北京大学
PEKING UNIVERSITY

Lower Bounds and Accelerated Algorithms in Distributed Optimization with Communication Compression

Kun Yuan (袁 坤)

Center for Machine Learning Research @ Peking University

May 15, 2023

Joint work with

Yutong He
(Peking U.)

Xinmeng Huang
(UPenn)

Yiming Chen
(Alibaba)

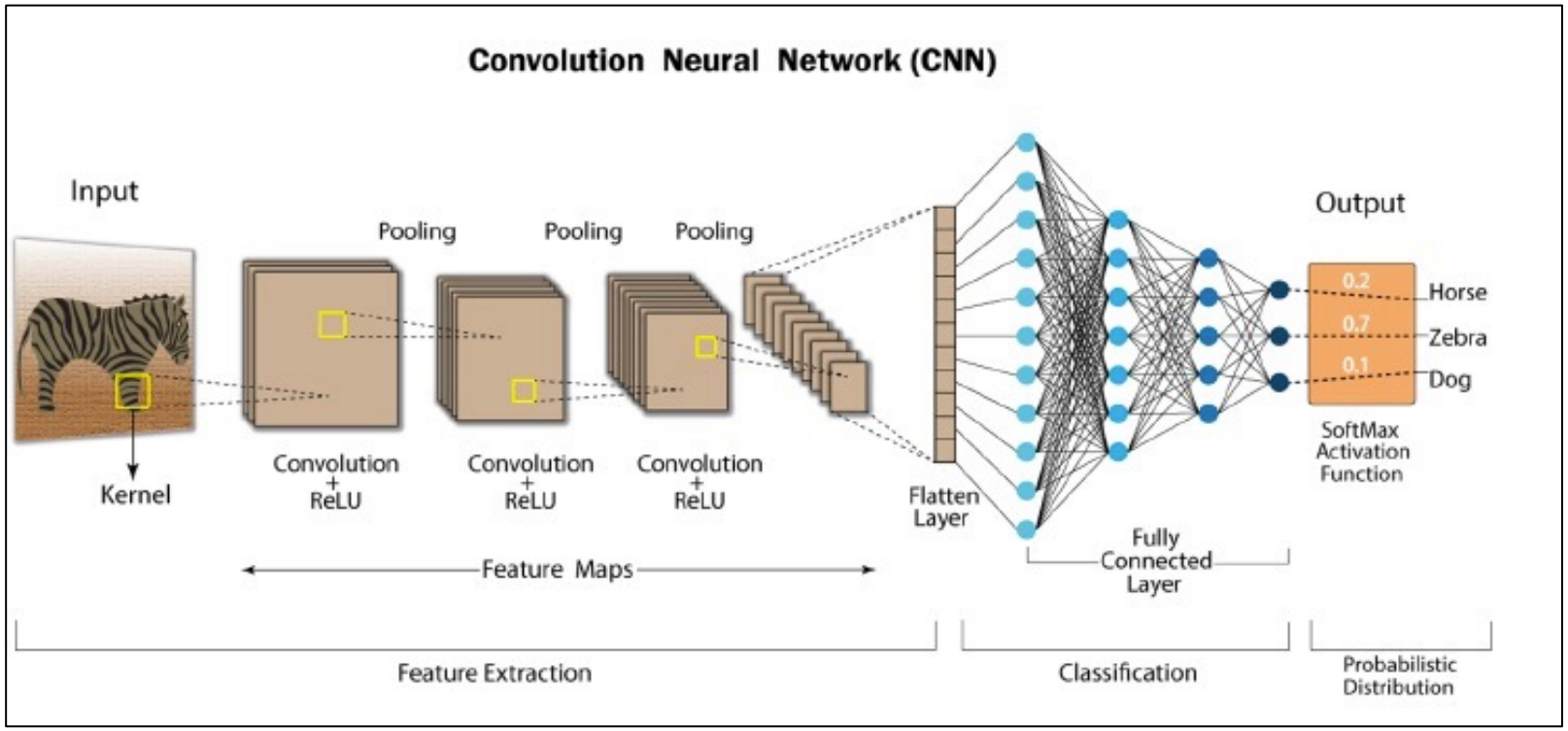
Wotao Yin
(Alibaba)

北京大学
PEKING UNIVERSITY

PART 01

Basics and Motivation

Training deep neural network is notoriously difficult



DNN training = non-convexity + **massive dataset** + huge models

Distributed learning

北京大学 PEKING UNIVERSITY

- Training deep neural networks typically requires **massive** datasets; efficient and scalable distributed optimization algorithms are in urgent need

- A network of n nodes (devices such as GPUs) collaborate to solve the problem:

$$\min_{x\in\mathbb{R}^d} \quad f(x)=\frac{1}{n}\sum_{i=1}^{n} f_i(x), \quad \text{where} \quad f_i(x)=\mathbb{E}_{\xi_i\sim D_i} F(x;\xi_i)$$

  - Each component $f_i : \mathbb{R}^d \to \mathbb{R}$ is local and private to node i
  - Random variable ξ_i denotes the local data that follows distribution D_i
  - Each local distribution D_i is different; data heterogeneity exists

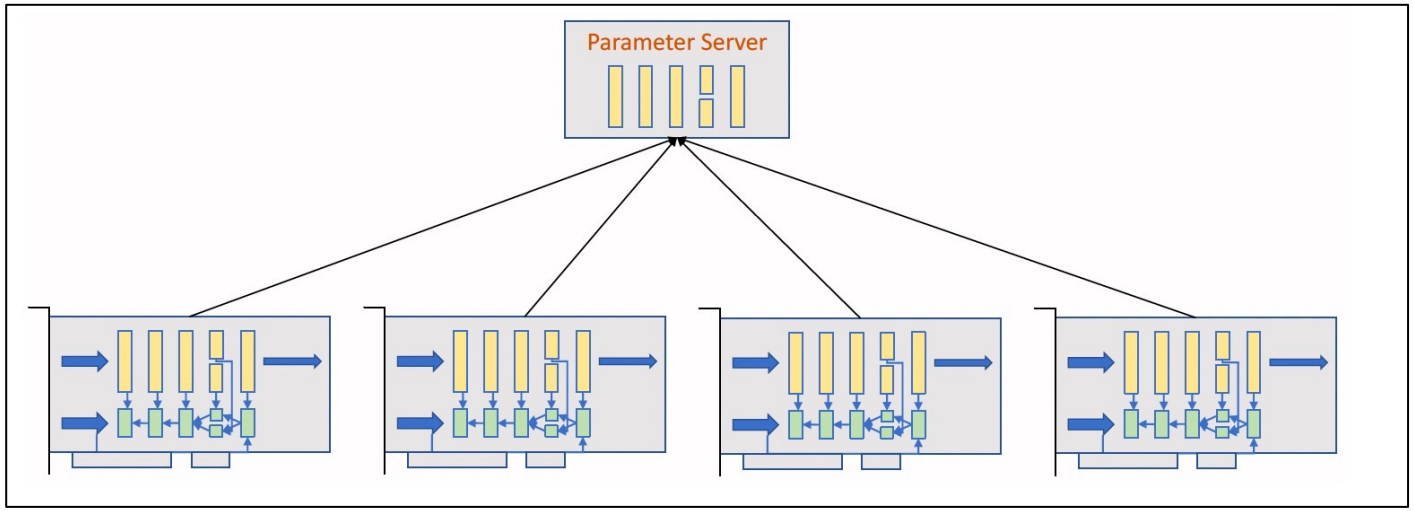
Vanilla parallel stochastic gradient descent (PSGD)

$$g_i^{(k)} = \nabla F(x^{(k)}; \xi_i^{(k)}) \qquad \text{(Local compt.)}$$

$$x^{(k+1)} = x^{(k)} - \frac{\gamma}{n} \sum_{i=1}^{n} g_i^{(k)} \qquad \text{(Global comm.)}$$

- Each node i samples data $\xi_i^{(k)}$ and computes gradient $\nabla F(x^{(k)}; \xi_i^{(k)})$
- All nodes synchronize (i.e. globally average) to update model x per iteration

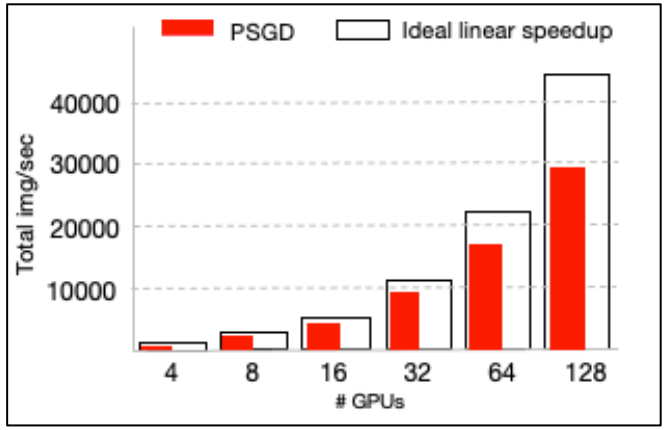
Vanilla parallel stochastic gradient descent (PSGD)



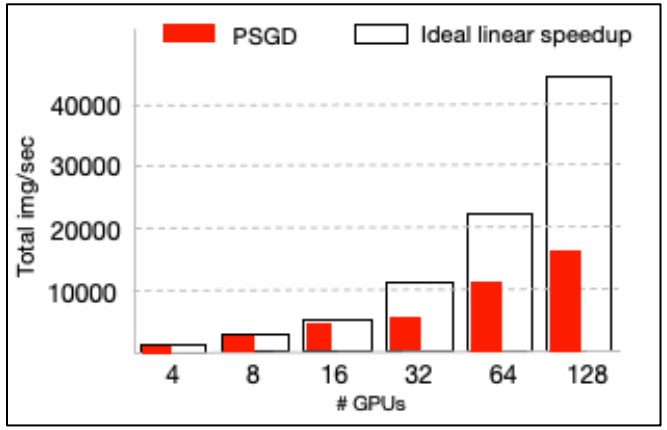
- Each node sends a full model (or gradient) to the server; proportional to dimension d
- When model dimension d is large, PSGD suffers severe communication overhead

PSGD cannot achieve linear speedup due to comm. overhead

- PSGD cannot achieve ideal linear speedup in throughput due to comm. overhead
- Larger comm-to-compt ratio leads to worse performance in PSGD



Small comm.-to-compt. ratio



Large comm.-to-compt. ratio

- How can we accelerate PSGD? **Distributed optimization with communication compression**

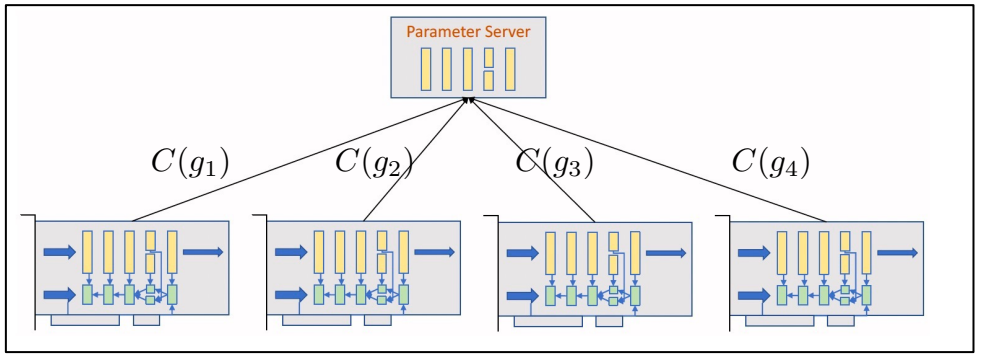
B. Ying, K. Yuan, H. Hu, Y. Chen and W. Yin, “BlueFog: Make decentralized algorithms practical for optimization and deep learning”, arXiv: 2111. 04287, 2021

Ccommunication compression

- A basic (but not state-of-the-art) algorithm is QSGD [Alistarh et. al., 2017]

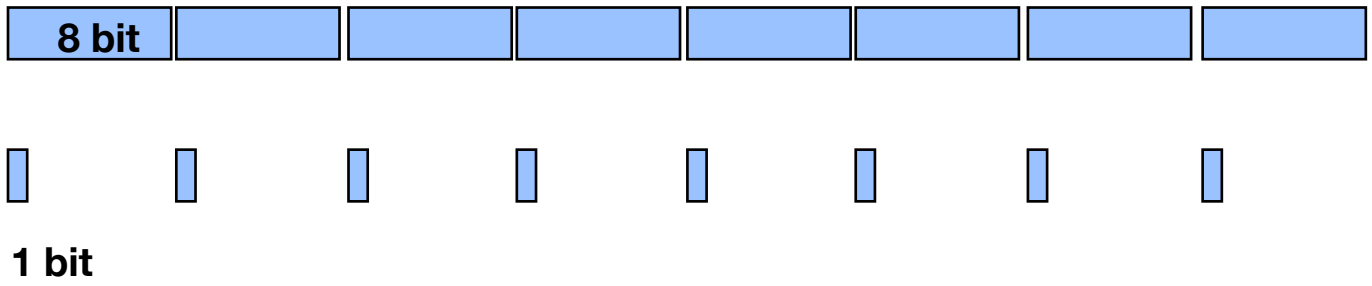
$$g_i^{(k)} = \nabla F(x_i^{(k)}; \xi_i^{(k)})$$

$$x_i^{(k+1)} = x_i^{(k)} - \frac{\gamma}{n} \sum_{j=1}^{n} C(g_j^{(k)})$$



- $C(\cdot)$ is a compressor. It can quantize or sparsify the full gradient

Quantization

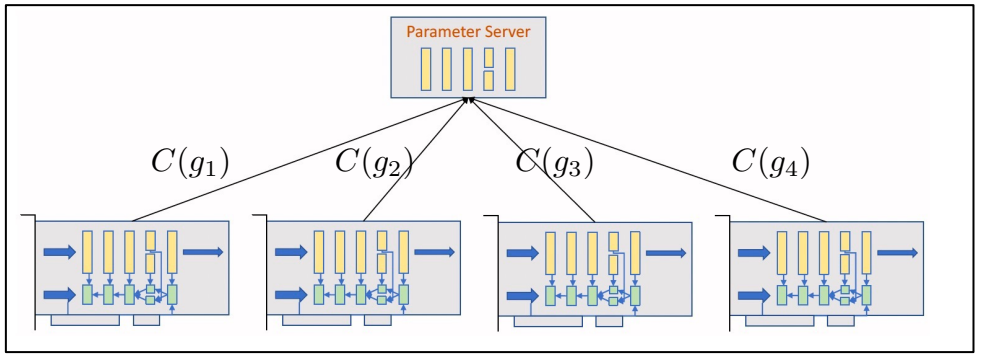


Ccommunication compression

- A basic (but not state-of-the-art) algorithm is QSGD [Alistarh et. al., 2017]

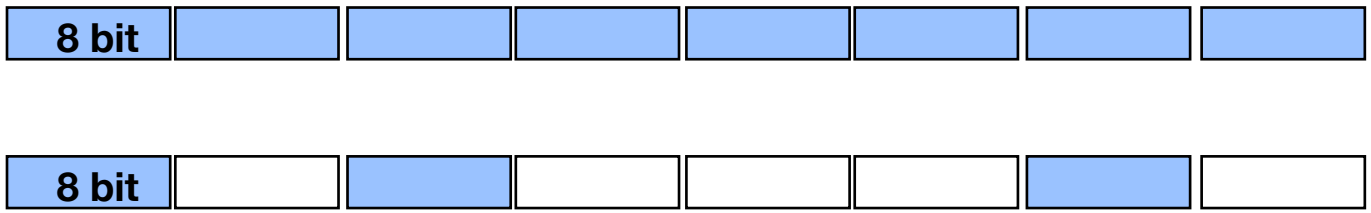
$$g_i^{(k)} = \nabla F(x_i^{(k)}; \xi_i^{(k)})$$

$$x_i^{(k+1)} = x_i^{(k)} - \frac{\gamma}{n} \sum_{j=1}^{n} C(g_j^{(k)})$$



- $C(\cdot)$ is a compressor. It can quantize or sparsify the full gradient

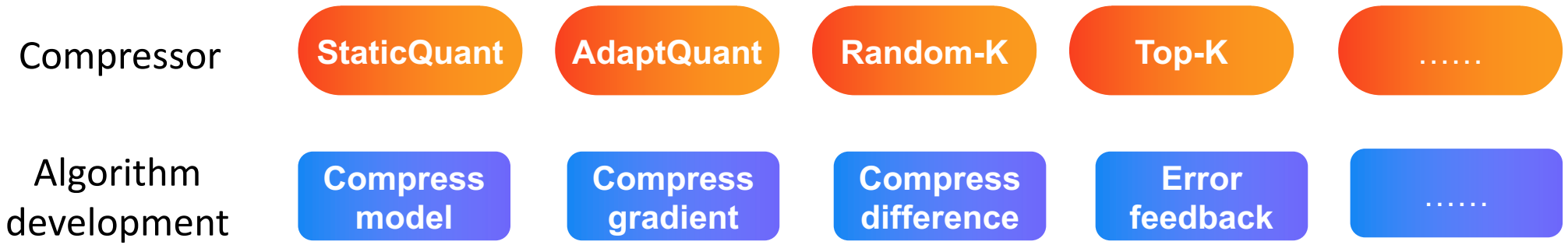
Sparsification



Ccommunication compression algorithms

北京大学 PEKING UNIVERSITY

- There are extensive studies in distributed learning with communication compression



- The combination of different compressors, algorithms, and strategies gives rise to

Q-SGD [Alistarh et. al., 2017], Mem-SGD [Stich et. al., 2018], EF21-SGD [Fatkhullin et. al., 2021], CSER [Xie et.al., 2020], Double Squeeze [Tang et. al., 2019], Artemis [Philippenko et.al. 2022], etc.

Fundamental open questions

北京大学 PEKING UNIVERSITY

- **What is the optimal complexity in distributed optimization with communication compression?**

- **Can we develop effective algorithms that can attain the optimal complexity?**

北京大学
PEKING UNIVERSITY

PART 02

Optimal Complexity Formulation

Function class and gradient oracle class

- Function class. We let $\mathcal{F}_{\mu,L}$ denote the set of all functions satisfying Assumption 1

Assumption 1 (Smoothness and Strong Convexity) Each local objective f_i has L-Lipschitz gradient and μ-strongly convex. Moreover, we assume that $f(x^{(0)}) - \inf_{x\in\mathbb{R}^d} f(x) \leq \Delta$ with $f = \frac{1}{n}\sum_{i=1}^n f_i$.

- Gradient oracle class. Each worker accesses local gradient $\nabla f_i(x)$ via a stochastic oracle

Assumption 2 (Stochastic gradient) The gradient oracles $\{O_i : 1 \leq i \leq n\}$ satisfy

$$\mathbb{E}_{\zeta_i}[O_i(x;\zeta_i)] = \nabla f_i(x) \quad \text{and} \quad \mathbb{E}_{\zeta_i}[\|O_i(x;\zeta_i) - \nabla f_i(x)\|^2] \leq \sigma^2, \quad \forall\, x \in \mathbb{R}^d.$$

Compressor Class

- Compressor class. Most compressors in literature are either **unbiased** or **contractive**

- We let $\mathcal{U}_\omega$ denote the set of unbiased compressors satisfying Assumption 3

Assump. 3 (Unbiased compressor) The compression operator $C : \mathbb{R}^d \to \mathbb{R}^d$ satisfies

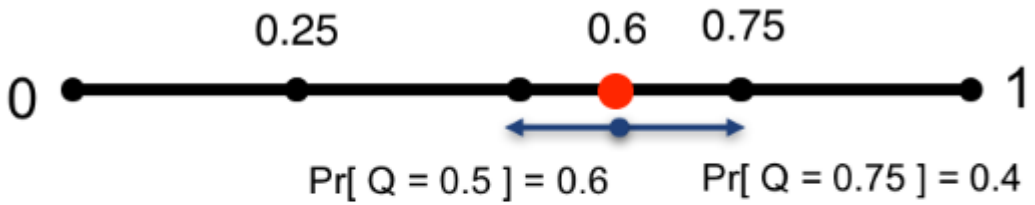
$$\mathbb{E}[C(x)] = x, \quad \mathbb{E}[\|C(x) - x\|^2] \leq \omega \|x\|^2, \quad \forall\, x \in \mathbb{R}^d$$

for constant $\omega \geq 0$, where the expectation is taken over the randomness of the compression operator C.

- Identity operator I (i.e. no compression) is an unbiased compressor with $\omega = 0$.

Example: random quantization

- Random quantization with 5 levels:



E[Q] = 0.6 × 0.5 + 0.4 * 0.75 = 0.6

Unbiased: $$\mathbb{E}[Q(x)] = \frac{Q_+(x) - x}{Q_+(x) - Q_-(x)} \cdot Q_-(x) + \frac{x - Q_-(x)}{Q_+(x) - Q_-(x)} \cdot Q_+(x) = x$$

Compressor Class

- Compressor class. Most compressors in literature are either **unbiased** or **contractive**

- We let $\mathcal{C}_\delta$ denote the set of unbiased compressors satisfying Assumption 4

Assump. 4 (Contractive compressor) The compression operator $C : \mathbb{R}^d \to \mathbb{R}^d$ satisfies

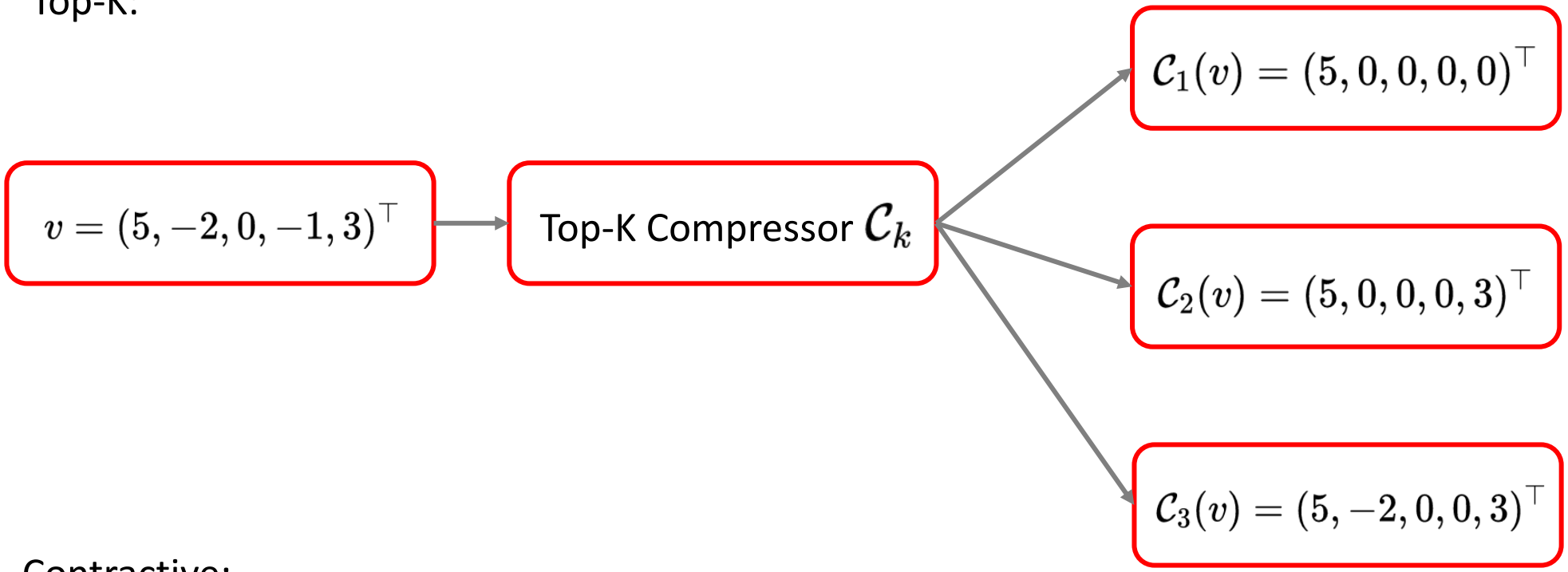
$$\mathbb{E}[\|C(x) - x\|^2] \leq (1 - \delta)\|x\|^2, \quad \forall\, x \in \mathbb{R}^d$$

for constant $\delta \in (0, 1]$, where the expectation is taken over the randomness of the compression operator C.

- Identity operator I (i.e. no compression) is a contractive compressor with $\delta = 1$.

Example: Top-K compressor

- Top-K:

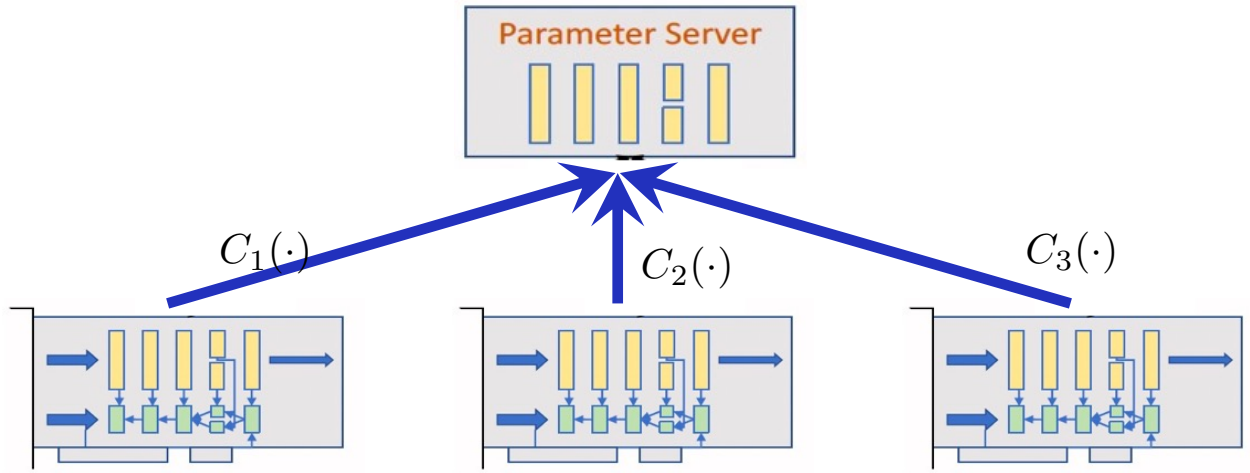


Contractive:

$$\|\mathcal{C}_k(v) - v\|^2 = \|v\|^2 - \|\mathcal{C}_k(v)\|^2 \leq (1 - k/d)\|v\|^2, \text{ i.e., } \delta = k/d \in (0, 1]$$

Algorithm Class

- Workers communicate directed with a central server. All iterations are synchronized.
- Each worker $i \in \{1, \cdots, n\}$ is endowed with C_i .
- Zero-respecting property: # non-zeros increase only by local update or comm. with the server



Complexity metric

- Let $\hat{x}_A^t$ denote the output of algorithm A after t communication rounds

- We define the complexity metric as

$$T_\epsilon(A, \{(f_i, C_i)\}_{i=1}^n) = \min \left\{ t \in \mathbb{N} : \mathbb{E}[f(\hat{x}_A^t)] - \min_x f(x) \leq \epsilon \right\}$$

- It indicates the smallest number of iterations to achieve an ϵ - accurate solution

Find optimal complexity

- The problem to find optimal complexity can be formulated into the problem

$$\inf_{A\in\mathcal{A}} \sup_{\{C_i\}_{i=1}^n\subseteq\mathcal{C}} \sup_{\{O_i\}_{i=1}^n\subseteq\mathcal{O}_{\sigma^2}} \sup_{\{f_i\}_{i=1}^n\subseteq\mathcal{F}_{\mu,L}} T_\epsilon(A,\{(f_i,C_i)\}_{i=1}^n)$$

- In other words, given a class of functions $\mathcal{F}_{\mu,L}$, gradient oracles $\mathcal{O}_{\sigma^2}$, compressors $\mathcal{C}$ (being $\mathcal{C}_\delta$ or $\mathcal{U}_\omega$), the formulation seeks the optimal algorithm and the convergence complexity it has.

Find optimal complexity

北京大学 PEKING UNIVERSITY

- The problem to find optimal complexity can be formulated into the problem

$$\inf_{A\in\mathcal{A}} \sup_{\{C_i\}_{i=1}^n\subseteq\mathcal{C}} \sup_{\{O_i\}_{i=1}^n\subseteq\mathcal{O}_{\sigma 2}} \sup_{\{f_i\}_{i=1}^n\subseteq\mathcal{F}_{\mu,L}} T_\epsilon(A,\{(f_i,C_i)\}_{i=1}^n)$$

- Very challenging to directly solve the above problem
- We will establish the lower bound first, then the upper bound, and finally show they match each other
- The algorithm to achieve the lower bound is optimal

北京大学
PEKING UNIVERSITY

PART 03

Find Lower Bound

Lower bound with unbiased compressors

Theorem 1 (Unbiased compression)

For every $\Delta, L > 0$, $n \geq 2$, $\omega \geq 0$, $\sigma > 0$, $T \geq (1+\omega)^2$, there exists a set of local loss functions $\{f_i\}_{i=1}^n \subseteq \mathcal{F}_{\mu,L}$, stochastic gradient oracles $\{O_i\}_{i=1}^n \subseteq \mathcal{O}_{\sigma^2}$, ω-unbiased compressors $\{C_i\}_{i=1}^n \subseteq \mathcal{U}_\omega$, such that for any algorithm $A \in \mathcal{A}$ starting from a given constant $x^{(0)}$, the lower bound complexity is given by

$$\Omega\left(\frac{\sigma^2}{\mu n \epsilon} + (1+\omega)\sqrt{\frac{L}{\mu}}\ln\left(\frac{1}{\epsilon}\right)\right),$$

where ϵ is the desired accuracy.

Consistency with existing lower bound

$$\Omega\left(\frac{\sigma^2}{\mu n\epsilon}+(1+\omega)\sqrt{\frac{L}{\mu}}\ln\left(\frac{1}{\epsilon}\right)\right)$$

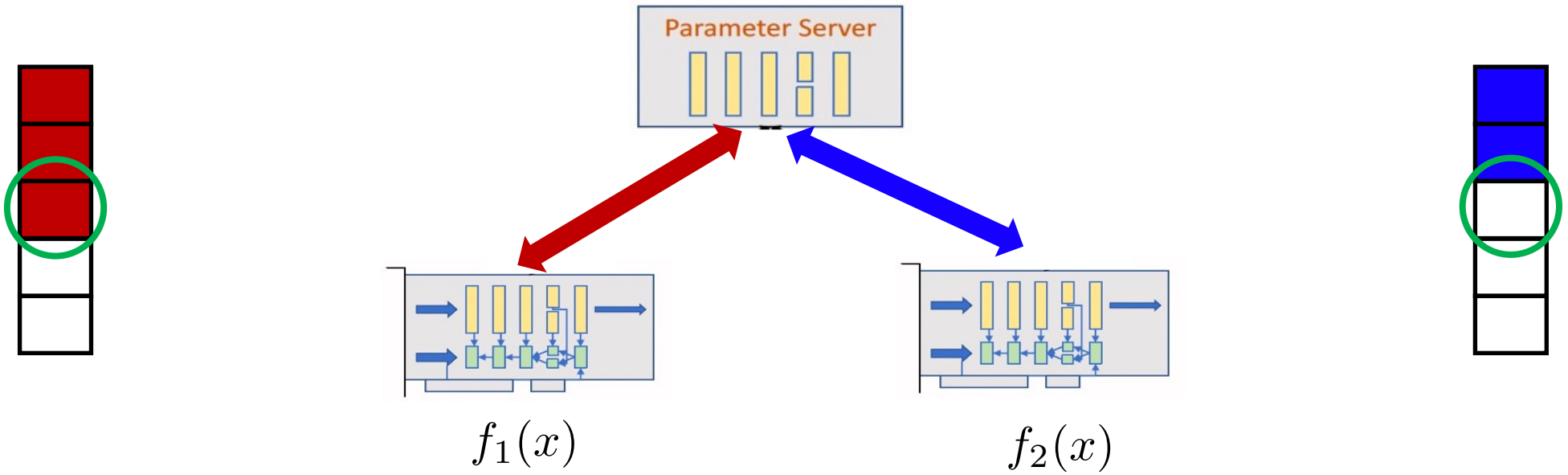
σ^2 is the gradient noise, n is the number of nodes, and ω gauges the compression ratio

- When $n=1$ and $\omega=0$, it recovers the lower bound in stochastic strongly-convex optimization
- When $n=1$, $\omega=0$ and $\sigma^2=0$, it recovers Nesterov lower bound

$$(n=1,\omega=0)\quad \Omega\left(\frac{\sigma^2}{\mu\epsilon}+\sqrt{\frac{L}{\mu}}\ln\left(\frac{1}{\epsilon}\right)\right) \qquad (n=1,\omega=0,\sigma=0)\quad \Omega\left(\sqrt{\frac{L}{\mu}}\ln\left(\frac{1}{\epsilon}\right)\right)$$

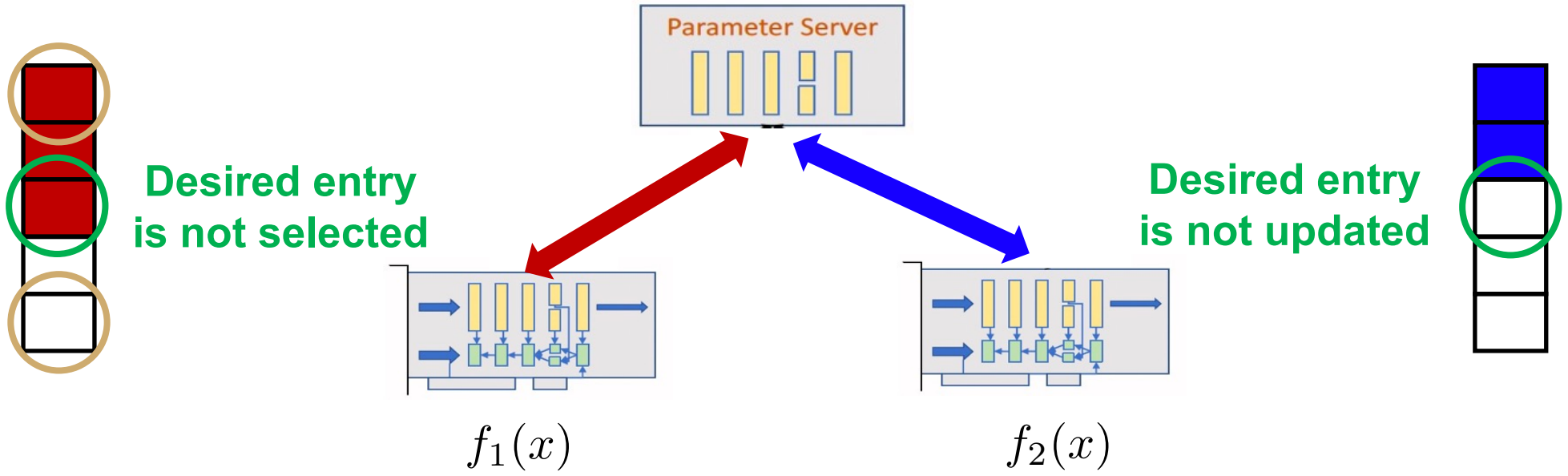
Distributed optimization with compression

- We construct adversarial objective functions
- The odd element in $f_2(x)$ can only be updated by receiving corresponding element from worker 1



Intuition behind lower bound

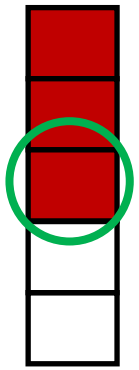
- With compression, the desired entry may not be able to communicate



- It explains why communication compression converges slower than vanilla distributed SGD

Intuition behind lower bound

- What's the probability that the desired entry is sent?



Desired entry is not selected

- We constructed an adversarial compressor: **rand-s**
- In rand-s compressor, s elements will be uniformly randomly chosen from all d elements
- Rand-s is an unbiased compressor (after scale) with $\omega = \frac{d}{s} - 1$
- The desired entry is sent with probability $s/d = (1+\omega)^{-1}$

Intuition behind lower bound

- In expectation, the desired entry will experience $(1+\omega)$ rounds to be sent

- Recall the lower bound complexity of standard distributed optimization as

$$\Omega\left(\sqrt{\frac{L}{\mu}}\ln\left(\frac{1}{\epsilon}\right)\right)$$

- Since sending the desired entry require $(1+\omega)$ rounds, the lower bound with compression is

$$\Omega\left((1+\omega)\sqrt{\frac{L}{\mu}}\ln\left(\frac{1}{\epsilon}\right)\right)$$

Lower bound with contractive compressor

北京大学 PEKING UNIVERSITY

Theorem 2 (Contractive compression)

For every $\Delta, L > 0$, $n \geq 2$, $\omega \geq 0$, $\sigma > 0$, $T \geq (1+\omega)^2$, there exists a set of local loss functions $\{f_i\}_{i=1}^n \subseteq \mathcal{F}_{\mu,L}$, stochastic gradient oracles $\{O_i\}_{i=1}^n \subseteq \mathcal{O}_{\sigma^2}$, ω-unbiased compressors $\{C_i\}_{i=1}^n \subseteq \mathcal{C}_\delta$, such that for any algorithm $A \in \mathcal{A}$ starting from a given constant $x^{(0)}$, it holds that

$$\mathbb{E}[f(\hat{x}) - f^\star] = \Omega\left(\frac{\sigma^2}{\mu n \epsilon} + \frac{1}{\delta}\sqrt{\frac{L}{\mu}}\ln\left(\frac{1}{\epsilon}\right)\right),$$

北京大学
PEKING UNIVERSITY

PART 04

Algorithm to Attain Lower Bounds

Optimal algorithm for distributed optimization

- Recall the problem $\min_{x\in\mathbb{R}^d} \quad f(x)=\frac{1}{n}\sum_{i=1}^{n} f_i(x), \quad \text{where} \quad f_i(x)=\mathbb{E}_{\xi_i\sim D_i}F(x;\xi_i).$
- We first consider the scenario in which the true gradient $\nabla f_i(x)$ is known and no compression is used
- The optimal algorithm is Nesterov Acceleration. For k=1,2,…,T

$$
\left.
\begin{aligned}
z^k &= \frac{1}{\gamma}x^k + (\frac{1}{p}-\frac{1}{\gamma})x^{k-1} + (1-\frac{1}{p})z^{k-1} \\
y^k &= (1-\frac{\gamma}{p})x^k + \frac{\gamma}{p}z^k
\end{aligned}
\right\} \text{(extrapolation)}
$$

$$g_i^k = \nabla f_i(y^k) \qquad \text{(true gradient)}$$

$$g^k = \frac{1}{n}\sum_{i=1}^{n} g_i^k \qquad \text{(precise communication)}$$

$$x^{k+1} = y^k - \frac{\eta}{p}g^k \qquad \text{(gradient descent)}$$

Extension to stochastic and compressed scenario

- A trivial extension: For k = 1,2,…,T

$$
\left.
\begin{aligned}
z^k &= \frac{1}{\gamma}x^k + (\frac{1}{p} - \frac{1}{\gamma})x^{k-1} + (1 - \frac{1}{p})z^{k-1} \\
y^k &= (1 - \frac{\gamma}{p})x^k + \frac{\gamma}{p}z^k
\end{aligned}
\right\} \text{(extrapolation)}
$$

$$
g_i^k = \nabla F(y^k; \xi_i^k) \qquad \text{(stochastic gradient)}
$$

$$
g^k = \frac{1}{n}\sum_{i=1}^{n} C_i(g_i^k) \qquad \text{(communication compression)}
$$

$$
x^{k+1} = y^k - \frac{\eta}{p}g^k \qquad \text{(gradient descent)}
$$

- Does not work well. Stochastic gradient and compression introduces too much noise.

Upper bound and nearly-optimal algorithms

- A refined algorithm: For k = 1,2,…,T/R

$$
\left.
\begin{aligned}
z^k &= \frac{1}{\gamma}x^k + (\frac{1}{p} - \frac{1}{\gamma})x^{k-1} + (1 - \frac{1}{p})z^{k-1} \\
y^k &= (1 - \frac{\gamma}{p})x^k + \frac{\gamma}{p}z^k
\end{aligned}
\right\} \quad \text{(extrapolation)}
$$

$$
g_i^k = \frac{1}{R}\sum_{r=1}^{R} \nabla F(y^k; \xi_i^{(k,r)}) \qquad \text{(large-batch stochastic gradient)}
$$

$$
g^k = \frac{1}{n}\sum_{i=1}^{n} \mathrm{MSC}(g_i^k, R) \qquad \text{(multi-step compression)}
$$

$$
x^{k+1} = y^k - \frac{\eta}{p}g^k \qquad \text{(gradient descent)}
$$

- The communication and sample budget are the same as previous algorithms

Upper bound and nearly-optimal algorithms

北京大学 PEKING UNIVERSITY

- Multi-step compression (MSC) protocol:

Input: Origical vector v and compression round R

Initialize $v^{(0)} = 0$;

for $r = 0, \cdots, R-1$ **do**

compress $v - v^{(r)}$ into $c^{(r)} = C(v - v^{(r)})$

send $c^{(r)}$ to the target

update $v^{(r+1)} = v^{(r)} + c^{(r)}$

end for

- With MSC, we can prove

$$v^{(R)} = \sum_{r=0}^{R-1} c^{(r)}$$

$$\mathbb{E}\|v^{(R)} - v\|^2 \leq (1-\delta)^R \|v\|^2$$

- It holds that

$$v^{(R)} \to v \text{ as } R \to \infty$$

Upper bound and nearly-optimal algorithms

北京大学 PEKING UNIVERSITY

- A refined algorithm NEOLITHIC: For k = 1,2,…,T/R

$$
\left.
\begin{aligned}
z^k &= \frac{1}{\gamma}x^k + (\frac{1}{p} - \frac{1}{\gamma})x^{k-1} + (1 - \frac{1}{p})z^{k-1} \\
y^k &= (1 - \frac{\gamma}{p})x^k + \frac{\gamma}{p}z^k
\end{aligned}
\right\} \quad \text{(extrapolation)}
$$

$$
g_i^k = \frac{1}{R}\sum_{r=1}^{R} \nabla F(y^k; \xi_i^{(k,r)}) \qquad \text{(large-batch stochastic gradient)}
$$

$$
g^k = \frac{1}{n}\sum_{i=1}^{n} \mathrm{MSC}(g_i^k, R) \qquad \text{(multi-step compression)}
$$

$$
x^{k+1} = y^k - \frac{\eta}{p}g^k \qquad \text{(gradient descent)}
$$

- As $R \to \infty$, the refined algorithm will approach to vanilla Nesterov; the introduced noise can be controlled

Upper bound and nearly-optimal algorithms

北京大学 PEKING UNIVERSITY

Theorem 1 (Upper bound)

(Informal) When utilizing unbiased compressors, the algorithm converges at

$$\mathbb{E}[f(x^K) - f^\star] = \tilde{\mathcal{O}}\left(\frac{\sigma^2}{\mu n \epsilon} + (1+\omega)\sqrt{\frac{L}{\mu}}\ln\left(\frac{1}{\epsilon}\right)\right),$$

where $\tilde{\mathcal{O}}(\cdot)$ hides logrithm terms that are independent of ϵ. When using contractive compressors, the algorithm converges at

$$\mathbb{E}[f(x^K) - f^\star] = \tilde{\mathcal{O}}\left(\frac{\sigma^2}{\mu n \epsilon} + \frac{1}{\delta}\sqrt{\frac{L}{\mu}}\ln\left(\frac{1}{\epsilon}\right)\right).$$

Upper bound and nearly-optimal algorithms

$$\inf_{A\in\mathcal{A}} \sup_{\{C_i\}_{i=1}^n\subseteq\mathcal{C}} \sup_{\{O_i\}_{i=1}^n\subseteq\mathcal{O}_{\sigma^2}} \sup_{\{f_i\}_{i=1}^n\subseteq\mathcal{F}_{\mu,L}} T_\epsilon(A,\{(f_i,C_i)\}_{i=1}^n)$$

- The lower bound and upper bound are nearly matched

$$\begin{aligned}\mathbb{E}[f(\hat{x})-f^\star] &= \Omega\left(\frac{\sigma^2}{\mu n\epsilon}+(1+\omega)\sqrt{\frac{L}{\mu}}\ln\left(\frac{1}{\epsilon}\right)\right)\\ &= \tilde{\mathcal{O}}\left(\frac{\sigma^2}{\mu n\epsilon}+(1+\omega)\sqrt{\frac{L}{\mu}}\ln\left(\frac{1}{\epsilon}\right)\right)\end{aligned}$$

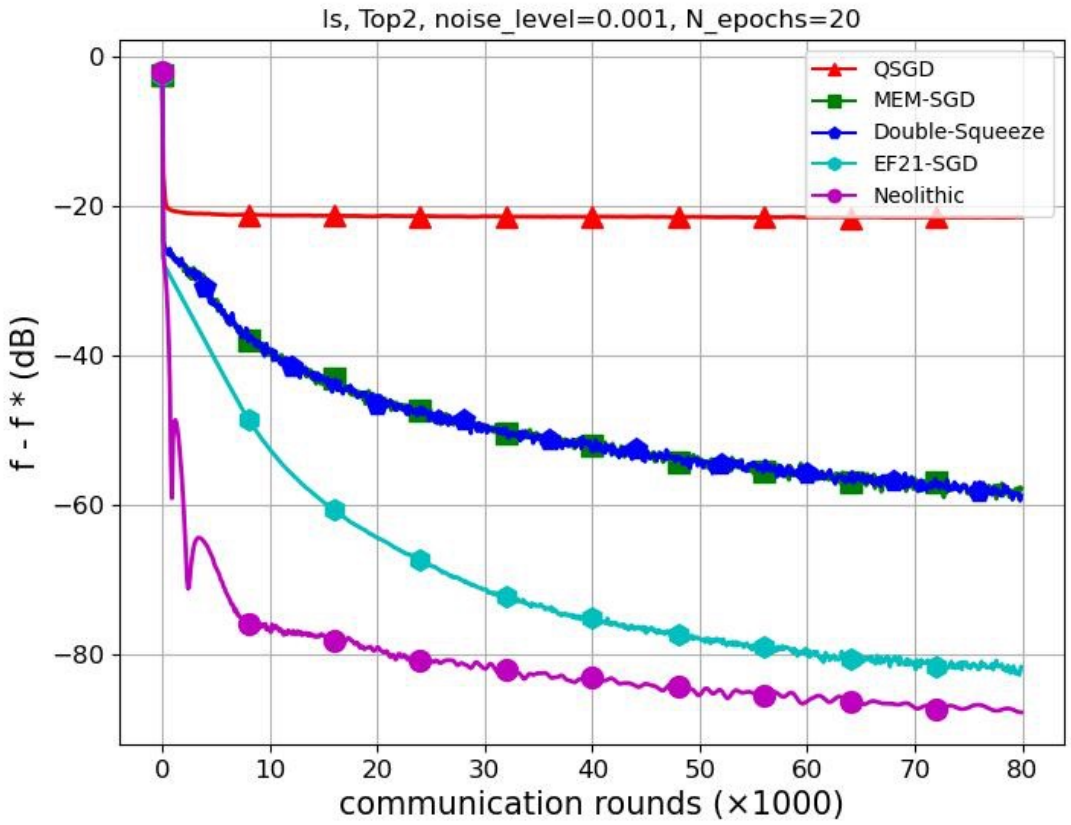
- **The established lower bound is tight. Our algorithm is optimal.**

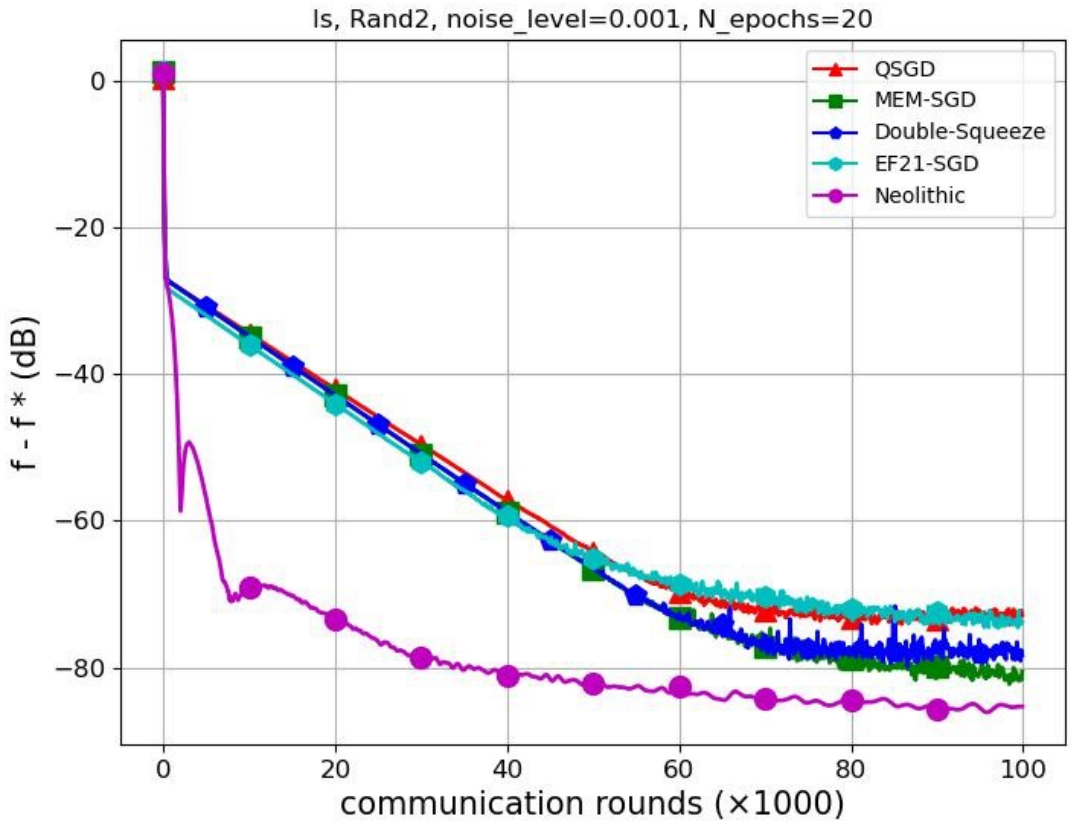
北京大学
PEKING UNIVERSITY

PART 05

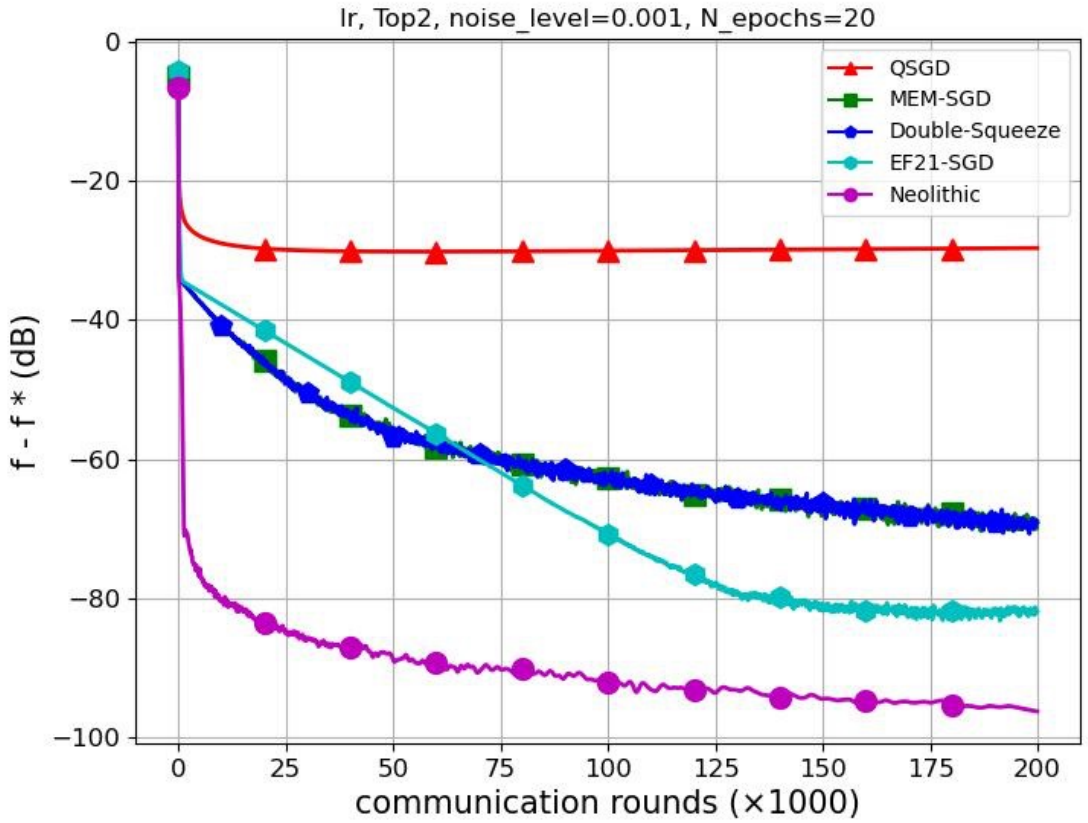
Empirical Studies

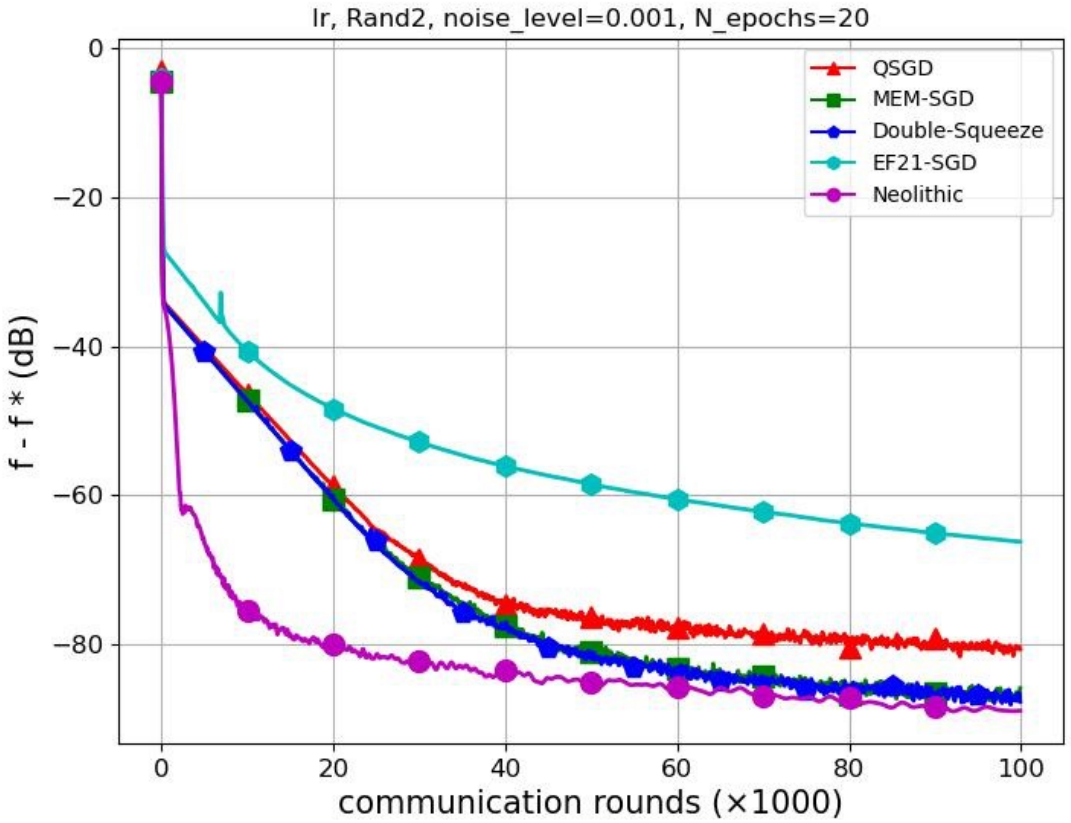
Least square with synthetic data





Logistic regression with synthetic data





北京大学
PEKING UNIVERSITY

PART 05

More results

Unbiased compression on generally-convex and non-convex scenarios

北京大学 PEKING UNIVERSITY

- We also study generally-convex and non-convex scenarios

Unbiased compressor

Method	NC	GC	SC
L.B.	$\Omega\left(\frac{L\Delta_f\sigma^2}{n\epsilon^2}+\frac{(1+\omega)L\Delta_f}{\epsilon}\right)$	$\Omega\left(\frac{\Delta_x\sigma^2}{n\epsilon^2}+\frac{(1+\omega)\sqrt{L\Delta_x}}{\sqrt{\epsilon}}\right)$	$\Omega\left(\frac{\sigma^2}{\mu n\epsilon}+(1+\omega)\sqrt{\frac{L}{\mu}}\ln\left(\frac{\mu\Delta_x}{\epsilon}\right)\right)$
Ours	$\tilde{\mathcal{O}}\left(\frac{L\Delta_f\sigma^2}{n\epsilon^2}+\frac{(1+\omega)L\Delta_f}{\epsilon}\right)$	$\tilde{\mathcal{O}}\left(\frac{\Delta_x\sigma^2}{n\epsilon^2}+\frac{(1+\omega)\sqrt{L\Delta_x}}{\sqrt{\epsilon}}\right)$	$\tilde{\mathcal{O}}\left(\frac{\sigma^2}{\mu n\epsilon}+(1+\omega)\sqrt{\frac{L}{\mu}}\ln\left(\frac{1}{\epsilon}\right)\right)$

- Lower bound and upper bound are **nearly matched** (up to logarithm terms)

- **The lower bound is tight. Our proposed NEOLITHIC is nearly optimal**

Contractive compression on generally-convex and non-convex scenarios

北京大学 PEKING UNIVERSITY

- We also study generally-convex and non-convex scenarios

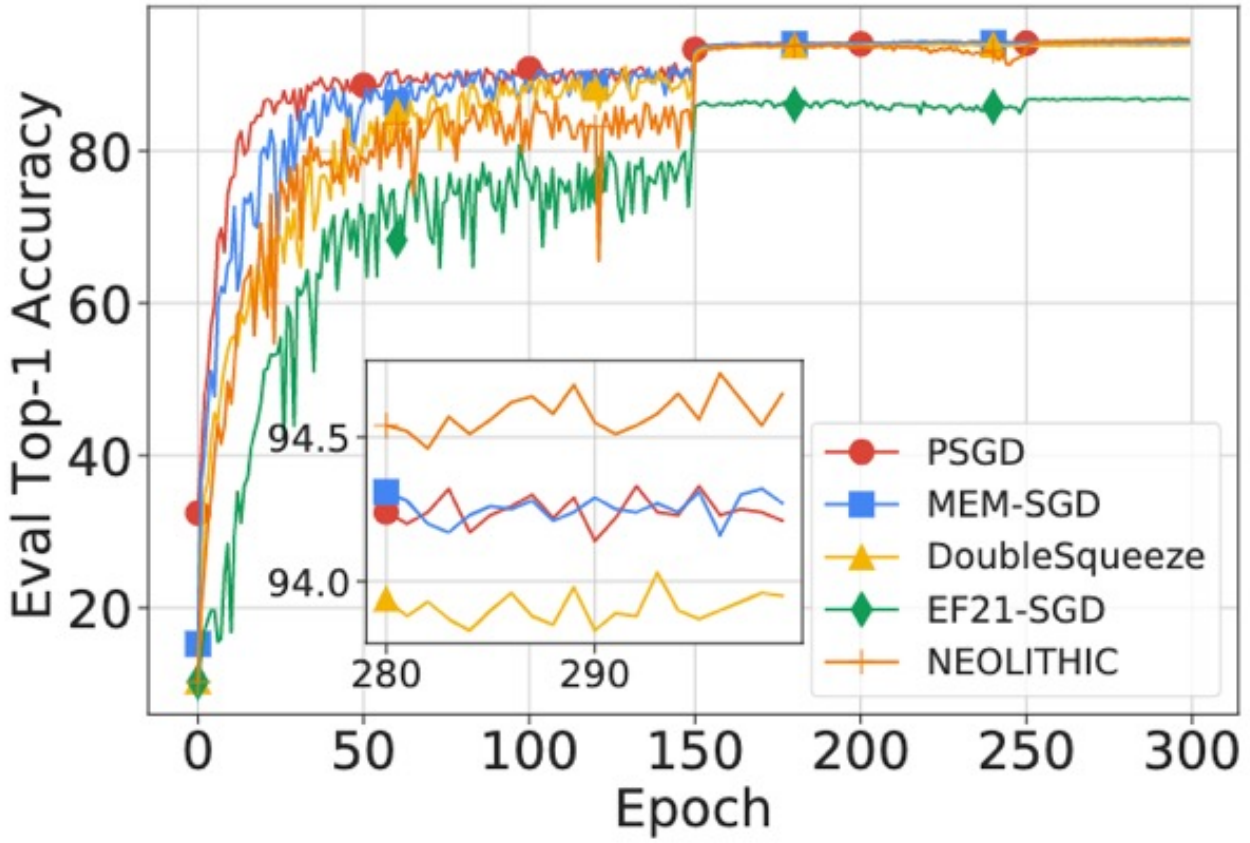
Contractive compressor

Method	NC	GC	SC
L.B.	$\Omega\left(\frac{L\Delta_f\sigma^2}{n\epsilon^2}+\frac{L\Delta_f}{\delta\epsilon}\right)$	$\Omega\left(\frac{\Delta_x\sigma^2}{n\epsilon^2}+\frac{\sqrt{L\Delta_x}}{\delta\sqrt{\epsilon}}\right)$	$\Omega\left(\frac{\sigma^2}{\mu n\epsilon}+\frac{1}{\delta}\sqrt{\frac{L}{\mu}}\ln\left(\frac{\mu\Delta_x}{\epsilon}\right)\right)$
Ours	$\tilde{\mathcal{O}}\left(\frac{L\Delta_f\sigma^2}{n\epsilon^2}+\frac{L\Delta_f}{\delta\epsilon}\ln\left(\frac{1}{\epsilon}\right)\right)^{\ddagger}$	$\tilde{\mathcal{O}}\left(\frac{\Delta_x\sigma^2}{n\epsilon^2}+\frac{\sqrt{L\Delta_x}}{\delta\sqrt{\epsilon}}\ln\left(\frac{1}{\epsilon}\right)\right)$	$\tilde{\mathcal{O}}\left(\frac{\sigma^2}{\mu n\epsilon}+\frac{1}{\delta}\sqrt{\frac{L}{\mu}}\ln\left(\frac{1}{\epsilon}\right)\right)$

- Lower bound and upper bound are **nearly matched** (up to logarithm terms)
- **The lower bound is tight. Our proposed NEOLITHIC is nearly optimal**

Deep learning experiments

- 8 workers, 1% compression ratio (top-k compressors), minibatch=128, R=2, ResNet18/ResNet20



Deep learning experiments

北京大学 PEKING UNIVERSITY

- 8 workers; 4-bit quantization (unbiased); minibatch=128; R=2

Table 2: Accuracy comparison with different algorithms on CIFAR-10.

Methods	ResNet18	ResNet20
PSGD	93.99 ± 0.52	91.62 ± 0.13
QSGD	92.86 ± 0.34	90.24± 0.22
MEM-SGD	94.47 ± 0.27	91.36 ± 0.07
Double-Squeeze	93.35 ± 0.39	90.89 ± 0.14
NEOLITHIC	**93.87 ± 0.46**	**91.25 ± 0.14**

Summary

北京大学 PEKING UNIVERSITY

- Compression is critical to save communication in distributed learning
- We establish the optimal complexity for distributed learning with communication compression
- We propose NEOLITHIC to nearly attain this optimal complexity
- Our results hold for non-convex, generally-convex, and strongly-convex scenarios

References

X. Huang, Y. Chen, W. Yin, K. Yuan, "Lower Bounds and Nearly Optimal Algorithms in Distributed Learning with Communication Compression", NeurIPS 2022

Y. He, X. Huang, Y. Chen, W. Yin, K. Yuan, “Lower Bounds and Accelerated Algorithms in Distributed Optimization with Communication Compression”, arXiv 2305.07612

北京大学
PEKING UNIVERSITY

Thank you!

Kun Yuan homepage: https://kunyuan827.github.io/

We have openings for PhD students and PostDocs