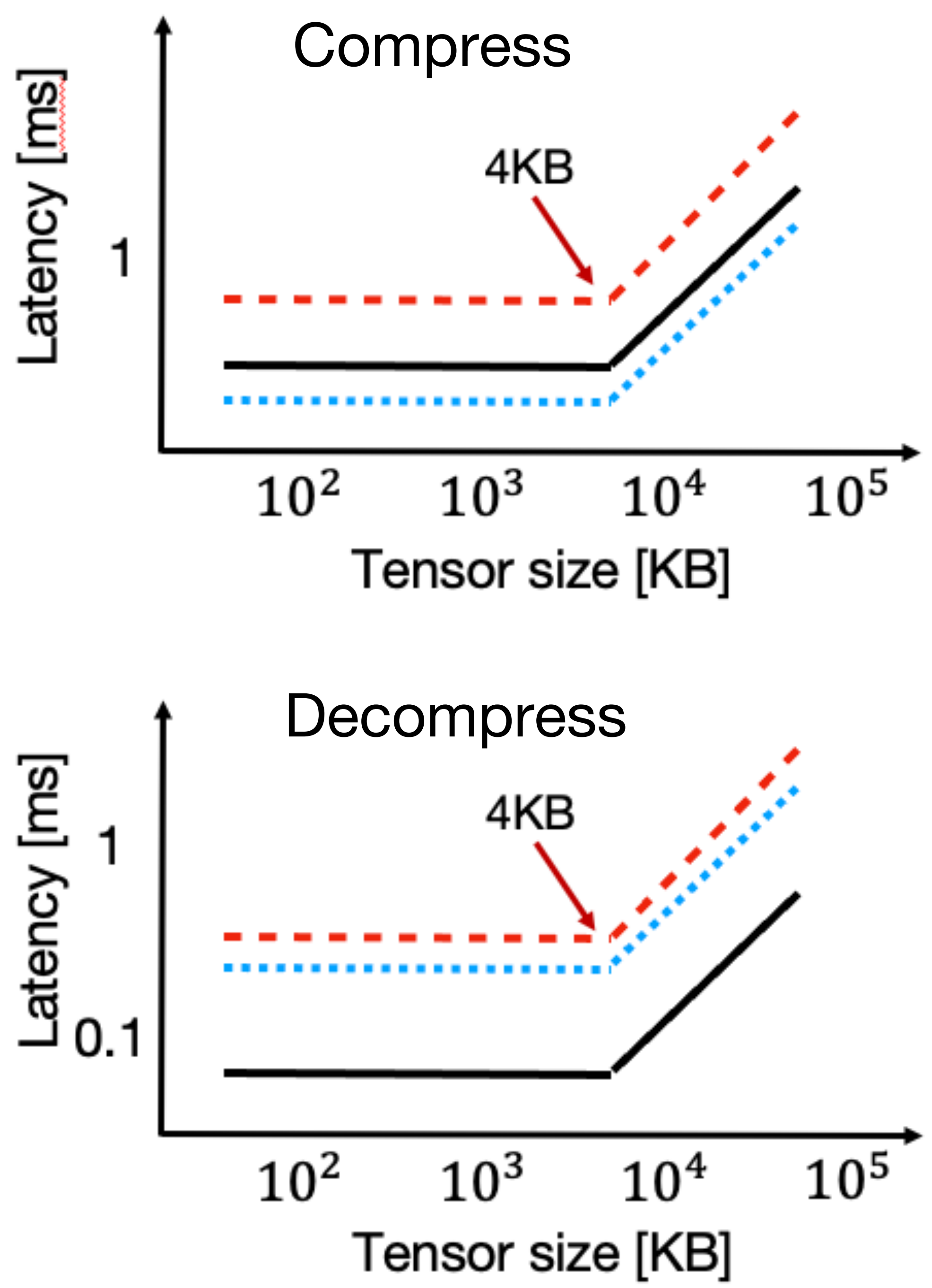
- Decompress overhead



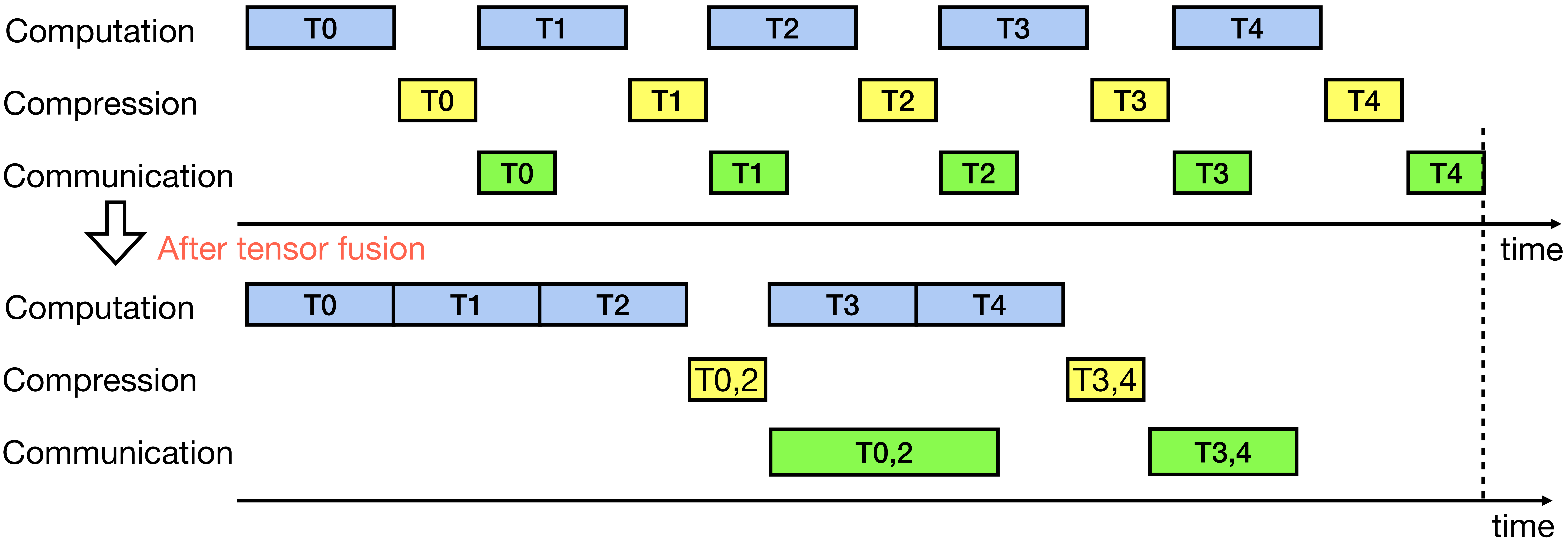
Why is compression costly?

- Existing approach to compress tensors
  - Tensor by tensor
  - Invokes compress and decompress operations for each tensor
- Many small tensors in DNN models



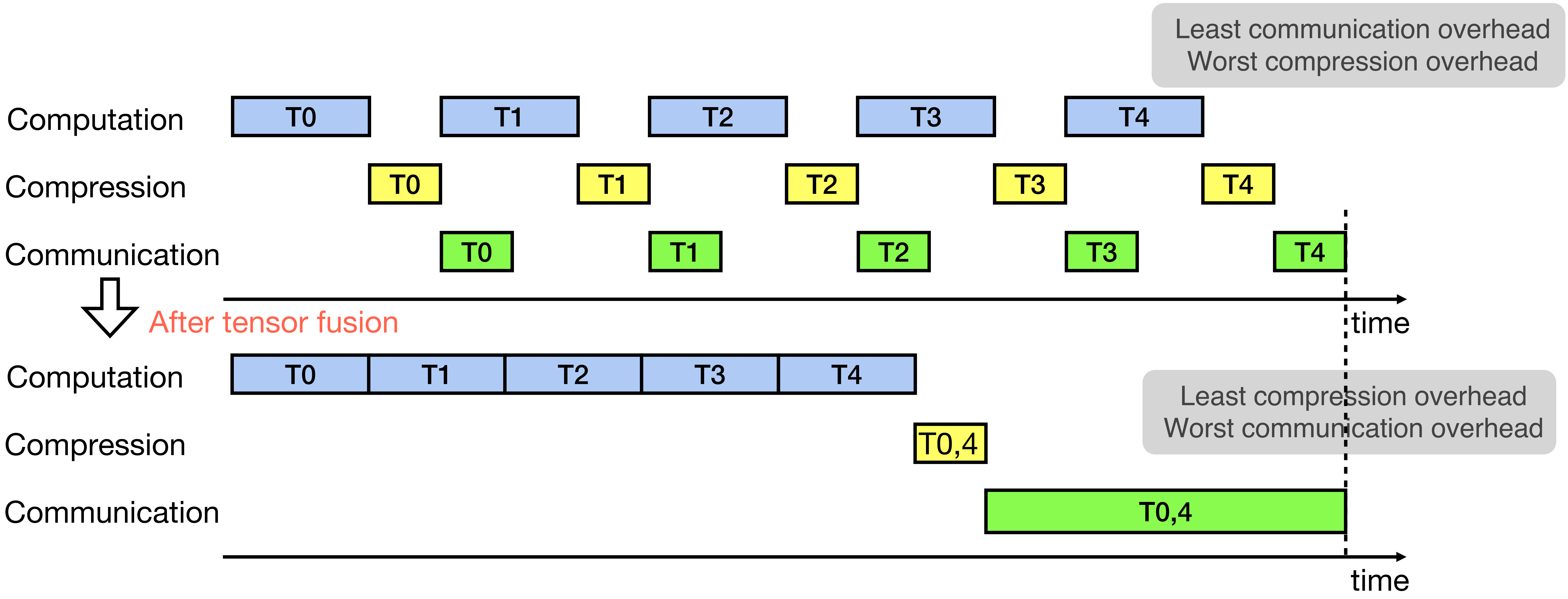


Fuse tensors to reduce compression overhead



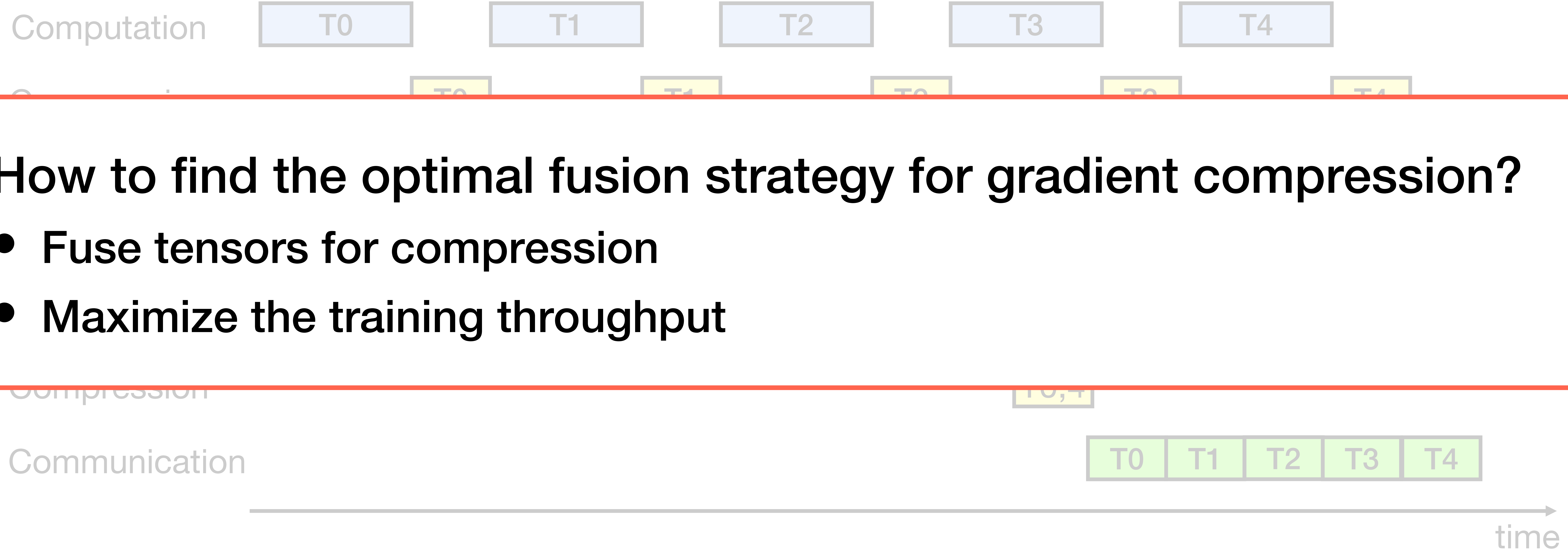
Challenges

Trade-off between compression and communication overhead



Challenges

Trade-off between compression and communication overhead



How to find the optimal fusion strategy for gradient compression?

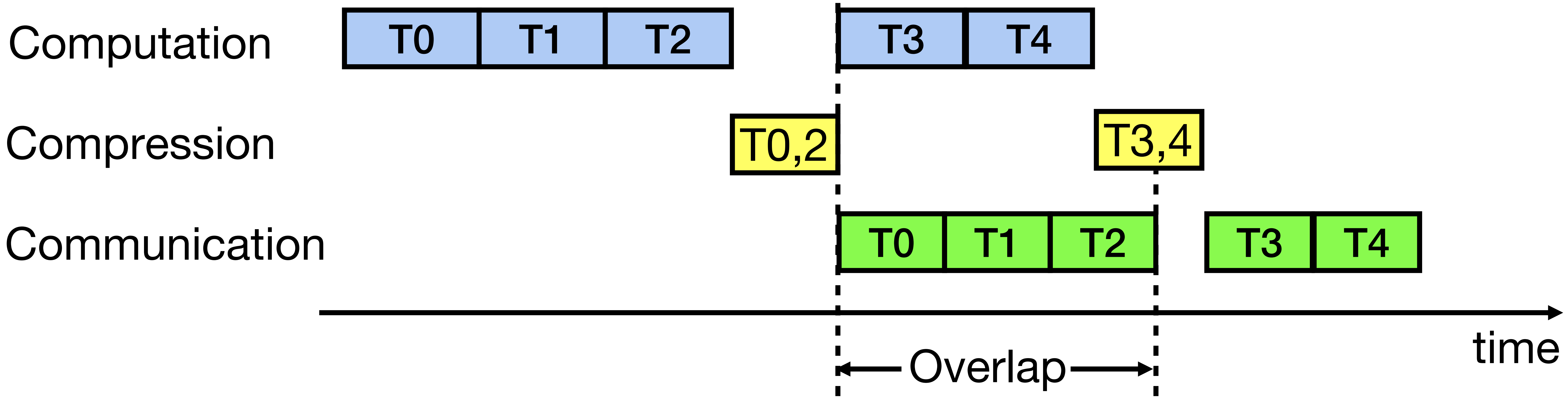
- Fuse tensors for compression
- Maximize the training throughput

Cupcake

Search for the optimal fusion strategy

- Formulation of the iteration time

Iteration time = Computation + Communication + Compression − Overlapping



Cupcake

Empirical measurements

- Expensive to test all fusion strategies with end-to-end training
- Our solution
  - Use measurements from production environment to model training process
  - Profile offline based on the system configurations
    - GPU computation capacity, the number of GPUs, and the network bandwidth

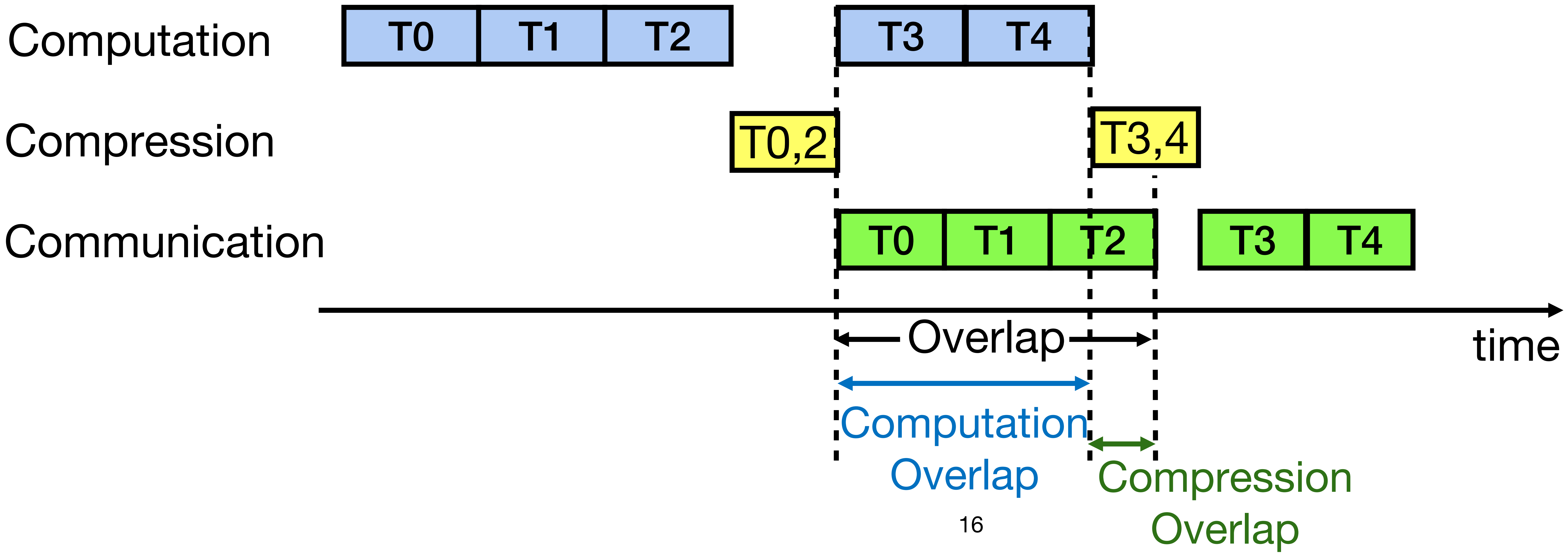
Tensor Computation time (forward/backward propagation)	Tensor Communication time (startup/transfer time)	Tensor Compression time (kernel/compress time)

  - Derive the timeline of training with any strategy

Cupcake

Determine overlap for fusion strategies

- Overlapping is specific to each fusion strategy
- Overlapping time is determined by the intricate interactions among tensors
    - Communication can overlap with both computation and compression



Cupcake

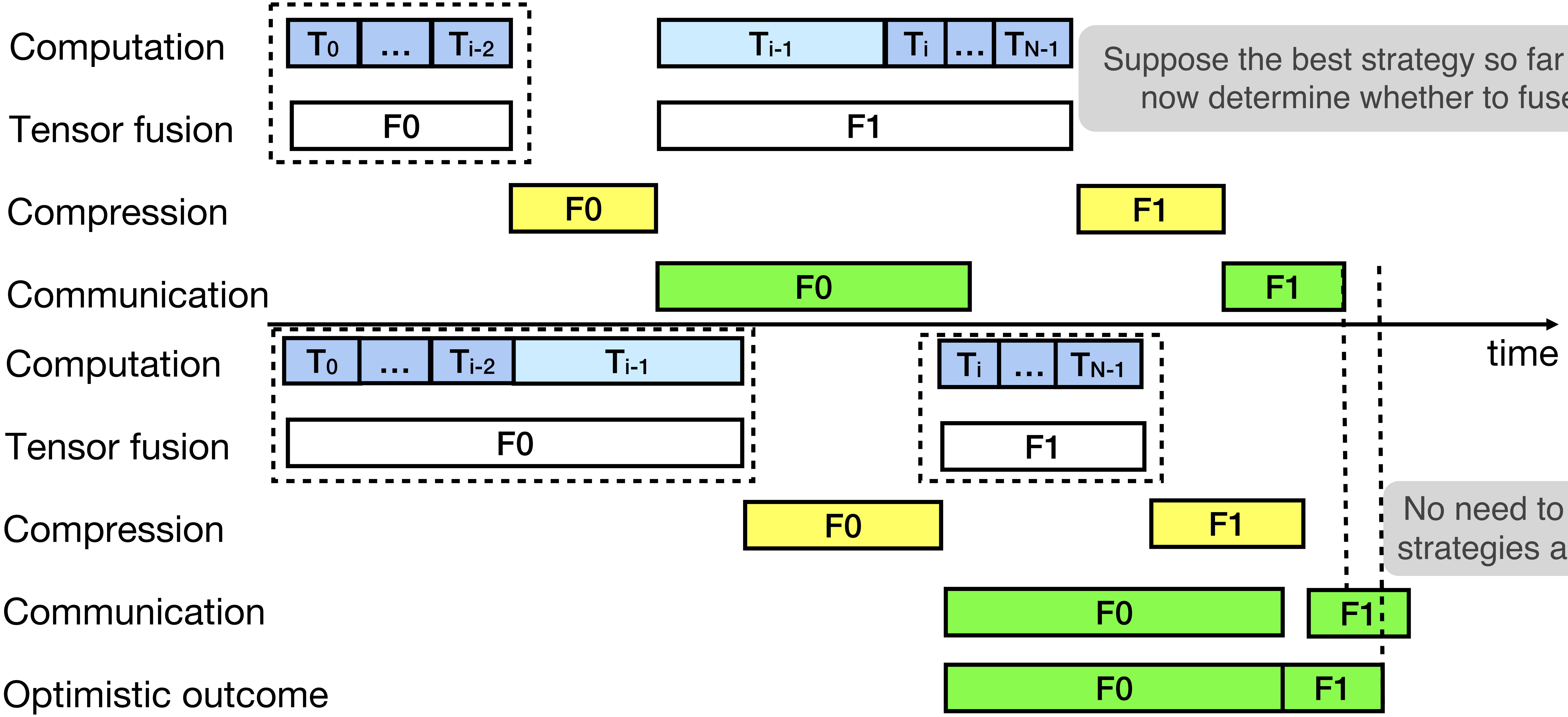
Search space

- A brute-force method will take exponential time

Time complexity: O(2^N)

Pruning techniques

No need to examine all cases for the formation of F_0



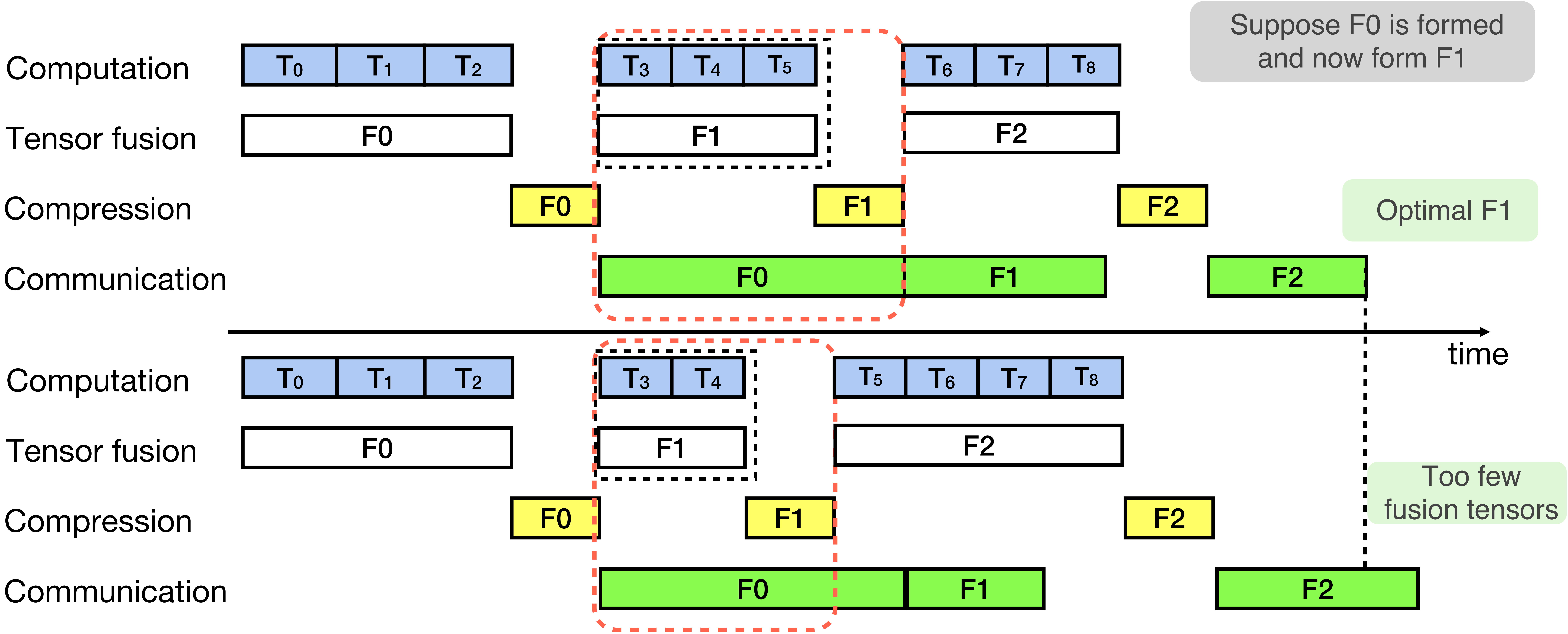
Suppose the best strategy so far fuses $\{T_0, T_{i-2}\}$; now determine whether to fuse T_{i-1} into F0

No need to explore fusion strategies among $\{T_i, T_{N-1}\}$

Prune a strategy if its optimistic outcome is greater than the best so far

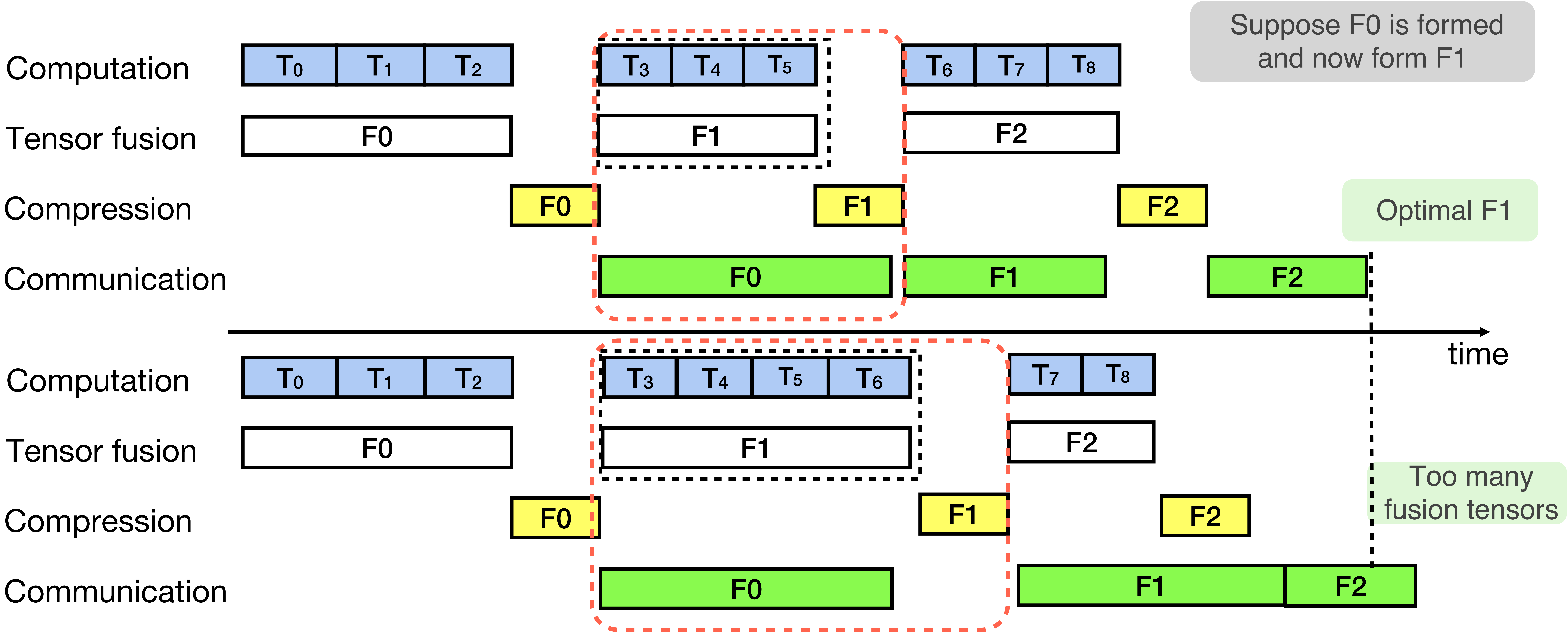
Pruning techniques (cont'd)

Fuse tensors to maximize the overlapping time



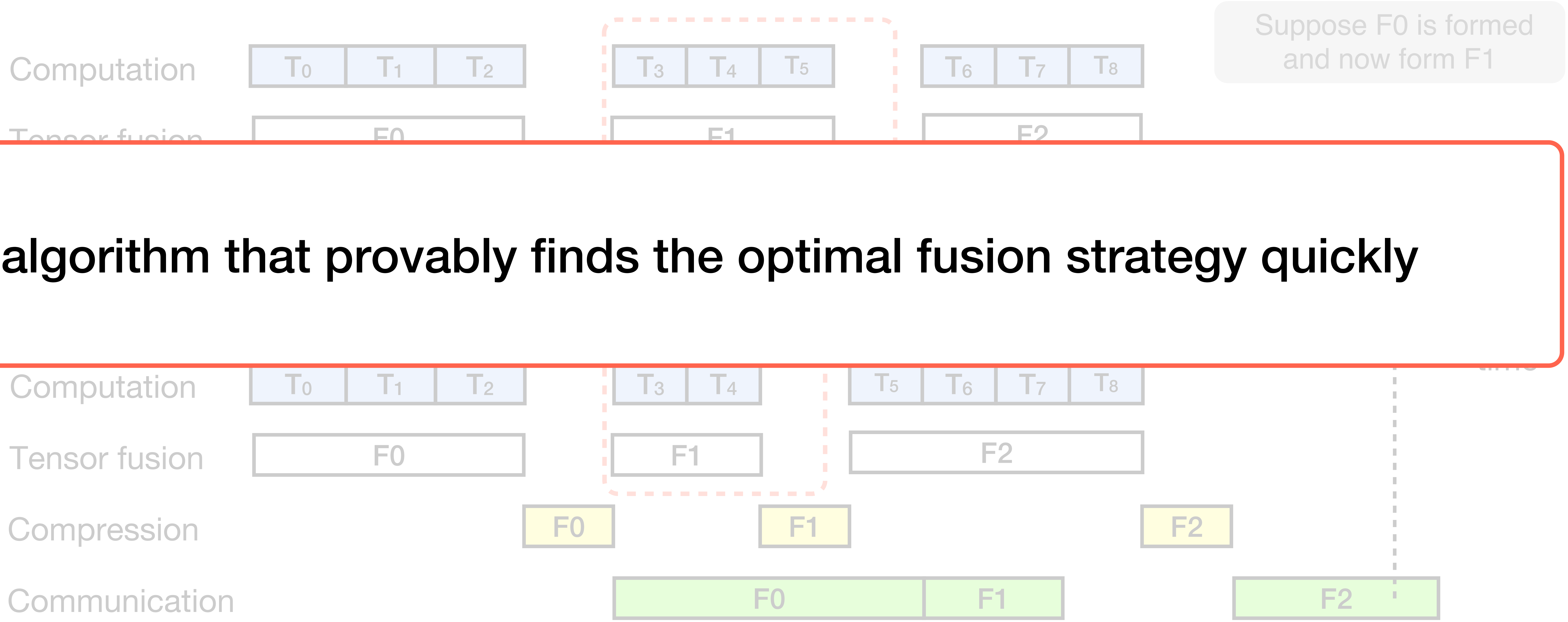
Pruning techniques (cont'd)

Fuse tensors to maximize the overlapping time



Pruning techniques (cont'd)

Fuse more tensors based on the communication progress

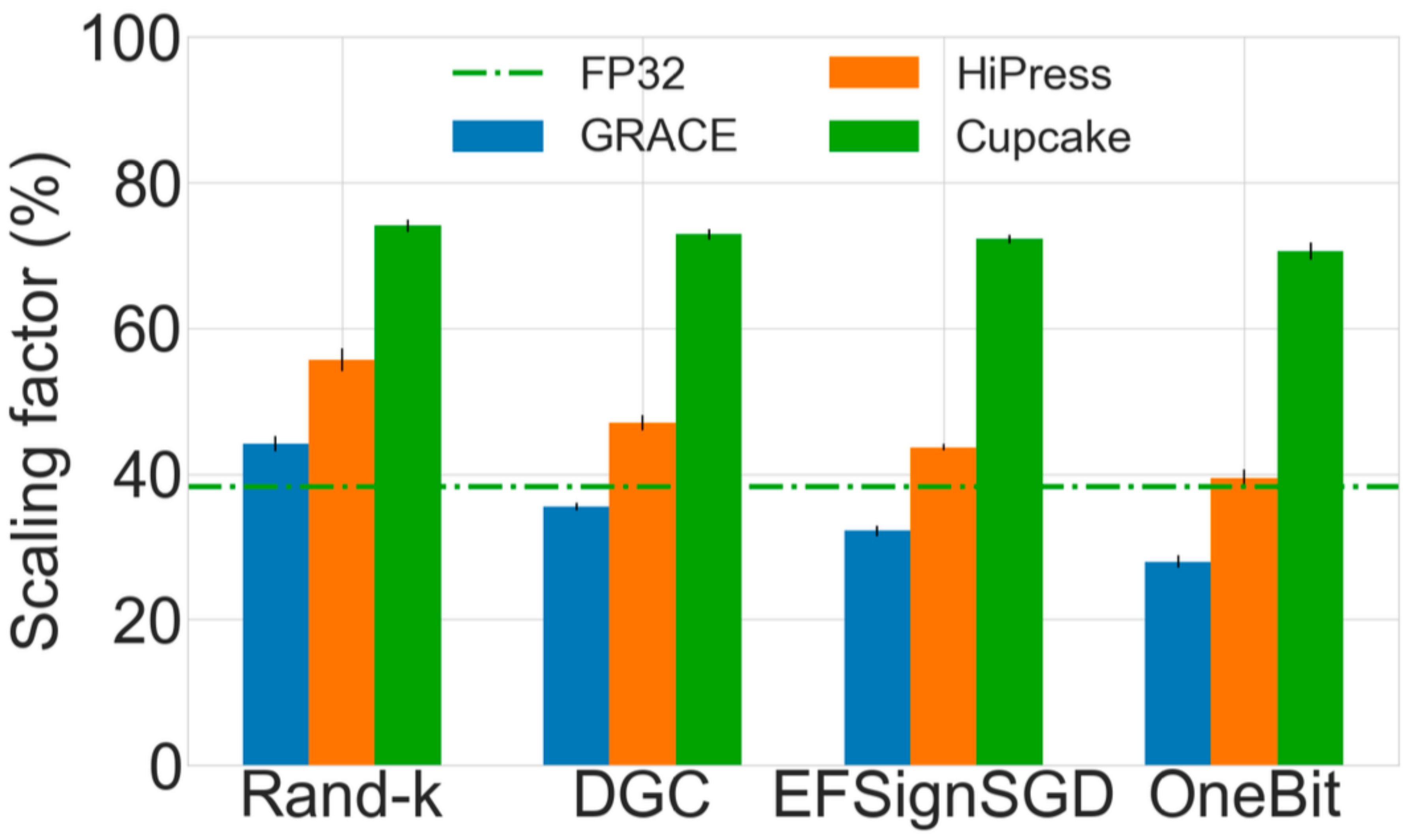


An algorithm that provably finds the optimal fusion strategy quickly

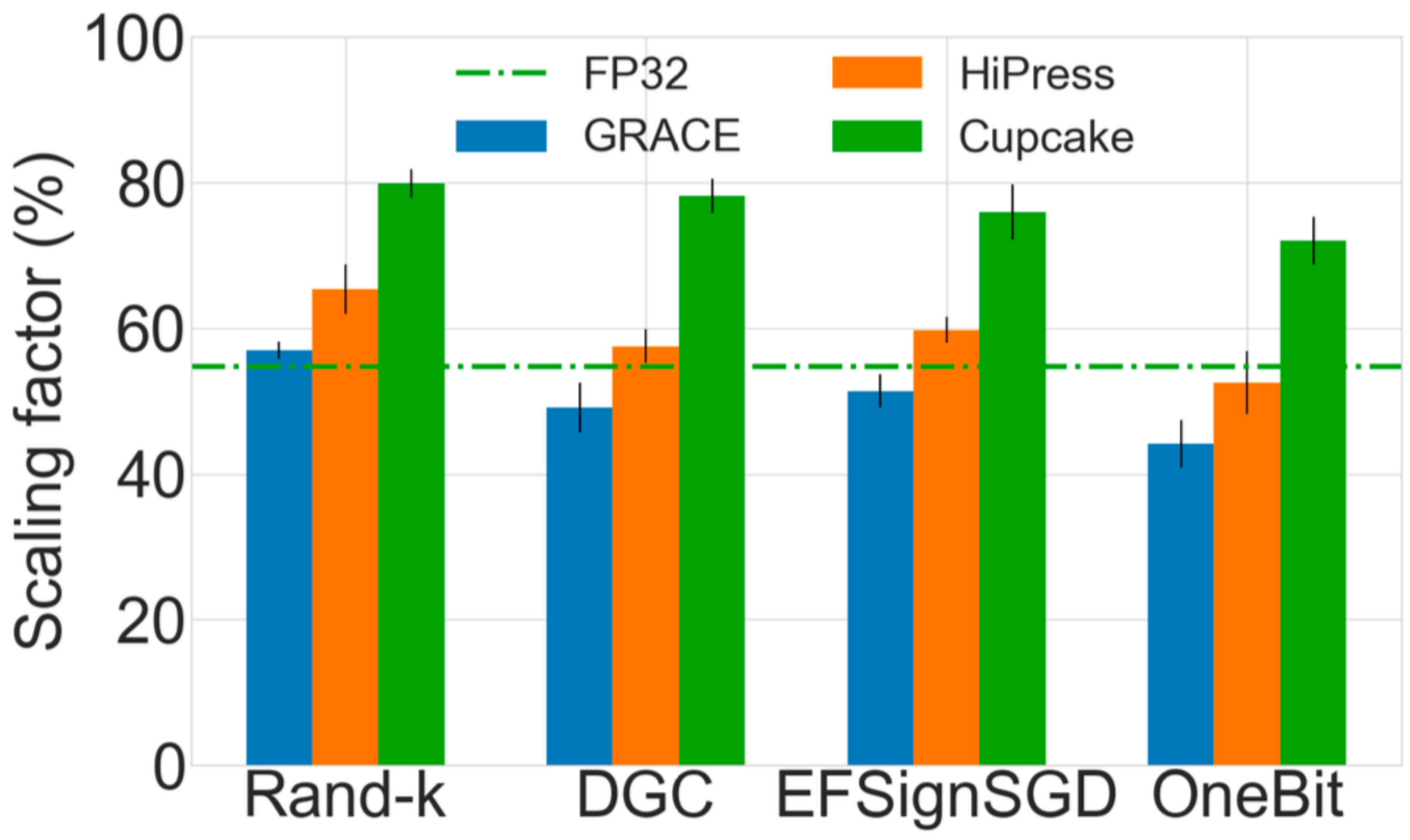
Results

25Gbps network, NVLink

- 8 GPU machines and each machine has 8 V100 GPUs



(a) ResNet50



(b) ResNet101

Up to 79% improvement

Results

25Gbps network, NVLink

- Training accuracy

Model	Dataset	GC	GRACE	Cupcake
ResNet50	CIFAR10	DGC	93.2%	93.2%
ResNet101	ImageNet-1K	EFSignSGD	76.6%	76.7%

Summary

- Layer-wise compression fashion causes prohibitive compression overhead
- Cupcake applies GC algorithms in a *fusion* fashion
- Provably find the optimal fusion strategy to maximize training throughput

Thank you!

(Xinyu Crystal Wu: xw64@rice.edu)