Lower Bounds and Nearly Optimal Algorithms in Distributed Learning with Communication Compression

Wotao Yin
DAMO Academy, Alibaba Group
Joint work with

Xinmeng Huang (UPenn)

Yiming Chen (Alibaba)

Kun Yuan (Alibaba)
Distributed learning

- A network of $n$ nodes (GPUs) collaborate to solve the problem:

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

- Each component $f_i : \mathbb{R}^d \to \mathbb{R}$ is local and private to node $i$

- Random variable $\xi_i$ denotes the local data that follows distribution $D_i$

- Each local distribution $D_i$ may be different; data heterogeneity exists
Vanilla parallel stochastic gradient descent (PSGD)

- 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

\[ g_i^{(k)} = \nabla F(x^{(k)}; \xi_i^{(k)}) \quad \text{(Local compt.)} \]

\[ x^{(k+1)} = x^{(k)} - \frac{\gamma}{n} \sum_{i=1}^{n} g_i^{(k)} \quad \text{(Global comm.)} \]
Expensive communication overhead in PSGD

- **Global average** incurs $O(n)$ comm. overhead; **proportional to network size** $n$
- Each node sends a **full model** (or gradient) to the server; **proportional to dimension** $d$
- Each node interacts with the server at **every** iteration; **proportional to iteration numbers**
Huge Communication overhead in PSGD

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

Methodologies to save communication

- **Global average** incurs $O(n)$ comm. overhead; proportional to network size $n$

  [Decentralized communication]

- Each node interacts with the server at **every** iteration; proportional to iteration numbers

  [Lazy communication]

- Each node sends a **full model** (or gradient) to the server; proportional to dimension $d$

  [Communication compression]
Decentralized SGD (DSGD)

DSGD Algorithm over one-peer exponential graphs

\[
x_i^{(k+\frac{1}{2})} = x_i^{(k)} - \gamma \nabla F(x_i^{(k)}; \xi_i^{(k)}) \quad \text{(Local update)}
\]

\[
x_i^{(k+1)} = \sum_{j \in N_i} w_{ij} x_j^{(k+\frac{1}{2})} \quad \text{(Partial averaging)}
\]

Takes $O(1)$ comm. overhead

DSGD is more communication-efficient than PSGD

- We implement DSGD with BlueFog
- DSGD has **better linear speedup** than PSGD

![Diagram showing comparison between BlueFog and Horovod for different batch sizes and number of GPUs. The diagram illustrates that BlueFog has a lower communication-to-computation ratio compared to Horovod, especially for larger batch sizes and more GPUs.](image)


Github address: https://github.com/Bluefog-Lib/bluefog
Lazy communication (Federated Average)

\[
x_i^{(k+\frac{1}{2})} = x_i^{(k)} - \gamma \nabla F(x_i^{(k)}; \xi_i^{(k)})
\]

(Local update)

\[
x_i^{(k+1)} = \begin{cases} 
  x_i^{(k+\frac{1}{2})} & \text{if } \text{mod}(k, \tau) \neq 0 \\
  \frac{1}{n} \sum_{j=1}^{n} x_j^{(k+\frac{1}{2})} & \text{if } \text{mod}(k, \tau) = 0
\end{cases}
\]

(Lazy comm. )

• Nodes communicate once every \( \tau \) iterations [Konecny et .al. 2015, 2016]

• Or nodes communicate when necessary, i.e., the lazily aggregated gradient [Chen et. al. 2018]

• In ProxSkip [Mishchenko et. al., 2022], lazy strategy is proved to save communication


[Mishchenko et.al. 2016] K. Mishchenko et.al., “ProxSkip: Yes! Local gradient steps provably lead to communication acceleration! Finally!”, ICML 2022

This talk will study distributed learning with communication compression
Communication 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

8 bit

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

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

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

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

• There are extensive studies in distributed learning with communication compression

Compressor
StaticQuant  AdaptQuant  Random-K  Top-K  …

Algorithm development
Compress model  Compress gradient  Compress difference  Error feedback  …

Compression strategy
Unidirectional compression  Bidirectional 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.

• How to understand the performance of different algorithms?
Function class $\mathcal{F}_{\Delta,L}$ and gradient oracle class $\mathcal{O}_{\sigma^2}$

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

**Assumption 1 (Smoothness)** Each local objective $f_i$ has $L$-Lipschitz gradient, i.e.,

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L \|x - y\|, \quad \forall \ x, y \in \mathbb{R}^d,$$

and $f(x^{(0)}) - \inf_{x \in \mathbb{R}^d} f(x) \leq \Delta$ 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

**Assump. 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 $\mathcal{U}_\omega$

- **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 $C : \mathbb{R}^d \rightarrow \mathbb{R}^d$ 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 $\omega = 0$. 
Compressor class $\mathcal{U}_\omega$ : examples

- Example I (random quantization [Alistarh et. al. 2017]).

For any $\mathbf{v} \in \mathbb{R}^n$, $C(\mathbf{v})$ (with tuning parameter $s$) is defined as $C(\mathbf{v}) = ||\mathbf{v}||_2 \cdot \text{sgn}(v_k) \cdot \xi(\mathbf{v}_k)_{1 \leq k \leq d}$ where if $|v_k|/||\mathbf{v}|| \in [\ell/s, (\ell+1)/s]$,

$$\xi(\mathbf{v}_k) = \begin{cases} 
(\ell + 1)/s & \text{with prob. } s|v_k|/||\mathbf{v}|| - \ell \\
\ell/s & \text{otherwise}
\end{cases}$$

The associated unbiasedness parameter is $\omega = \min\{d/s^2, \sqrt{d}/s\}$.

- Example II (random sparsification [Wangni et.al., 2018]).

For any $\mathbf{v} \in \mathbb{R}^n$, $C(\mathbf{v})$ (with tuning parameter $\epsilon$) is defined as $C(\mathbf{v}) = ||\mathbf{v}||_2 \cdot \text{Bernoulli}(p_k/p_k)_{1 \leq k \leq d}$ where $\{p_k\}_{1 \leq k \leq d}$ are the solution to

$$\min_{p_k} \sum_{k=1}^{d} p_k \quad \text{s.t.} \quad \sum_{k=1}^{d} v_k^2/p_k \leq (1 + \epsilon)||\mathbf{v}||^2.$$ 

The associated unbiasedness parameter is $\omega = 1 + \epsilon$ (if the solution exists).


Compressor class \( C_\delta \)

- We let \( C_\delta \) denote the family of contractive 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 \( \delta = 1 \).
Compressor class $\mathcal{C}_\delta$

- Example I (top-k/rand-k [Stich et. al., 2018]).

For any $\mathbf{v} \in \mathbb{R}^n$, $\mathcal{C}(\mathbf{v})$ (with tuning parameter $k$) is defined by

maintaining the largest $k$ entries or random $k$ entries, and zeroing out the rest.

The associated contraction parameter is $\delta = d/k$.

- Example II (random sketching [Stich, 2020]).

For any $\mathbf{v} \in \mathbb{R}^n$, $\mathcal{C}(\mathbf{v}) = S(S^\top S)^\dagger S^\top \mathbf{v}$ with a possibly random matrix $S$ (usually sparse or low-rank). The associated contraction parameter is $\delta = 1 - \|I - S(S^\top S)^\dagger S^\top\|^2_2$.


Algorithm class $\mathcal{A}$

- Workers communicate directed with a central server. All iterations are synchronized.
- Each worker $i \in \{1, \ldots, n\}$ is endowed with $C_i$. Server is endowed with compressor $C_0$.
- If some $C_i = I$, then worker $i$ conducts no compression. If $C_0 = I$, then compression is unidirectional.
- Zero-respecting property: # non-zeros increase only by local update or comm. with the server.
## Existing convergence rates (non-convex)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Convergence Rate</th>
<th>Compression</th>
<th>Trans. Compl.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q-SGD</td>
<td>$O \left( \frac{(1+\omega)^{0.5} \sigma + \omega^{0.5} b}{\sqrt{nT}} \right)$</td>
<td>Unidirectional i.i.d, Unbiased</td>
<td>–</td>
</tr>
<tr>
<td>MEM-SGD</td>
<td>$O \left( \frac{\sigma}{\sqrt{nT}} + \frac{G^{2/3}}{\delta^{2/3} T^{2/3}} + \frac{1}{T} \right)$</td>
<td>Unidirectional Contractive</td>
<td>$O(\frac{n^3}{\delta^4})$</td>
</tr>
<tr>
<td>Double Squeeze</td>
<td>$O \left( \frac{\sigma}{\sqrt{nT}} + \frac{G^{2/3}}{\delta^{4/3} T^{2/3}} + \frac{1}{T} \right)$</td>
<td>Bidirectional Contractive</td>
<td>$O(\frac{n^3}{\delta^8})$</td>
</tr>
<tr>
<td>CSER</td>
<td>$O \left( \frac{\sigma}{\sqrt{nT}} + \frac{G^{2/3}}{\delta^{2/3} T^{2/3}} + \frac{1}{T} \right)$</td>
<td>Unidirectional Contractive</td>
<td>$O(\frac{n^3}{\delta^4})$</td>
</tr>
<tr>
<td>EF21-SGD</td>
<td>$O \left( \frac{\sigma}{\sqrt{\delta^3 T}} + \frac{1}{\delta T} \right)$</td>
<td>Unidirectional Contractive</td>
<td>–</td>
</tr>
</tbody>
</table>
Existing convergence rate (non-convex)

- When $T$ is sufficiently large so that $\sigma/\sqrt{nT}$ dominates the rate, the algorithm achieves **linear speedup**

  \[
  \text{To guarantee } \frac{\sigma}{\sqrt{nT}} \leq \epsilon, \text{ we require } T \geq \frac{\sigma^2}{\epsilon n^2} \text{ (inversely prop. to } n)\]

EF21-SGD and Q-SGD cannot achieve linear speedup

- **Transient iterations** refer to those before an algorithm achieves linear speedup
  - Reflect sensitivity to compressions
  - The shorter the better

![Graph showing the convergence of Quantized SGD and Parallel SGD with Transient Iterations highlighted.](image-url)
Existing convergence rate (non-convex)

• When $T$ is sufficiently large so that $\sigma/\sqrt{nT}$ dominates the rate, the algorithm achieves **linear speedup**

  To guarantee $\frac{\sigma}{\sqrt{nT}} \leq \epsilon$, we require $T \geq \frac{\sigma^2}{n\epsilon^2}$ (inversely prop. to $n$)

EF21-SGD and Q-SGD cannot achieve linear speedup

• **Transient iterations** refer to those before an algorithm achieves linear speedup
  
  o Reflect sensitivity to compressions
  
  o The shorter the better

• Mem-SGD, Double Squeeze, and CSER additionally require bounded gradients $\mathbb{E}_i \|\nabla f_i(x)\|^2 \leq G$
What is the optimal convergence rate for approaches using $C_\delta$ or $\mathcal{U}_\omega$?
Mathematical formulation

To address these questions, we consider the following formulation

$$\inf_{A \in \mathcal{A}} \sup_{\{C_i\}_{i=0}^n \subseteq \mathcal{C}} \sup_{\{O_i\}_{i=1}^n \subseteq \mathcal{O}_{\sigma^2}} \sup_{\{f_i\}_{i=1}^n \subseteq \mathcal{F}_{\Delta,L}} \mathbb{E}[\|\nabla f(\hat{x}_A,\{f_i\}_{i=1}^n,\{O_i\}_{i=1}^n,\{C_i\}_{i=0}^n,T)\|^2].$$

where $\hat{x}_A,\{f_i\}_{i=1}^n,\{O_i\}_{i=1}^n,\{C_i\}_{i=0}^n,T$ are the output of algorithm $A$ with no more than $T$ gradient queries and communications on each worker.

In other words, given a class of functions $\mathcal{F}_{\Delta,L}$, 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 rate it has.
Why supremum over compressors?

To gauge the algorithmic performance over the entire family of unbiased or contractive compressors

To gauge the algorithmic performance without further assumptions on compressors
Theorem 1 (Unidirectional 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 $\{f_i\}_{i=1}^{n} \subseteq \mathcal{F}_{\Delta, L}$, stochastic gradient oracles $\{O_i\}_{i=1}^{n} \subseteq \mathcal{O}_{\sigma^2}$, $\omega$-unbiased compressors $\{C_i\}_{i=0}^{n} \subseteq \mathcal{U}_\omega$ with $C_0 = I$, such that for any algorithm $A \in \mathcal{A}$ starting from a given constant $x^{(0)}$, it holds that

$$
\mathbb{E}[\|\nabla f(x_A, \{f_i\}_{i=1}^{n}, \{O_i\}_{i=1}^{n}, \{C_i\}_{i=0}^{n}, T)\|^2] = \Omega \left( \left( \frac{\Delta L \sigma^2}{nT} \right)^{\frac{1}{2}} + \frac{(1 + \omega)\Delta L}{T} \right).
$$

- When $n = 1$ and $\omega = 0$, it recovers the bound in stochastic non-convex optimization [Arjevani 2022]
- When $n = 1$, $\omega = 0$ and $\sigma^2 = 0$, it recovers deterministic non-convex optimization [Carmon 2022]
Corollary 1 (Bidirectional unbiased compression)

Under the same settings, there exists a set of local objectives \( \{f_i\}_{i=1}^n \subseteq \mathcal{F}_{\Delta,L} \), stochastic gradient oracles \( \{O_i\}_{i=1}^n \subseteq \mathcal{O}_{\sigma^2} \), \( \omega \)-unbiased compressors \( \{C_i\}_{i=0}^n \subseteq \mathcal{U}_\omega \) such that for any algorithm \( A \in \mathcal{A} \) starting from \( x^{(0)} \), the same lower bound is also valid.

- Unidirectional and bidirectional unbiased compression share the same lower bound.
Theorem 2 (Unidirectional contractive compression)

For every $\Delta, L > 0$, $n \geq 2$, $\omega \geq 0$, $\sigma > 0$, $T \geq \delta^{-22}$, there exists a set of local loss functions $\{f_i\}_{i=1}^n \subseteq \mathcal{F}_{\Delta,L}$, stochastic gradient oracles $\{O_i\}_{i=1}^n \subseteq \mathcal{O}_{\sigma^2}$, $\omega$-unbiased compressors $\{C_i\}_{i=0}^n \subseteq \mathcal{U}_\omega$ with $C_0 = I$, such that for any algorithm $A \in \mathcal{A}$ starting from a given constant $x^{(0)}$, it holds that

$$\mathbb{E}[\|\nabla f(\hat{x}_A,\{f_i\}_{i=1}^n,\{O_i\}_{i=1}^n,\{C_i\}_{i=0}^n,T)\|^2] = \Omega \left( \left( \frac{\Delta L \sigma^2}{nT} \right)^{\frac{1}{2}} + \frac{\Delta L}{\delta T} \right).$$

- The same bound also holds for bidirectional contractive compression
Have the lower bound limits been attained by existing algorithms?
A big gap exists between established lower bound and existing upper bounds

For example, contractive lower bound tran. compl. \(O(n/\delta^2)\) is far shorter than existing ones
Can we develop new algorithms to (nearly) achieve these lower bounds?
Fast compressed communication (FCC)

- We propose a novel module named as fast compressed communication (FCC).
- FCC is compatible with both contractive and unbiased compressors.

**Algorithm 1** $v^{(k,R)} = \text{FCC}(v^{(k,*)}, C, R, \text{target receiver(s)})$

**Input:** The vector $v^{(k,*)}$ aimed to communicate at iteration $k$; a compressor $C$; rounds $R$; initial vector $v^{(k,0)} = 0$; target receiver(s);

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

- Compress $v^{(k,*)} - v^{(k,r)}$ into $c^{(k,r)} = C(v^{(k,*)} - v^{(k,r)})$
- Send $c^{(k,r)}$ to the target receiver(s)
- Update $v^{(k,r+1)} = v^{(k,r)} + c^{(k,r)}$

end for

return Variable $v^{(k,R)}$.  \(\triangleright\) The set $\{c^{(k,r)}\}_{r=0}^{R-1}$ will be sent to the receiver during the for-loop

\(\triangleright\) It holds that $v^{(k,R)} = \sum_{r=0}^{R-1} c^{(k,r)}$
Fast compressed communication (FCC)

• FCC module has $R$ rounds of compressions per call

• When $R = 1$, FCC reduces to a standard compression $v^{(k,1)} = C(v^{(k,*)})$

• When $R > 1$, FCC yields exponentially smaller errors. When $R \to \infty$, FCC yields **lossless** compression

---

**Lemma 1 (FCC property)**

Let $C$ be a $\delta$-contractive compressor and $v^{(k,R)} = \text{FCC}(v^{(k,*)}, C, R)$. It holds for any $R \geq 1$ and $v^{(k,*)} \in \mathbb{R}^d$ that

$$\mathbb{E}[\|v^{(k,R)} - v^{(k,*)}\|^2] \leq (1 - \delta)^R \|v^{(k,*)}\|^2, \quad \forall k = 0, 1, 2, \ldots$$

• FCC lies between a standard one-round compression and a lossless compression
The backbone: Double Squeeze

- Double Squeeze [Tang et. al., 2019] is effective to conduct uni-/bi-directional compression.

---

**Algorithm 1: Double Squeeze**

**Input:** Initialize $x^{(0)}$; learning rate $\gamma$; compression round $R$; $\nu^{(0)} = v^{(0)} = 0$, $\forall i \in [n]$

for $k = 0, 1, \ldots, K - 1$ do

**On all workers in parallel:**

Query stochastic gradients $\tilde{g}^{(k)}_i = O_i(x^{(k)}; \zeta_i^{(k,0)})$  
Error compensate $\tilde{g}^{(k)}_i = \tilde{g}^{(k)}_i + \nu_i^{(k)}$  
Update error $\nu_i^{(k+1)} = \tilde{g}^{(k)}_i - C_i(\tilde{g}^{(k)}_i)$  
▷ Gradient calculation

**On server:**

Error compensate $\tilde{g}^{(k)} = \frac{1}{n} \sum_{i=1}^{n} C_i(\tilde{g}^{(k)}_i) + \nu^{(k)}$  
Update error $\nu^{(k+1)} = \tilde{g}^{(k)} - C_0(\tilde{g}^{(k)})$  
▷ $C_i(\tilde{g}^{(k)}_i)$ received from workers

**On all workers in parallel:**

Update model parameter $x^{(k+1)} = x^{(k)} - \gamma C_0(\tilde{g}^{(k)})$  
▷ $C_0(\tilde{g}^{(k)})$ received from server

end for
NEOLITHIC: A nearly optimal algorithm

• Change 1: Replace the standard compression with R-round FCC compression

Algorithm 1: NEOLITHIC

Input: Initialize $x^{(0)}$; learning rate $\gamma$; compression round $R$; $v^{(0)} = v^{(0)}_i = 0, \forall i \in [n]$  
for $k = 0, 1, \ldots, K - 1$ do  
On all workers in parallel:  
Query stochastic gradients $\hat{g}_i^{(k)} = \frac{1}{R} \sum_{r=0}^{R-1} O_i(x^{(k)}; \zeta_i^{(k,r)})$  
Error compensate $\tilde{g}_i^{(k)} = \hat{g}_i^{(k)} + v^{(k)}_i$  
Update error $v^{(k+1)}_i = \tilde{g}_i^{(k)} - \text{FCC}(\tilde{g}_i^{(k)}, C_i, R, \text{server})$ 
▷ Gradient accumulation  
On server:  
Error compensate $\tilde{g}^{(k)} = \frac{1}{n} \sum_{i=1}^{n} \sum_{r=0}^{R-1} c_i^{(k,r)} + v^{(k)}$  
Update error $v^{(k+1)} = \tilde{g}^{(k)} - \text{FCC}(\tilde{g}^{(k)}, C_0, R, \text{all workers})$  
▷ $\{c_i^{(k,r)}\}$ received from workers  
On all workers in parallel:  
Update model parameter $x^{(k+1)} = x^{(k)} - \gamma \sum_{r=0}^{R-1} c^{(k,r)}$  
▷ $\{c^{(k,r)}\}$ received from server  
end for
NEOLITHIC: A nearly optimal algorithm

- Change 2: Conduct **R-batch gradient accumulation** to balance with R-round compression

---

**Algorithm 1: NEOLITHIC**

**Input:** Initialize $x^{(0)}$; learning rate $\gamma$; compression round $R$; $v^{(0)} = v_i^{(0)} = 0$, $\forall i \in [n]$

for $k = 0, 1, \cdots, K - 1$ do

**On all workers in parallel:**

Query stochastic gradients $\hat{g}_i^{(k)} = \frac{1}{R} \sum_{r=0}^{R-1} O_i(x^{(k)}; \zeta_i^{(k,r)})$ $\quad$ ▶ Gradient accumulation

Error compensate $g_i^{(k)} = \hat{g}_i^{(k)} + v_i^{(k)}$

Update error $v_i^{(k+1)} = g_i^{(k)} - \text{FCC}(\hat{g}_i^{(k)}, C_i, R, \text{server})$ $\quad$ ▶ Worker sends $\{c_i^{(k,r)}\}$ to server

**On server:**

Error compensate $\tilde{g}^{(k)} = \frac{1}{n} \sum_{i=1}^{n} \sum_{r=0}^{R-1} c_i^{(k,r)} + v^{(k)}$ $\quad$ ▶ $\{c_i^{(k,r)}\}$ received from workers

Update error $v^{(k+1)} = \tilde{g}^{(k)} - \text{FCC}(\tilde{g}^{(k)}, C_0, R, \text{all workers})$ $\quad$ ▶ Server sends $\{c^{(k,r)}\}$ to workers

**On all workers in parallel:**

Update model parameter $x^{(k+1)} = x^{(k)} - \gamma \sum_{r=0}^{R-1} c^{(k,r)}$ $\quad$ ▶ $\{c^{(k,r)}\}$ received from server

end for
NEOLITHIC: A nearly optimal algorithm

• NEOLITHIC can conduct either unidirectional or bidirectional compression

• NEOLITHIC is compatible with both unbiased and contractive compression

• For each iteration, NEOLITHIC conducts $R$ gradient calculations and $R$ compressions

• Given compression round budget $T$, we shall consider $T/R$ iterations in NEOLITHIC for fair comparison
Upper bounds for contractive compressors

Theorem. 3 (NEOLITHIC with bidirectional contractive compression)

Given any constants $n \geq 1, \delta \in (0, 1]$, assume $\frac{1}{n} \sum_{i=1}^{n} \|\nabla f_i(x) - \nabla f(x)\|^2 \leq b^2$ for any $x \in \mathbb{R}^d$, and let $x^{(k)}$ be generated by NEOLITHIC. By setting $R$ and the learning rate appropriately, it holds for any $K \geq 0$ and compressors $\{C_i\}_{i=0}^{n} \subseteq C_\delta$ that

$$
\frac{1}{K+1} \sum_{k=0}^{K} \mathbb{E}[\|\nabla f(x^{(k)})\|^2] = \tilde{O} \left( \left( \frac{\Delta L \sigma^2}{nT} \right)^{\frac{1}{2}} + \frac{\Delta L}{\delta T} \right),
$$

where $T = KR$ is the total number of gradient queries (communication rounds) on each worker.

- $\tilde{O}(\cdot)$ omits logarithmic terms
- Recall the established lower bound $\Omega \left( \left( \frac{\Delta L \sigma^2}{nT} \right)^{\frac{1}{2}} + \frac{\Delta L}{\delta T} \right)$, we find it is nearly attained
- Letting $C_0 = I$, the same lower bound for unidirectional contractive compression also holds
Upper bounds for unbiased compressors

Theorem. 4 (NEOLITHIC with bidirectional unbiased compression)

Under the same assumptions as in Theorem 1, it holds for any $K \geq 0$ and compressors $\{C_i\}_{i=0}^n \subseteq \mathcal{U}_\omega$ that

$$\frac{1}{K+1} \sum_{k=0}^{K} \mathbb{E}[\| \nabla f(x^{(k)}) \|^2] = \tilde{O}\left(\left(\frac{\Delta L \sigma^2}{nT}\right)^{\frac{1}{2}} + \frac{(1 + \omega)\Delta L}{T}\right).$$

This further leads to a transient complexity of $\tilde{O}(n(1 + \omega)^2)$.

• Recall the established lower bound $\Omega\left(\left(\frac{\Delta L \sigma^2}{nT}\right)^{\frac{1}{2}} + \frac{(1 + \omega)\Delta L}{T}\right)$, we find it is nearly attained by NEOLITHIC

• NEOLITHIC also attains the lower bound with unidirectional unbiased compression
NEOLITHIC (nearly) attains the optimal convergence rate

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Convergence Rate</th>
<th>Compression</th>
<th>Trans. Compl.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lower Bound</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Theorem 2</td>
<td>$\Omega \left( \frac{\sigma}{\sqrt{nT}} + \frac{1}{\delta T} \right)$</td>
<td>Uni/Bidirectional Contractive</td>
<td>$\mathcal{O}(n/\delta^2)$</td>
</tr>
<tr>
<td>Theorem 1</td>
<td>$\Omega \left( \frac{\sigma}{\sqrt{nT}} + \frac{1+\omega}{T} \right)$</td>
<td>Uni/Bidirectional Unbiased</td>
<td>$\mathcal{O}(n(1+\omega)^2)$</td>
</tr>
<tr>
<td><strong>Upper Bound</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Theorem 3</td>
<td>$\tilde{\Omega} \left( \frac{\sigma}{\sqrt{nT}} + \frac{1}{\delta T} \right)$</td>
<td>Uni/Bidirectional Contractive</td>
<td>$\tilde{\Omega}(n/\delta^2)$</td>
</tr>
<tr>
<td>Theorem 4</td>
<td>$\tilde{\Omega} \left( \frac{\sigma}{\sqrt{nT}} + \frac{1+\omega}{T} \right)$</td>
<td>Uni/Bidirectional Unbiased</td>
<td>$\tilde{\Omega}(n(1+\omega)^2)$</td>
</tr>
<tr>
<td>Q-SGD</td>
<td>$\mathcal{O} \left( \frac{(1+\omega)^{0.5} \sigma + \omega^{0.5} b}{\sqrt{nT}} \right)$</td>
<td>Unidirectional i.i.d. Unbiased</td>
<td>$\mathcal{O}(n/\delta^2)$</td>
</tr>
<tr>
<td>MEM-SGD</td>
<td>$\mathcal{O} \left( \frac{\sigma}{\sqrt{nT}} + \frac{G^{2/3}}{\delta^{2/3} T^{2/3}} + \frac{1}{T} \right)$</td>
<td>Unidirectional Contractive</td>
<td>$\mathcal{O}(n^3/\delta^4)$</td>
</tr>
<tr>
<td>Double Squeeze</td>
<td>$\mathcal{O} \left( \frac{\sigma}{\sqrt{nT}} + \frac{G^{2/3}}{\delta^{4/3} T^{2/3}} + \frac{1}{T} \right)$</td>
<td>Bidirectional Contractive</td>
<td>$\mathcal{O}(n^3/\delta^8)$</td>
</tr>
<tr>
<td>CSER</td>
<td>$\mathcal{O} \left( \frac{\sigma}{\sqrt{nT}} + \frac{G^{2/3}}{\delta^{2/3} T^{2/3}} + \frac{1}{T} \right)$</td>
<td>Unidirectional Contractive</td>
<td>$\mathcal{O}(n^3/\delta^4)$</td>
</tr>
<tr>
<td>EF21-SGD</td>
<td>$\mathcal{O} \left( \frac{\sigma}{\sqrt{\delta^3 T}} + \frac{1}{\delta T} \right)$</td>
<td>Unidirectional Contractive</td>
<td>$\mathcal{O}(n^3/\delta^4)$</td>
</tr>
</tbody>
</table>
Experiments: synthetic simulation

- We compare algorithms for **least square** and **logistic regression**, using rand-1 compressors and $R=4$

- Though with compression, NEOLITHIC almost matches with P-SGD (note P-SGD has no compression)
Experiments: image classification on Cifar-10

- 8 workers; top-k compressors (contractive); minibatch=128; R=2

Table 1: Accuracy comparison with different algorithms on CIFAR-10.

<table>
<thead>
<tr>
<th>Comp. ratio</th>
<th>Methods</th>
<th>ResNet18</th>
<th>ResNet20</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSGD</td>
<td>93.99 ± 0.52</td>
<td>91.62 ± 0.13</td>
</tr>
<tr>
<td>5%</td>
<td>MEM-SGD</td>
<td>94.35 ± 0.01</td>
<td>91.27 ± 0.08</td>
</tr>
<tr>
<td></td>
<td>Double-Squeeze</td>
<td>94.11 ± 0.14</td>
<td>90.73 ± 0.02</td>
</tr>
<tr>
<td></td>
<td>EF21-SGD</td>
<td>87.37 ± 0.49</td>
<td>65.82 ± 4.86</td>
</tr>
<tr>
<td></td>
<td>NEOLITHIC</td>
<td><strong>94.63 ± 0.09</strong></td>
<td><strong>91.43 ± 0.10</strong></td>
</tr>
<tr>
<td>1%</td>
<td>MEM-SGD</td>
<td>93.99 ± 0.11</td>
<td>89.68 ± 0.17</td>
</tr>
<tr>
<td></td>
<td>Double-Squeeze</td>
<td>93.54 ± 0.17</td>
<td>89.35 ± 0.04</td>
</tr>
<tr>
<td></td>
<td>EF21-SGD</td>
<td>67.78 ± 2.14</td>
<td>56.0 ± 2.257</td>
</tr>
<tr>
<td></td>
<td>NEOLITHIC</td>
<td><strong>94.155 ± 0.10</strong></td>
<td><strong>89.82 ± 0.37</strong></td>
</tr>
</tbody>
</table>
Experiments: deep training tasks

- 8 workers, 1% compression ratio (top-k compressors), minibatch=128, R=2, ResNet18/ResNet20
Experiments: image classification on Cifar-10

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

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

<table>
<thead>
<tr>
<th>METHODS</th>
<th>ResNet18</th>
<th>ResNet20</th>
</tr>
</thead>
<tbody>
<tr>
<td>PSGD</td>
<td>93.99 ± 0.52</td>
<td>91.62 ± 0.13</td>
</tr>
<tr>
<td>QSGD</td>
<td>92.86 ± 0.34</td>
<td>90.24 ± 0.22</td>
</tr>
<tr>
<td>MEM-SGD</td>
<td>94.47 ± 0.27</td>
<td>91.36 ± 0.07</td>
</tr>
<tr>
<td>Double-Squeeze</td>
<td>93.35 ± 0.39</td>
<td>90.89 ± 0.14</td>
</tr>
<tr>
<td>Neolithic</td>
<td><strong>93.87 ± 0.46</strong></td>
<td><strong>91.25 ± 0.14</strong></td>
</tr>
</tbody>
</table>
Experiments: influence of hyper parameter R

- We empirically investigate the influence of $R$ for the performance of NEOLTHIC

<table>
<thead>
<tr>
<th>Rounds</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEOLTHIC (5%)</td>
<td>94.63 ± 0.09</td>
<td>93.32 ± 0.08</td>
<td>92.55 ± 0.12</td>
<td>91.48 ± 0.18</td>
</tr>
<tr>
<td>NEOLTHIC (1%)</td>
<td>94.16 ± 0.10</td>
<td>93.15 ± 0.11</td>
<td>92.27 ± 0.08</td>
<td>91.32 ± 0.12</td>
</tr>
</tbody>
</table>

- Conjecture: large-batch gradient accumulation helps optimization but may hurt generalization

- Advice: using NEOLTHIC in scenarios that are friendly to large-batch training
Conclusion

• Compression can save communication overhead in distributed learning

• We established the lower bounds for alg. with uni/bidirectional and unbiased/contractive compression

• We developed NEOLITHIC to nearly attain these optimal rates

• To further improve the algorithmic performance, we have to explore new compressor properties rather than consider how to apply unbiased or contractive compressors more cleverly to algorithms.
Thank you!