SAFusion: Efficient Tensor Fusion with Sparsification Ahead for High-Performance Distributed DNN Training

Zhangqiang Ming[1,2], Yuchong Hu[1], Xinjue Zheng[1],
Wenxiang Zhou[1], Dan Feng[1]

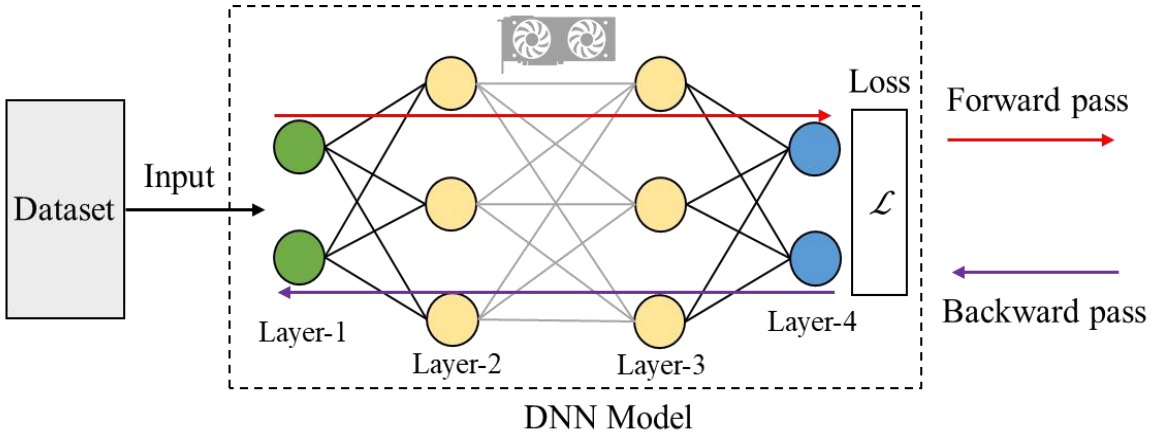
[1]Huazhong University of Science and Technology, Wuhan 430074, China
[2]Innovation Research Institute of Cethik Group Co., Ltd, Hangzhou 311100, China

華中科技大學
HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

CETC
中电海康集团有限公司

Background

Distributed DNN Training

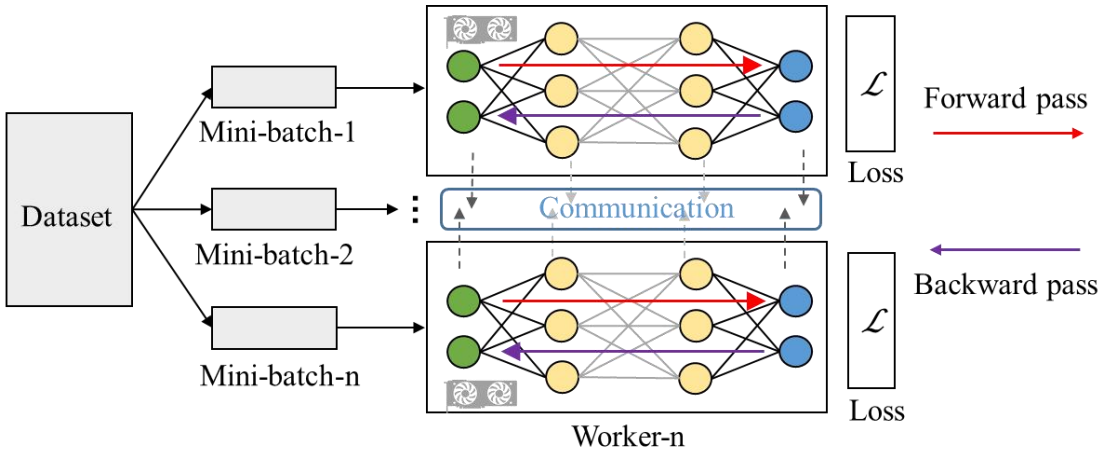


(a) Traditional DNN training

$$g_t = \nabla \mathcal{L}_{\xi_t \sim \mathcal{D}}(x_t, \xi_t)$$

$$x_{t+1} = x_t - \eta_t \cdot g_t$$

- Iterative algorithm to update model parameters x_t;
- Stochastic gradient g_t at t-th iteration;
- Learning rate η_t at t-th iteration.



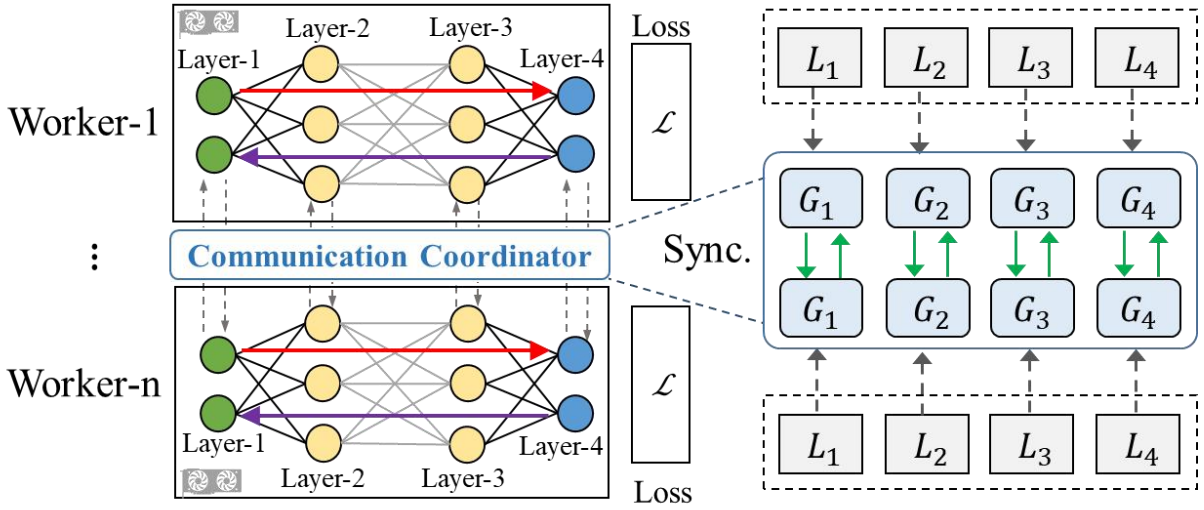
(b) Distributed DNN training

$$g_{i,t} = \nabla \mathcal{L}_{\xi_{i,t} \sim \mathcal{D}}\left(x_{i,t}, \xi_{i,t}\right)$$

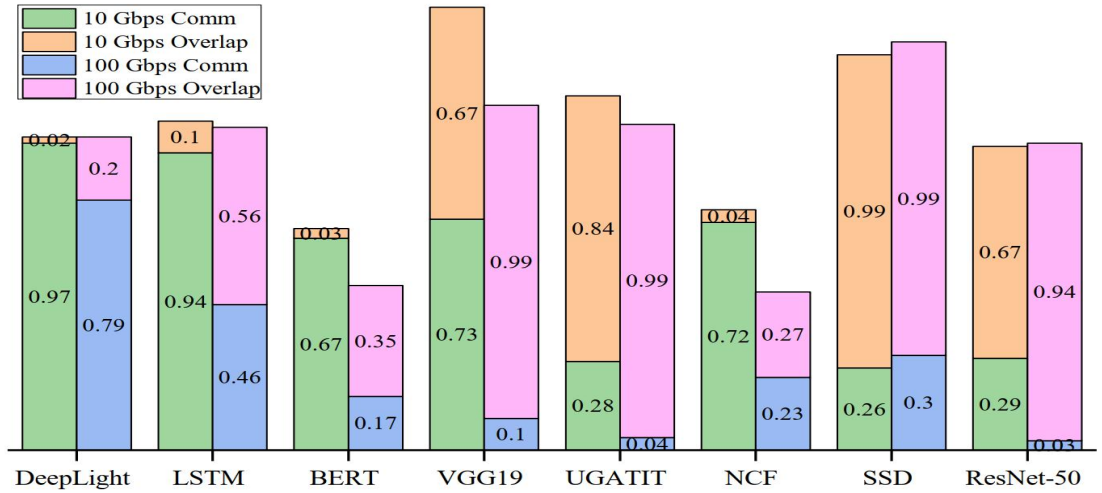
$$\overline{g}_{i,t} = \frac{1}{K}\sum_{i=1}^{K} g_{i,t} \ , \ \ x_{i,t+1} = x_{i,t} - \eta_{i,t} \cdot \overline{g}_{i,t}$$

- Iterative algorithm to update Worker-i's model parameters $x_{i,t}$;
- Stochastic gradient $g_{i,t}$ at t-th iteration and Worker-i;
- Learning rate $\eta_{i,t}$ at t-th iteration and Worker-i..

Performance Bottleneck



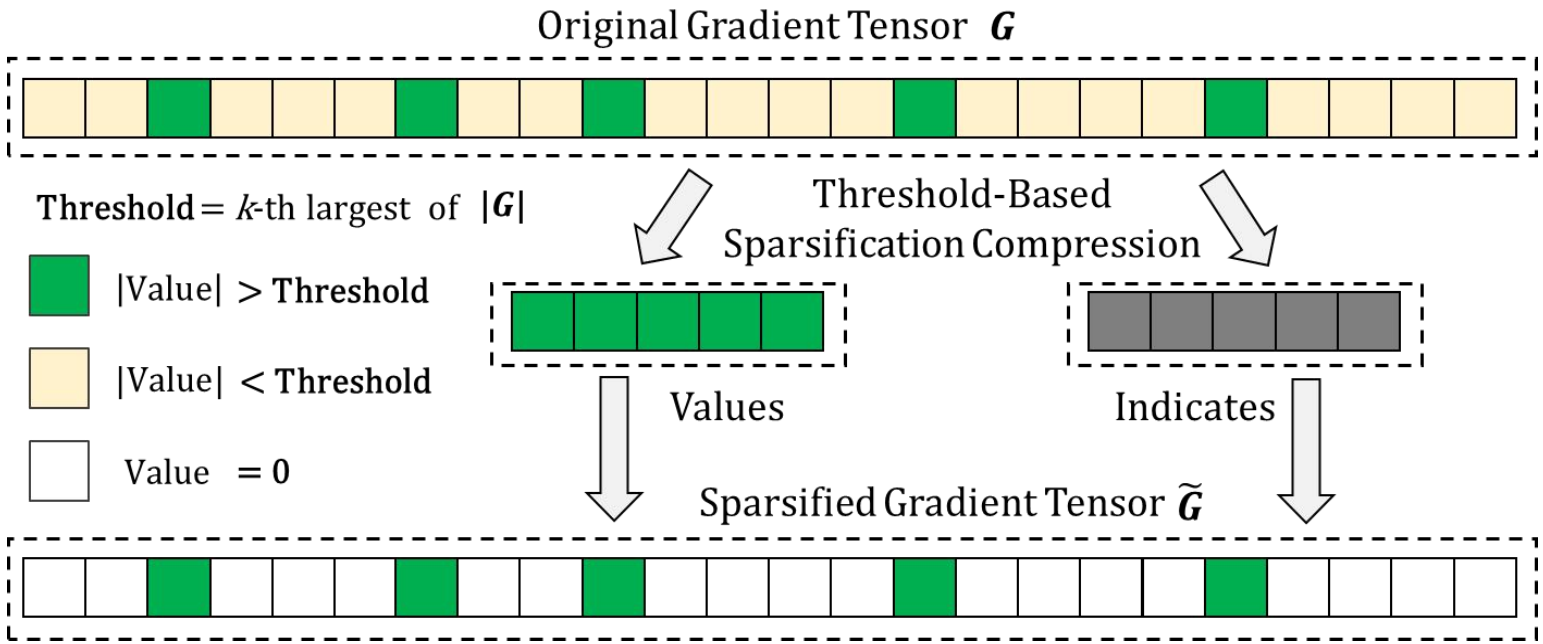
(a) Huge and frequent gradient synchronization



(b) Communication bottleneck

- **Huge Transmission Traffic:** DNN models contain millions to billions of parameters, leading to extremely high data transfer volumes during gradient synchronization across workers.
- **Frequent Communication Startups:** Traditional gradient synchronization incurs frequent communication startups, resulting in significant overhead and degraded training performance.

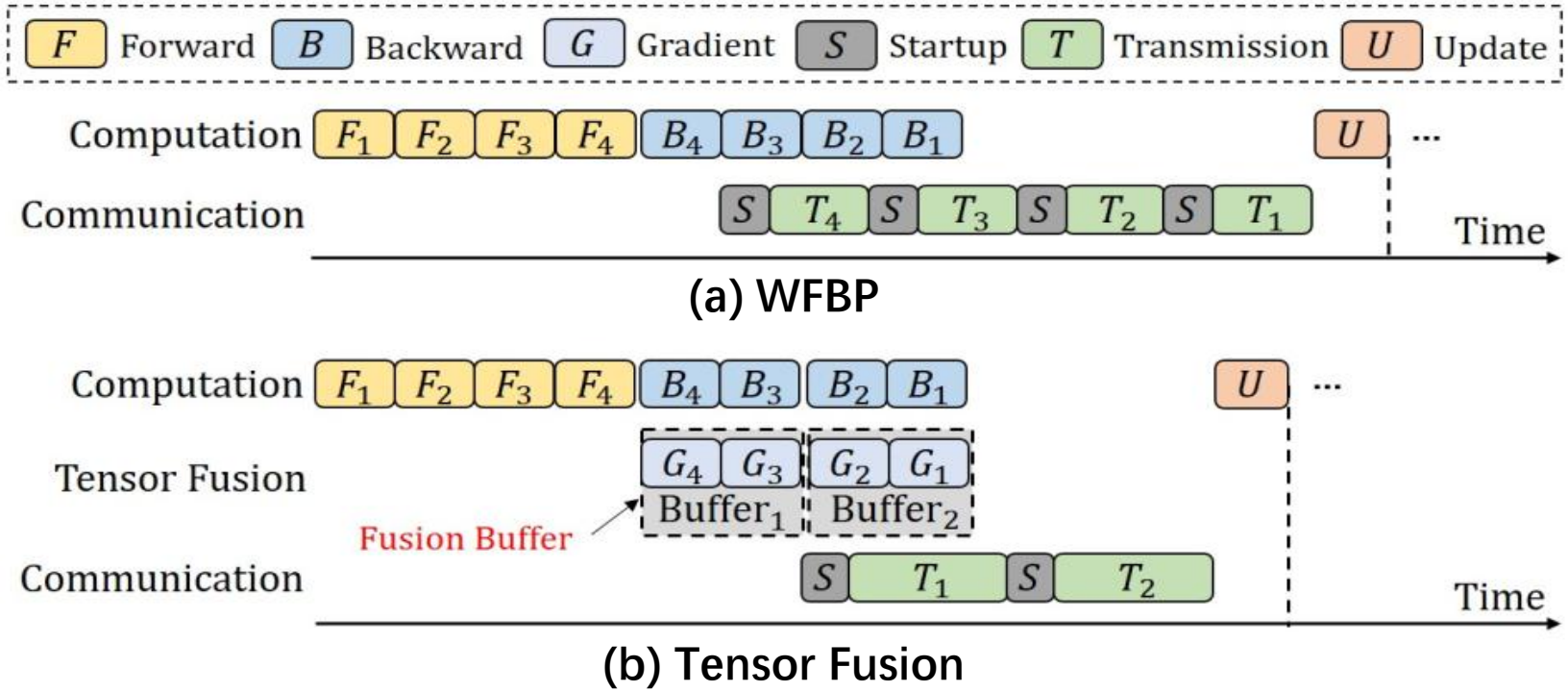
Sparsification Compression Technology



Gradient sparsification compression

- **Threshold Estimation:** Select only gradient elements whose absolute value is greater than the given threshold;
- **Gradient Element Selection:** Only the selected gradient values and its indices are transmitted for synchronous communication, and the other elements are set to 0.

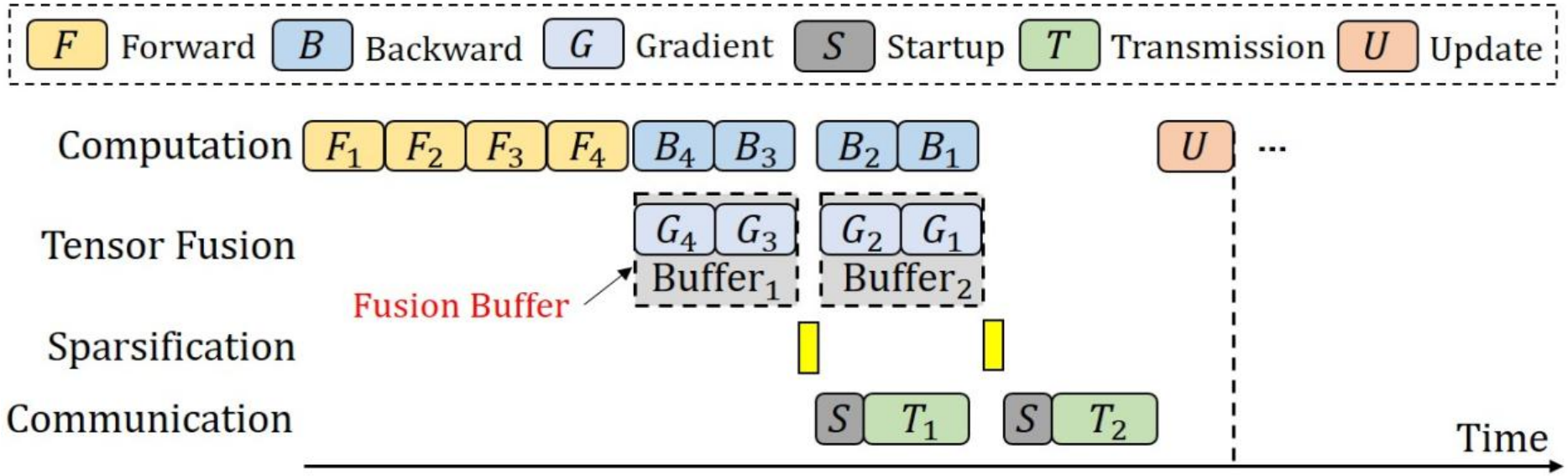
Tensor Fusion Technology



- **(a) Wait-Free Backpropagation (WFBP):** WFBP performs all-reduce operation independently for each gradient tensor, which causes a large communication startup overhead;

- **(b) Tensor Fusion:** Multiple gradient tensors are merged into a single fusion buffer, which performs only one all-reduce operation together.

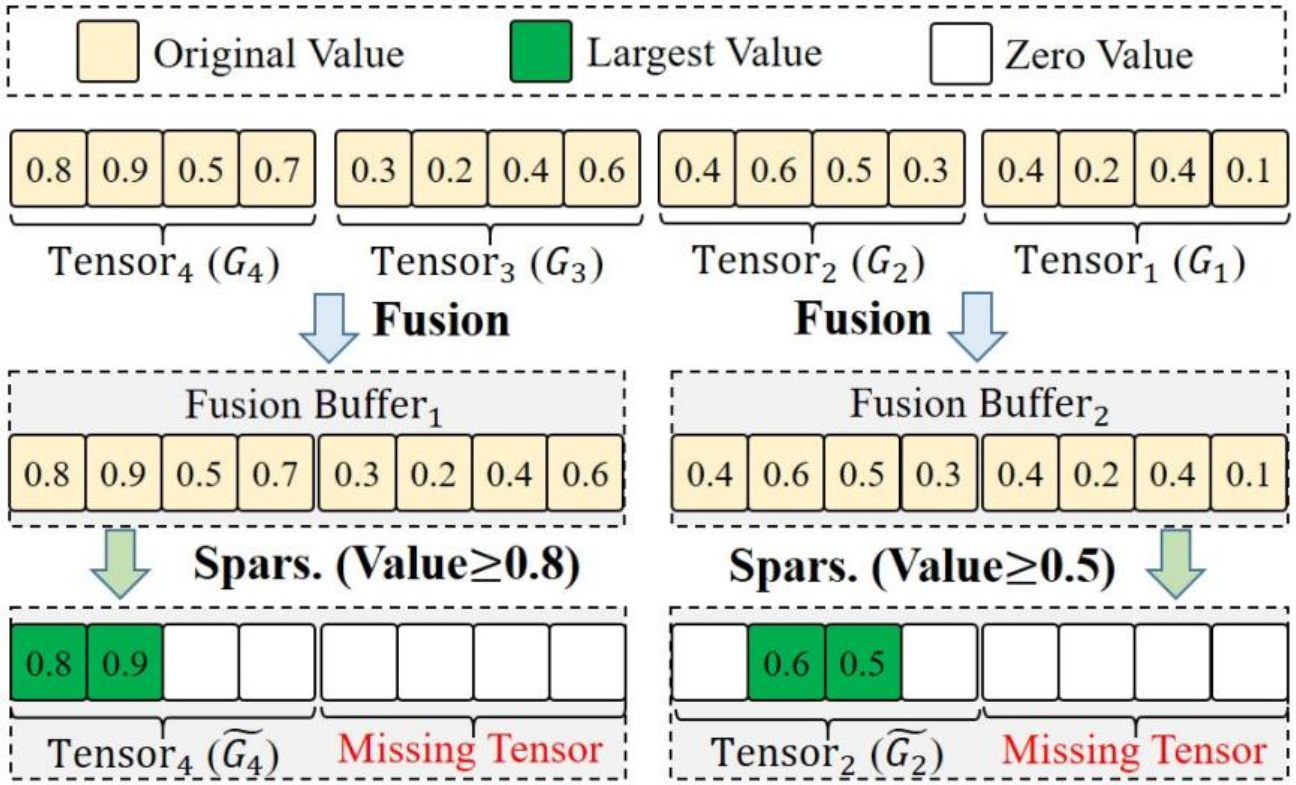
Combining Tensor Fusion with Sparsification



Example of combing tensor fusion with sparsification compression

- **Tensor Fusion with Sparsification:** To further improve the communication efficiency, recent state-of-the-art focus on combining tensor fusion and sparsification compression, which first performs tensor fusion that merges multiple gradient tensors and then performs sparsification compression.

Tensor Fusion with Sparsification Behind



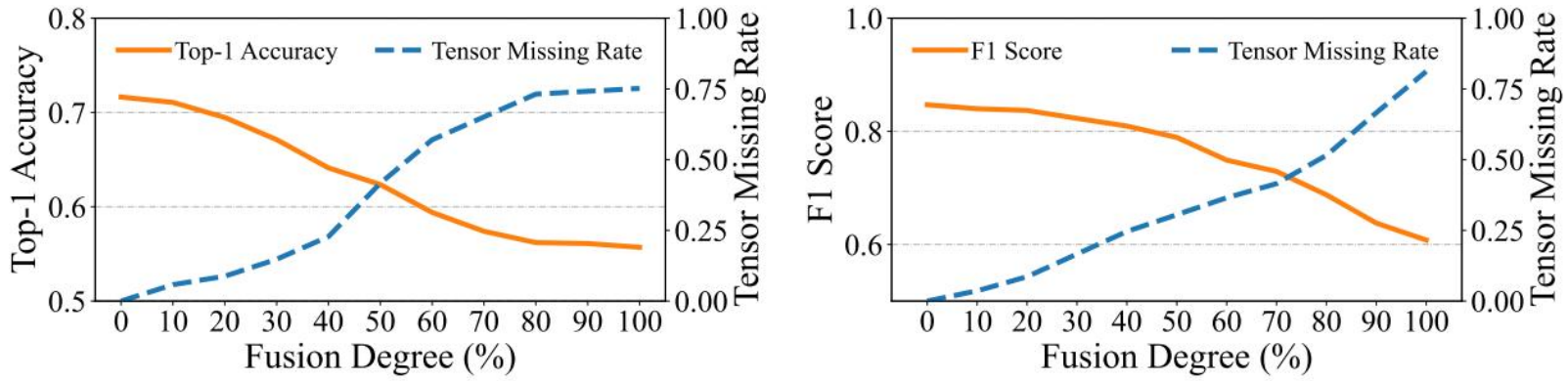
Traditional sparsification-behind tensor fusion (density = 25%)

- Existing state-of-the-art tensor fusion schemes typically compress compresses all the fused gradient tensors in the fusion buffer, which we call **sparsification-behind tensor fusion**.

- **The challenge of sparsification-behind tensor fusion: Gradient tensors are often missing** atop sparsification-behind tensor fusion, which can cause degradation in convergence performance.

Challenge and Motivation

Observation #1 (Challenge): Tensor Missing Leads to Low Convergence Accuracy

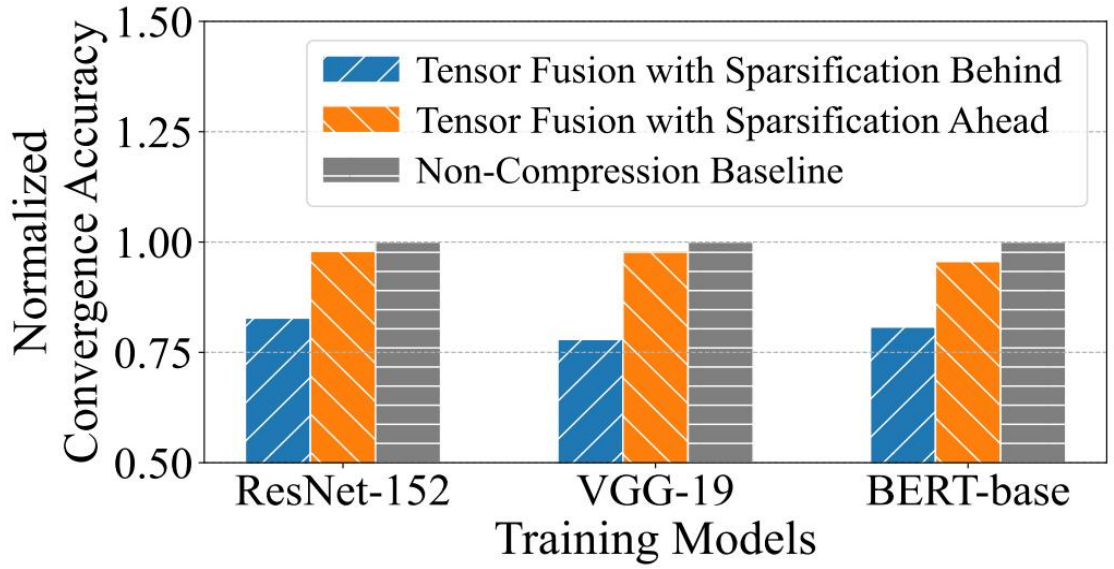


(a) ResNet-152 [50] on Cifar-100 [52] **(b) BERT-base [2] on SQuAD [53]**

The tensor missing rate and convergence accuracy for two training tasks with different fusion degree

- **Challenge:** Tensor missing after compression will lead to a degradation of the convergence performance, as confirmed in existing studies.

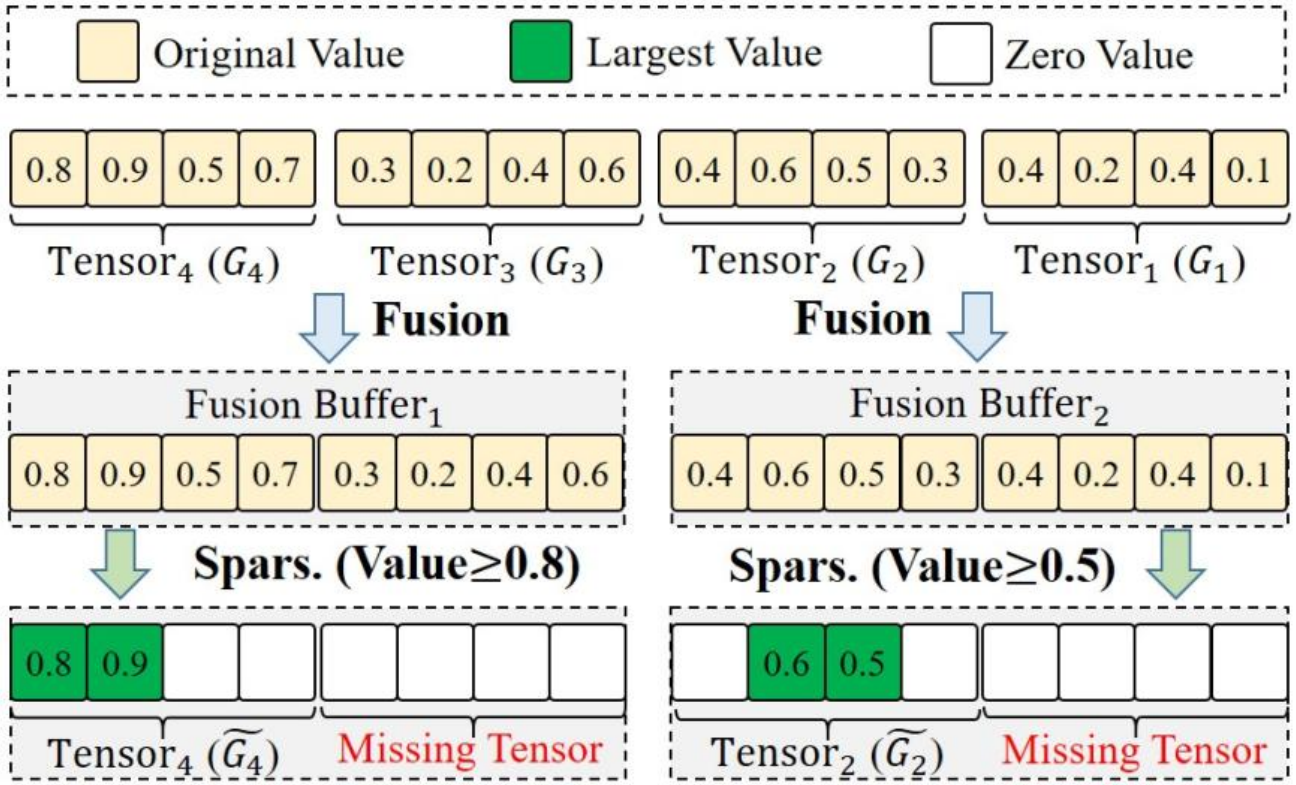
Observation #2 (Motivation): Tensor Fusion with Sparsification Ahead Avoids Tensor Missing



Comparison between sparsification-behind (traditional) tensor fusion and sparsification-ahead tensor fusion

- **Motivation:** For ResNet-152, VGG-19, and BERT-base, fusion with sparsification ahead improves convergence accuracy by 19.4%, 26.8%, and 17.6% over sparsification behind, and achieves accuracy close to the non-compression baseline.

Analysis



Sparsification-behind tensor fusion causes tensor missing

- **Sparsification-behind tensor fusion:**
  Selected gradient elements tend to be concentrated in the gradient tensors with larger magnitudes.

- **Sparsification-ahead tensor fusion:**
  Each tensor is independently sparsified before fused, ensuring elements from both large- and small-magnitude tensors are retained.

Our Insights and Main idea

➢ Insights:

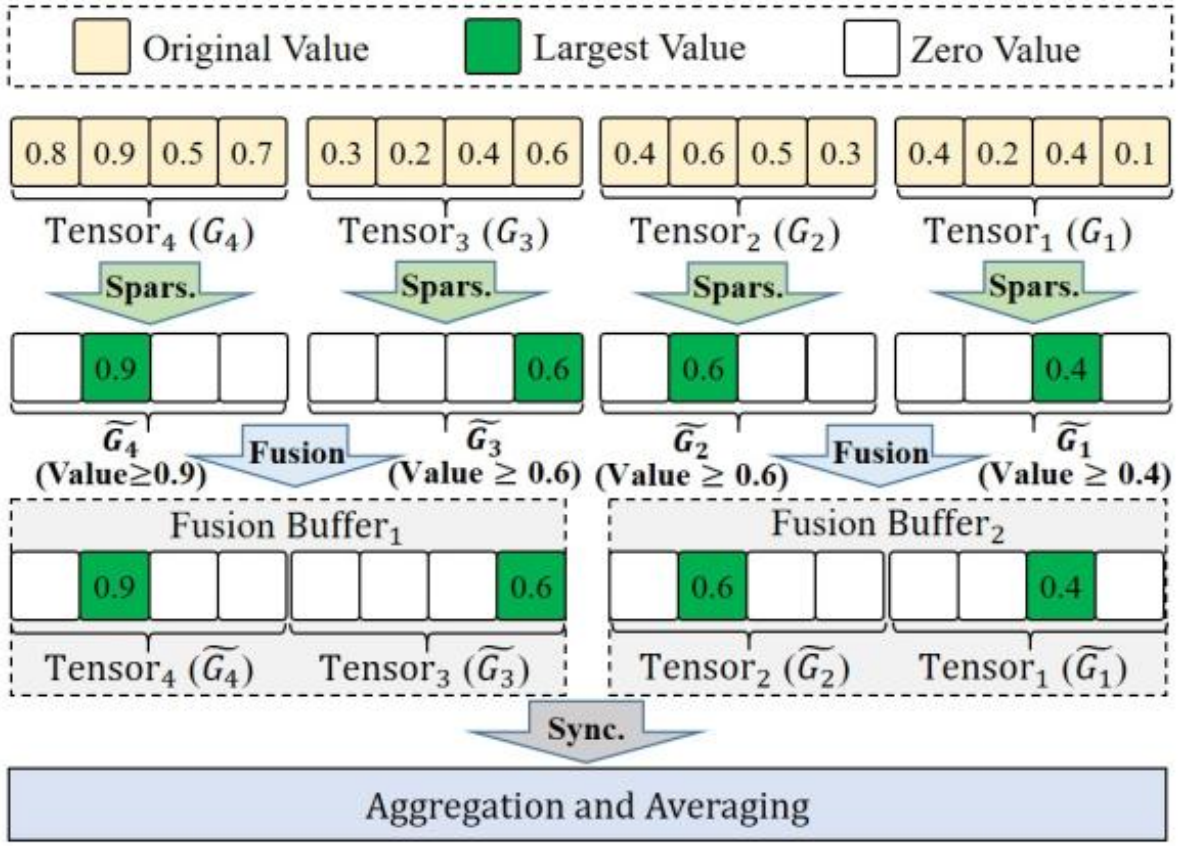
- Sparsification-behind tensor fusion performs sparsification on entire fused buffer **after** tensor fusion, which causes tensor missing;
- **Tensor missing** causes convergence performance degradation;

➢ Main idea:

- Performing sparsification on each gradient **before** tensor fusion can avoid tensor missing and achieve the higher convergence accuracy;

Design

Design SAFusion: Tensor Fusion with Sparsification Ahead

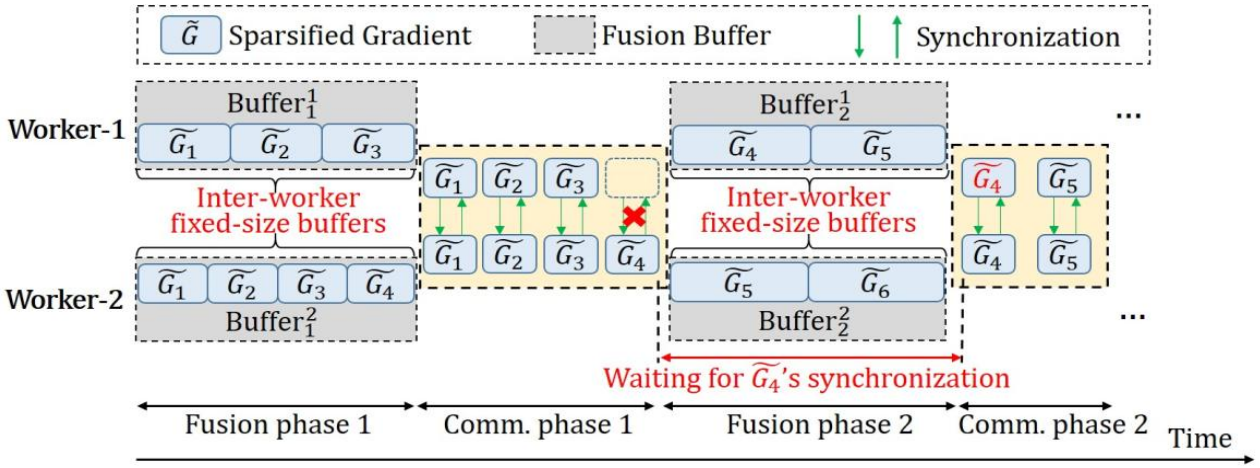


The process of SAFusion (density = 25%)

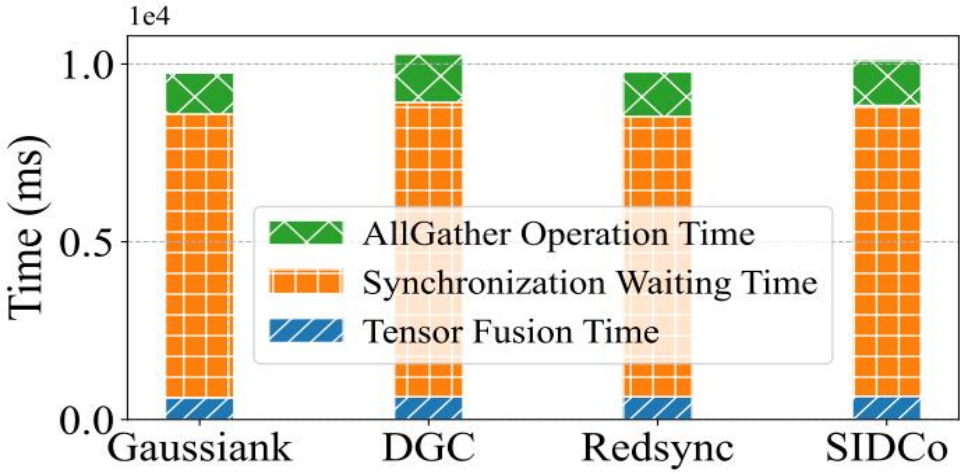
- **Design goals:** Avoiding tensor missing and improving convergence performance.

- **Design details:**
  - **Step-1:** Perform the threshold-based gradient sparsification compression;
  - **Step-2:** Merge sparsified gradients into the fixed-size fusion buffer;
  - **Step-3:** Pull fused gradients for synchronization.

Limitation #1 of SAFusion: Inter-Worker Long Synchronization Waiting



SAFusion may have long synchronization waiting

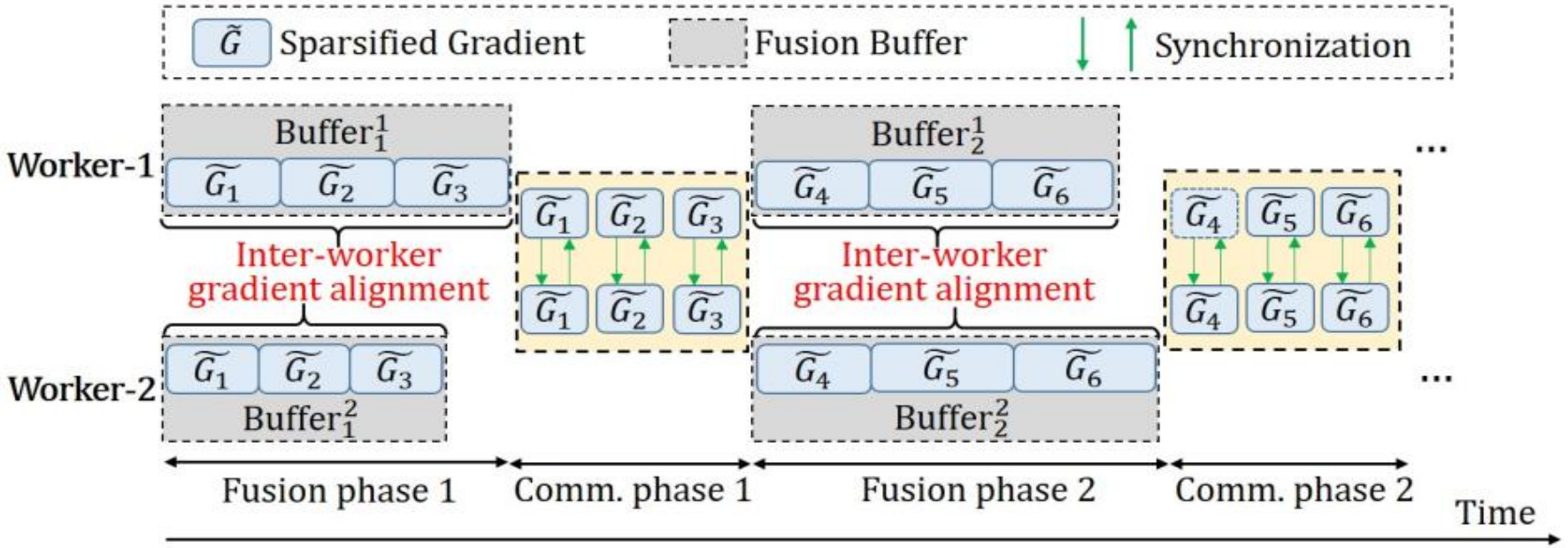


The breakdown of the training time in SAFusion with fixed buffer size

➢ Limitation #1 of SAFusion:

- **Findings**: 1) Fixed-size inter-worker buffer; 2) Varied-size sparsified gradients of different workers;
- Different workers have different numbers of sparsified gradients at the same fusion phase, thus some gradients may have to wait a long time for synchronization.

Design SAFusion-Inter: Inter-Worker Gradient Alignment Fusion

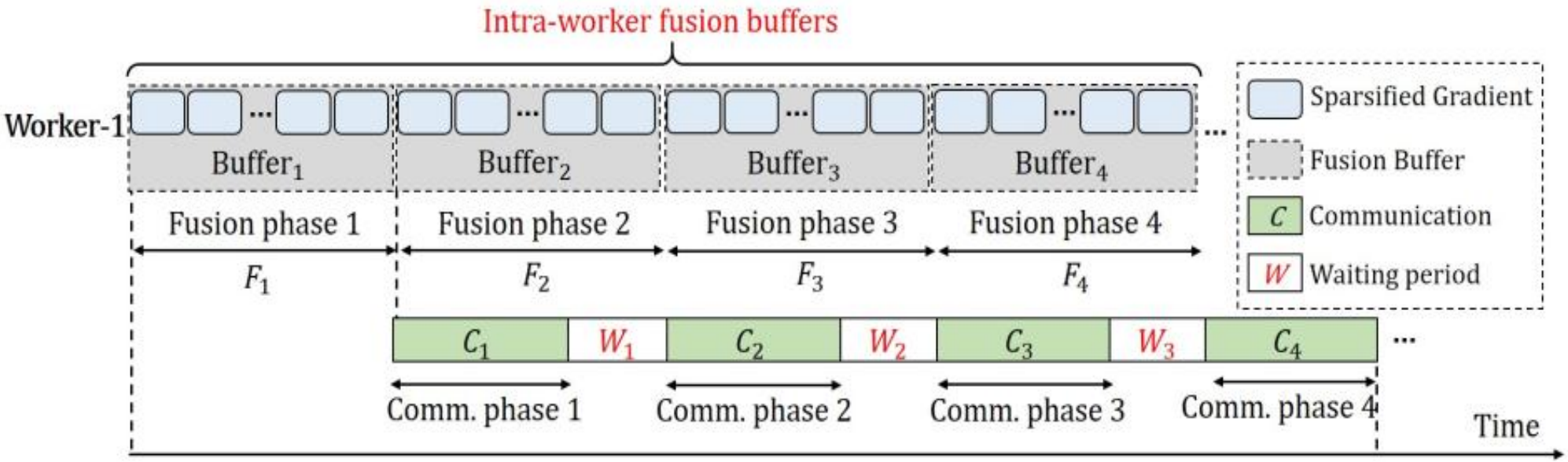


Example of aligned inter-worker tensor fusion

➢ Design idea:

- **Align** the same number of variable-size sparsified gradient tensors of the inter-worker buffers during the same fusion phase, instead of a fixed fusion buffer size.

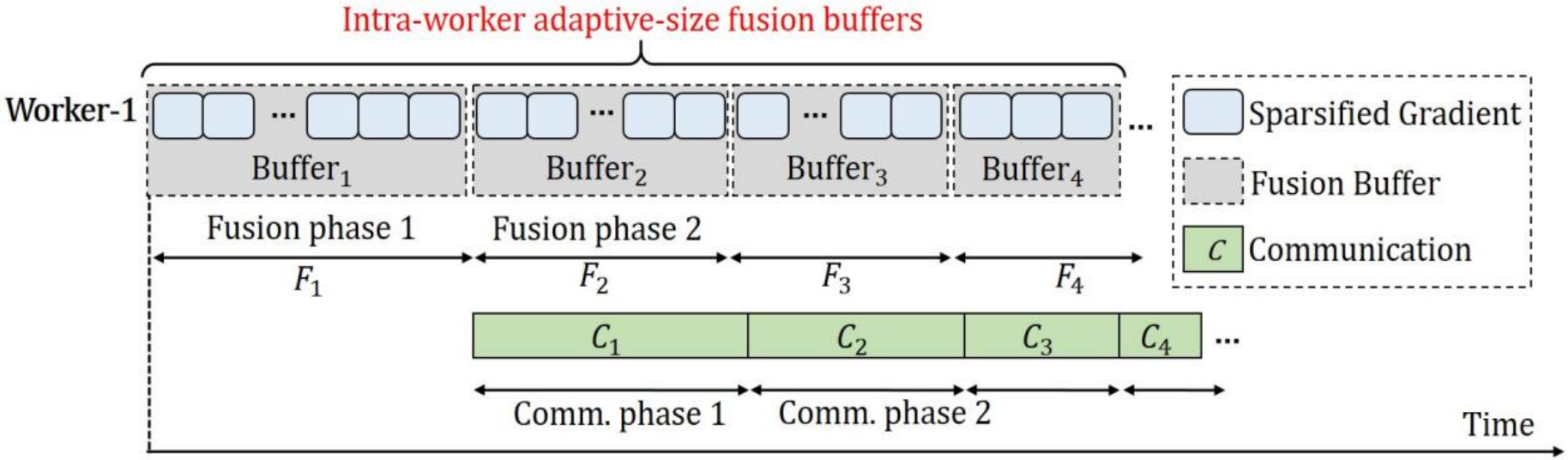
Limitation #2 of SAFusion: Multiple Intra-Worker Waiting Periods



A general case of intra-worker communication pipeline for SAFusion

- **Limitation #2 of SAFusion:**
  - Sparse tensor fusion computation time greater than communication time;
  - Multiple intra-worker waiting periods within each iteration;

Design SAFusion-Intra: Intra-Worker Adaptive Fusion



Example of intra-worker tensor fusion with adaptive buffer size

- **Design idea:**
  - **Dynamically adjust** the number of fused tensors in the previous fusion buffer until its communication time begins to only slightly greater than or equal to the sparse tensor fusion time in the current fusion buffer;

Implementation

➢ Generator Module

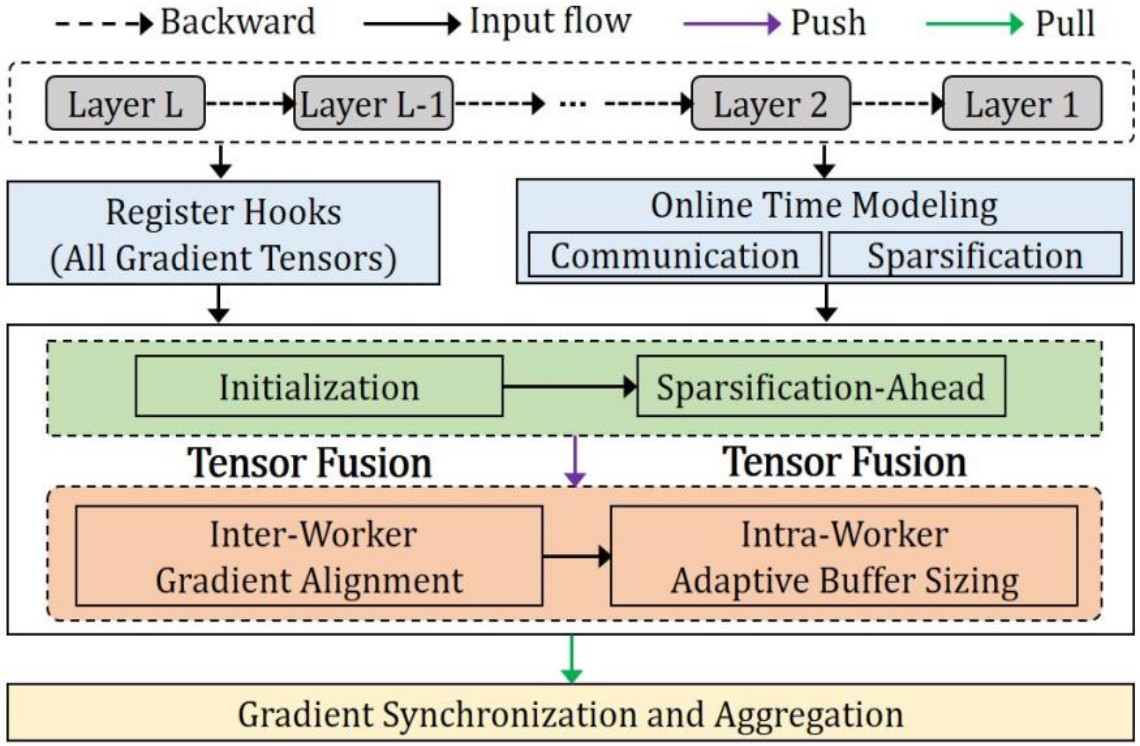
- Initialization and sparsification-ahead;
- Inter-worker gradient alignment tensor fusion;
- Intra-worker adaptive tensor fusion;

➢ Controller Module

- Control sparsified gradient pushing and pulling;
- Perform all-gather operation on the fusion buffer;

➢ Sparsification Compression Module

- Implement state-of-the-art gradient sparsification compression libraries, including DGC, Gaussiank, Redsync, and SIDCo;



The workflow of SAFusion generator

Evaluation and Conclusion

Experiment Setup

➢ Testbeds

- **Cloud GPU cluster:** 16 instances with 64 NVIDIA A100 GPUs connected by a 200Gbps InfiniBand;
- **Local GPU cluster:** 8 nodes with 8 NVIDIA V100 GPUs connected by a 100Gbps InfiniBand;

➢ Workloads

- Image classification: ResNet-152, VGG-19, and ViT-large on Cifar-100 and ImageNet;
- Nature language processing: LSTM and GPT2-large on WikiText-2/103, BERT-base/large on SQuAD;

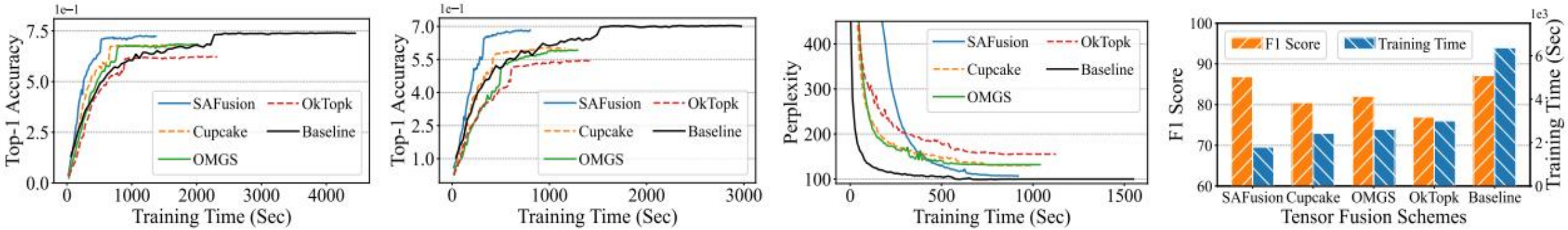
➢ Baseline and comparative approaches

- No-fusion Baseline, OkTopk, OMGS, Cupcake;
- SAFusion, SAFusion-Inter, and SAFusion-(Inter+Intra);

➢ Performance metric

- Training Throughput, Convergence performance;

Experiment-1: Convergence Performance

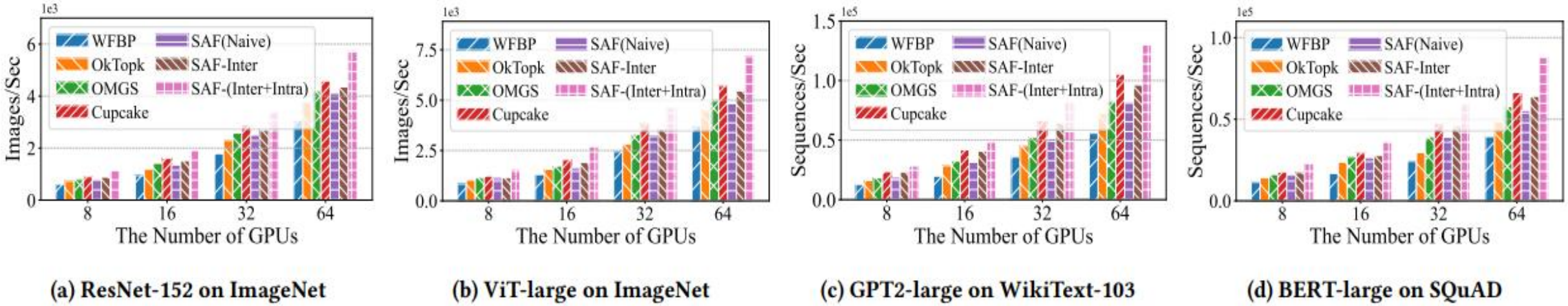


(a) ResNet-152 on Cifar-100 (b) VGG-19 on Cifar-100 (c) LSTM on WikiText-2 (d) BERT-large on SQuAD

Experiment 1: Training time and convergence accuracy of DNN training tasks

- **Image classification tasks:**
  - For ResNet-152 and VGG19 on Cifar-100, the Top-1 accuracy of SAFusion are slightly lower than that of Baseline, but larger than other state-of-the-art tensor fusion methods.
- **Natural language processing tasks:**
  - For LSTM and BERT-large, SAFusion also maintains the same convergence performance as the non-compression Baseline.

Experiment-2: Training Throughput

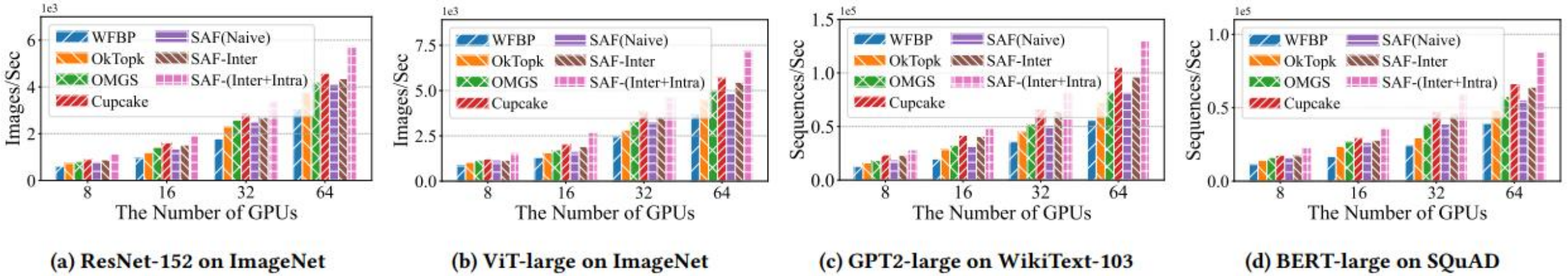


(a) ResNet-152 on ImageNet (b) ViT-large on ImageNet (c) GPT2-large on WikiText-103 (d) BERT-large on SQuAD

Experiment 2: Throughput of DNN training tasks on a cloud cluster

- **Image classification tasks:**
  - For ResNet-152 and ViT-large on ImageNet, SAFusion's training throughput of outperforms other methods by 18%-104%, versus different GPU cluster sizes.
- **Natural language processing tasks:**
  - For GPT2-large and BERT-large, SAFusion improves training throughput by 20%-144% versus different GPU cluster sizes.

Experiment-3: Effectiveness of Various Optimization

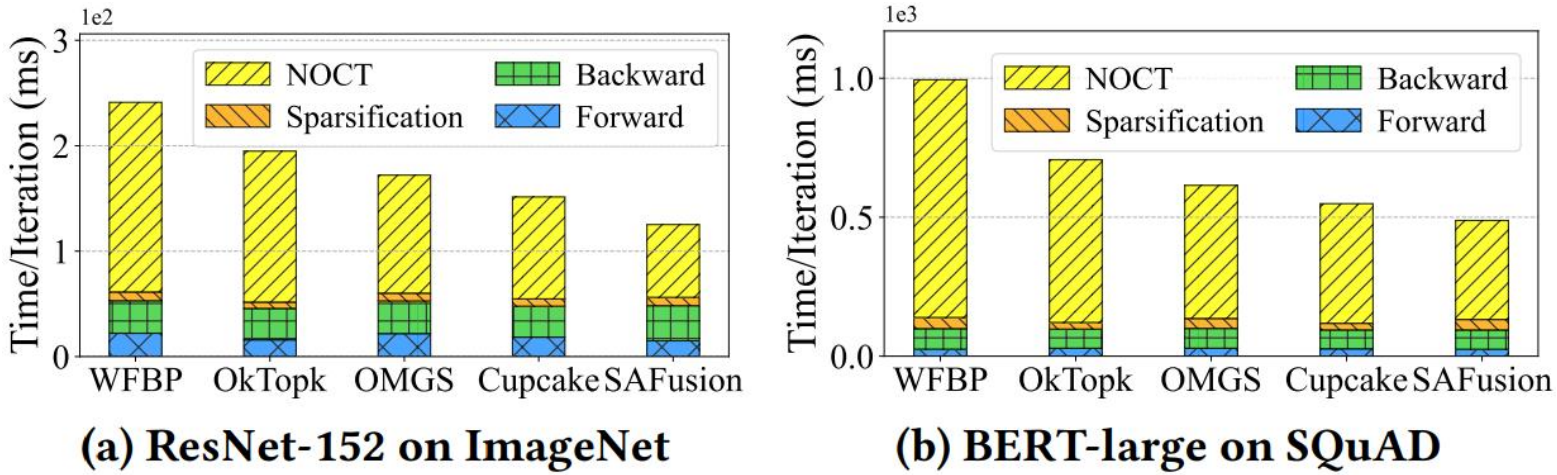


Experiment 3: Throughput of DNN training tasks

➢ Ablation experiment:

- With the inter-worker gradient alignment scheme (SAF-Inter), the training throughput improves by up to 12.4%, 18.2%, 30.8%, and 16.8% over SAF (Naive) on the four training tasks.
- Compared to SAF-Inter, the intra-worker adaptive fusion scheme (SAF-Inter+Intra) improves training throughput by up to 34.7%, 46.4%, 35.3%, and 37.7% across four tasks, respectively.

Experiment-4: Training Time Breakdown

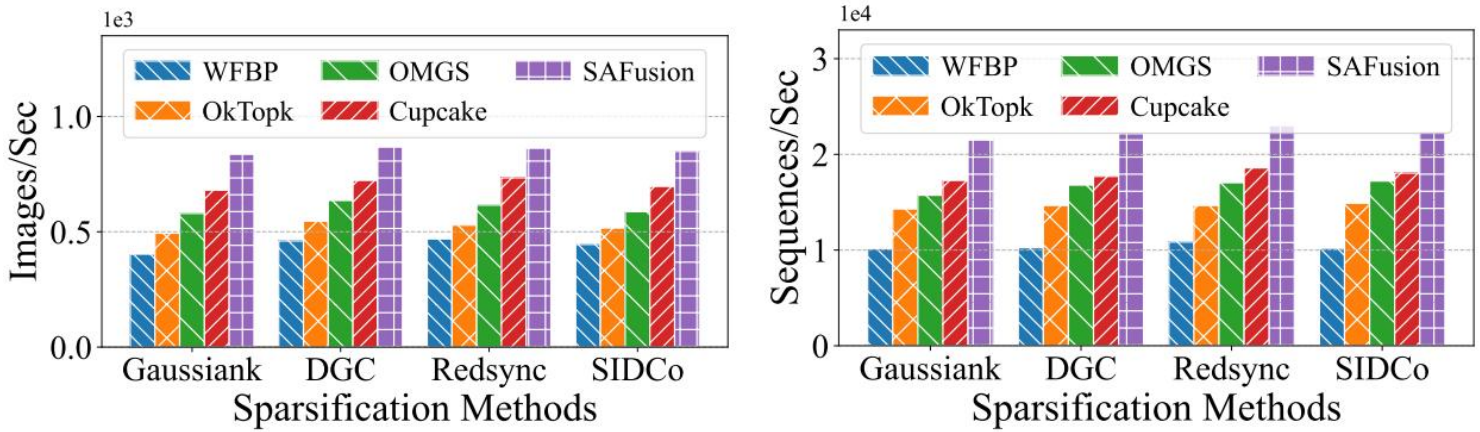


Experiment 4: Training time breakdown for different tensor fusion schemes on two training tasks

➢ Training time breakdown:

- For the two training tasks, SAFusion reduces non-overlapping communication time (NOCT) by 132%-160% compared to WFBP, and by 20%-107% compared to other tensor fusion methods.
- SAFusion's compression time increases by only 4.5%-6.2% compared to OMGS and Cupcake.

Experiment-5: Impact of Different Sparsification Methods

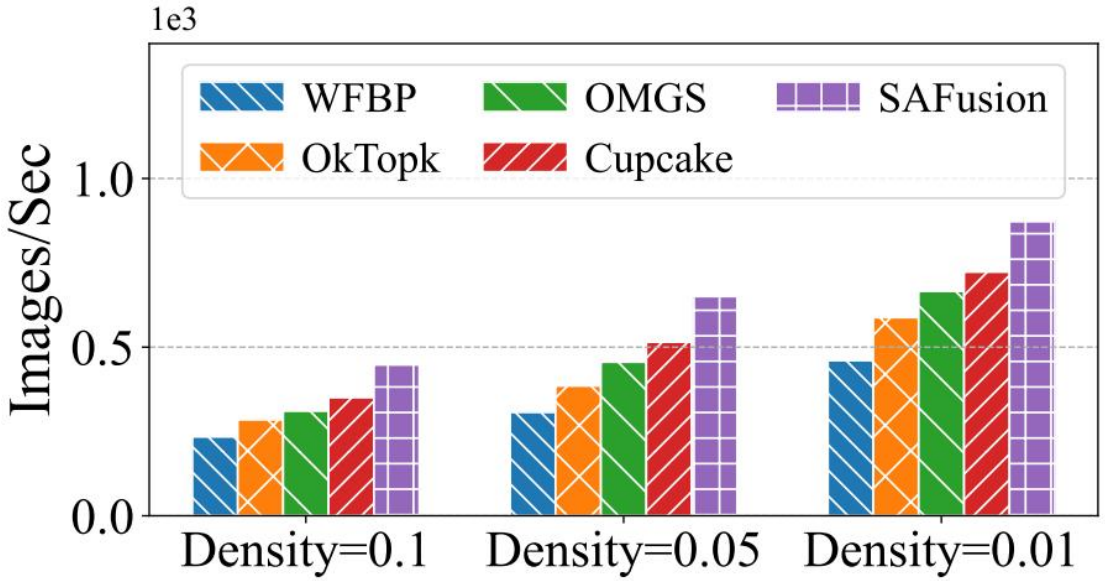


(a) ResNet-152 on ImageNet (b) BERT-large on SQuAD

Experiment 5: Training performance comparison using different sparsification compression methods

- **Different sparsification methods:**
  - For ResNet-152 on ImageNet, compared to WFBP, OkTopk, OMGS, and Cupcake, SAFusion achieves throughput improvements of 14%-77%;
  - For BERT-large on SQuAD, SAFusion also increases the training throughput by 10.2%-90.6%.

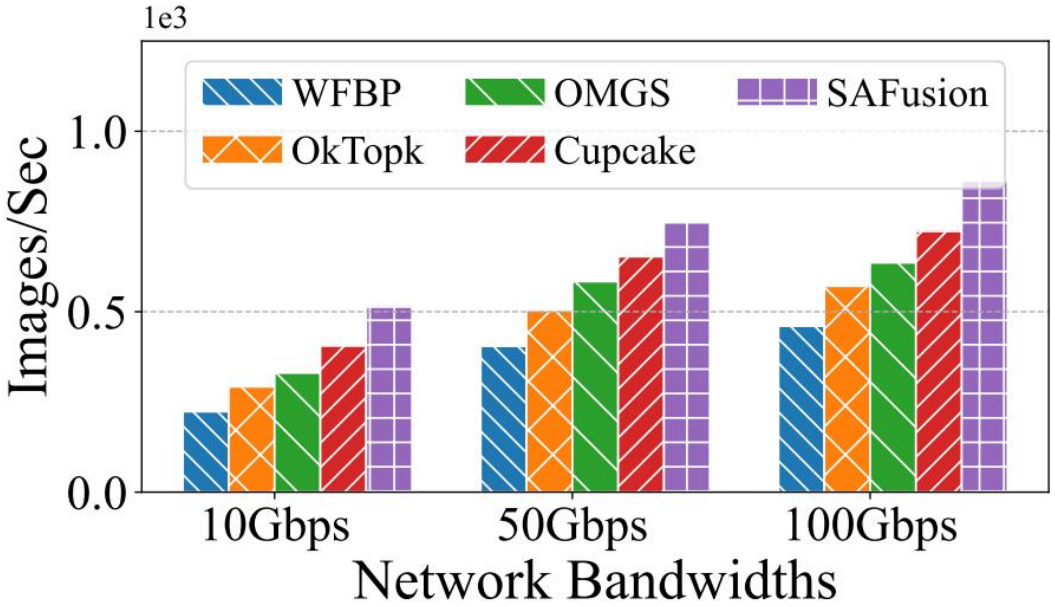
Experiment-6: Impact of Sparsification Densities



Experiment 6: Training performance comparison using different sparsification density

- **Different sparsification densities:**
  - SAFusion's training throughput outperforms other tensor fusion schemes by 27.7%-78.6%, 26.2%-83.1%, and 14.2%-77.2% when the density is reduced from 0.1 to 0.05 and 0.01, respectively.

Experiment-7: Impact of Network Bandwidths



Experiment 7: Training performance comparison using different bandwidths

- **Different network bandwidths:**
  - Under the three network environments, SAFusion's training throughput outperforms other state-of-the-art tensor fusion schemes by 12.1%-85.9%, 9.2%-75.3%, and 11.2%-77.2%, respectively.

Conclusion

- How to optimize tensor fusion with sparsification in high-performance distributed DNN training systems?
- Contributions
  - **SAFusion: avoids gradient tensor missing** via a sparsification-ahead tensor fusion mechanism to improve the convergence performance;
  - **SAFusion-Inter: aligns the number of sparsified gradients among fusion buffers** of different workers to reduce the long inter-worker gradient synchronization waiting;
  - **SAFusion-Intra: dynamically adjusts the number of gradients merged in the fusion buffer** for each worker to improve the intra-worker training pipeline efficiency.
- Prototype
  - **https://github.com/YuchongHu/SAFusion**
- Contacts
  - zqming@hust.edu.cn and yuchonghu@hust.edu.cn

Thanks!

Zhangqiang Ming: zqming@hust.edu.cn

華中科技大學
HUAZHONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

CETC
中电海康集团有限公司