Communication-Efficient Distributed SGD with Compressed Sensing

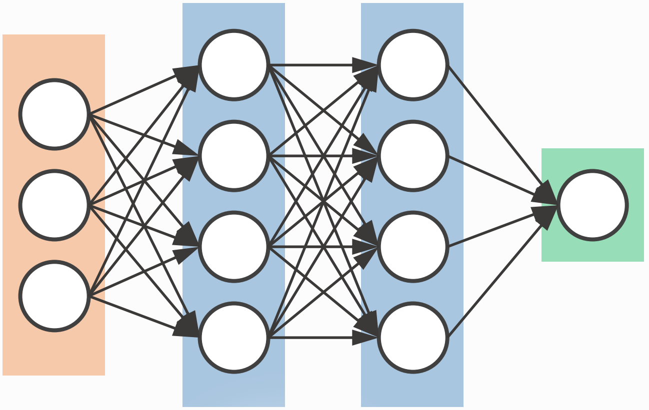
Yujie Tang · Vikram Ramanathan
Junshan Zhang · Na Li

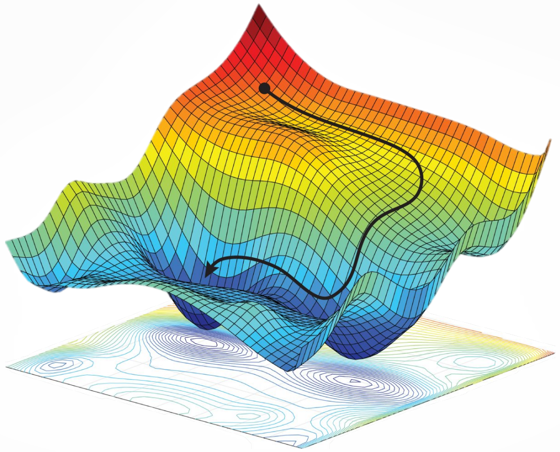
Harvard John A. Paulson School of Engineering and Applied Sciences

Ira A. Fulton Schools of Engineering
Arizona State University

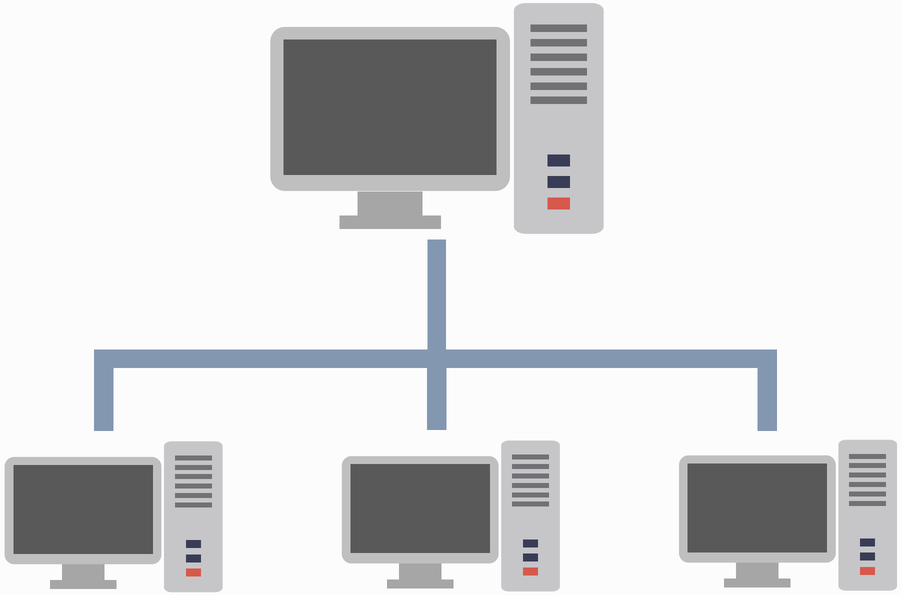
- Motivation & Problem Setup
- Literature Review
- Algorithm Design & Convergence Guarantees
- Numerical Experiments
- Summary & Future Directions

- Motivation & Problem Setup
- Literature Review
- Algorithm Design & Convergence Guarantees
- Numerical Experiments
- Summary & Future Directions

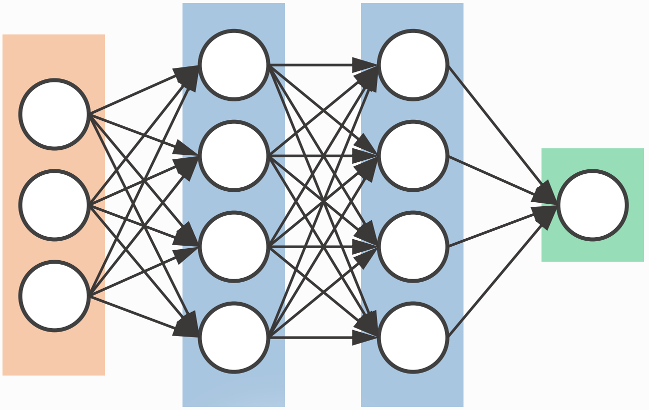


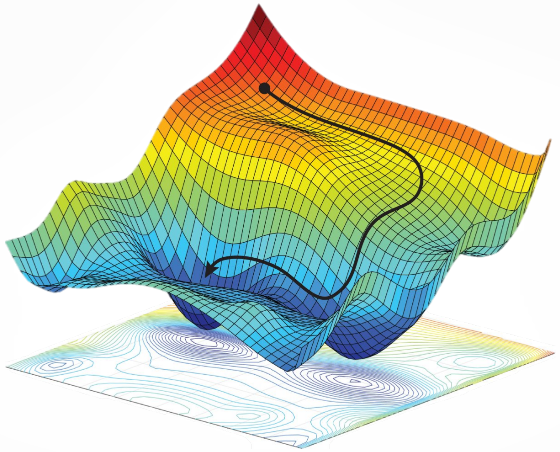


Credit: N. Azizan and B. Hassibi

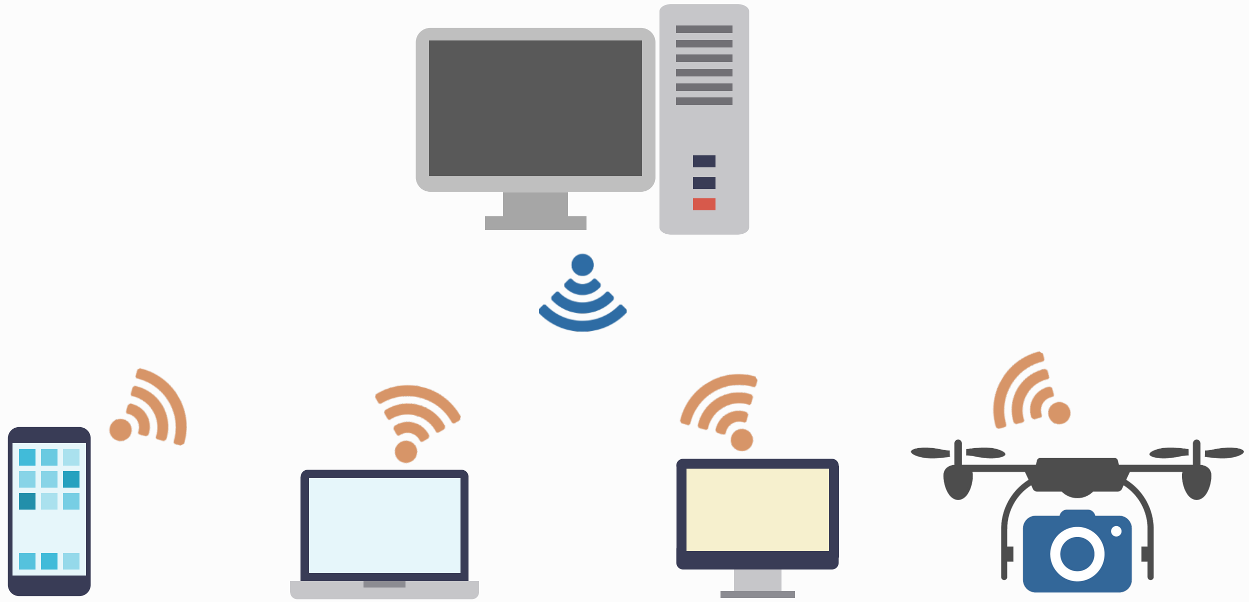


- Large models
- Massive datasets



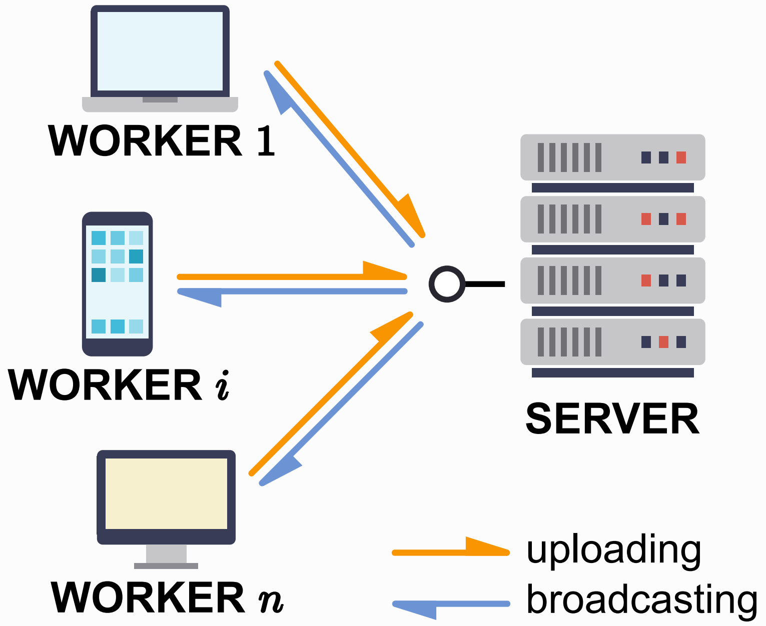


Credit: N. Azizan and B. Hassibi



- Edge devices capable of data collection and processing for machine learning task
- Preferrable to keep data locally
- Wireless channels
  Lossy, unreliable and have limited bandwidth

Problem Setup



Each worker

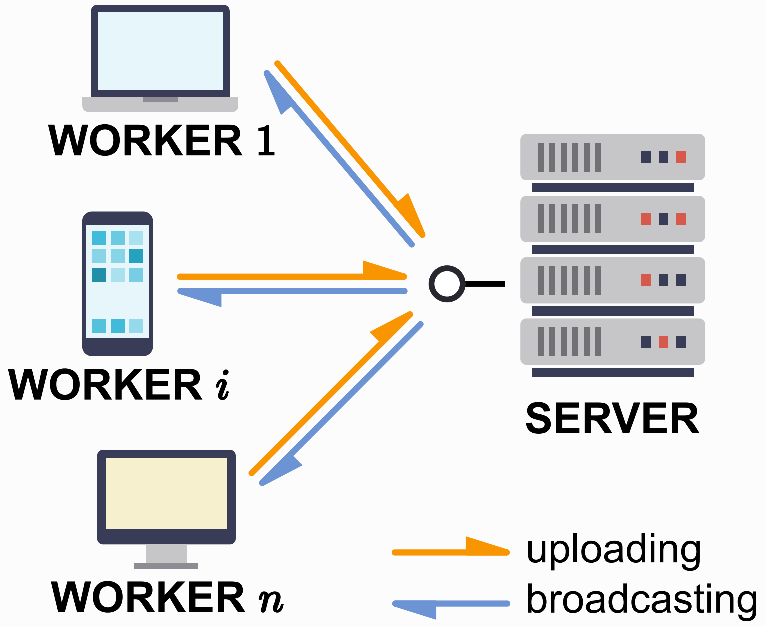
- ❑ local objective $f_i(x), \; x \in \mathbb{R}^d$
- ❑ stochastic gradient $\mathrm{g}_i(x)$
  - unbiased: $\mathbb{E}[\mathrm{g}_i(x)] = \nabla f_i(x)$

Communication links

- ❑ broadcasting
- ❑ uploading

The server $\underset{x \in \mathbb{R}^d}{\text{minimize}} \; \frac{1}{n}\sum_{i=1}^{n} f_i(x)$

A Common Approach



Server:

Randomly choose m workers

Broadcast $x(t)$

Each chosen worker:

Query stochastic gradient $g_i(t) = \mathrm{g}_i(x(t))$

Upload $g_i(t) \in \mathbb{R}^d$

Server:

Aggregate $g(t) = \frac{1}{m} \sum_i g_i(t)$

Update $x(t+1) = x(t) - \eta g(t)$

A Common Approach

- Collecting local gradients can be **costly** when d **is large**
- Reducing m does **not** help: Smaller m requires more iterations.

Server:

Randomly choose m workers
Broadcast $x(t)$

Each chosen worker:

Query stochastic gradient $g_i(t) = \mathrm{g}_i(x(t))$
Upload $g_i(t) \in \mathbb{R}^d$

Server:

Aggregate $g(t) = \frac{1}{m}\sum_i g_i(t)$
Update $x(t{+}1) = x(t) - \eta g(t)$

- Motivation & Problem Setup
- **Literature Review**
- Algorithm Design & Convergence Guarantees
- Numerical Experiments
- Summary & Future Directions

Communication-Efficient SGD

Local SGD/FedAvg

Gradient Compression

Communication-Efficient SGD

Local SGD/FedAvg Gradient Compression

Server:

Randomly choose m workers

Broadcast $x(t)$

Each chosen worker:

Initialize $x_i(0;t) = x(t)$

Run multiple SGD iterations $x_i(\tau+1;t) = x_i(\tau;t) - \eta\, \mathrm{g}_i(x_i(\tau;t))$

Upload $x_i(T;t)$

Server:

Aggregate $x(t+1) = \frac{1}{m}\sum_i x_i(T;t)$

Communication-Efficient SGD

Local SGD/FedAvg | Gradient Compression

- Application in federated learning [McMahan 2017]
- Convergence for **i.i.d. case** (identical local objectives/stochastic gradients)
  [Stich 2018a] [Wang 2018] [Yu 2019]
- Convergence for **non-i.i.d. case (heterogeneous** objectives/stochastic gradients)
  [Li 2018] [Khaled 2019] [Li 2019] [Wang 2020]
  - Requires **bounded dissimilarities** of local objectives/gradients

Communication-Efficient SGD

Local SGD/FedAvg

Gradient Compression

- **Quantization**
  [Seide 2014] [Alistarh 2017] [Bernstein 2018]
- **Sparsification**
  [Alistarh 2018] [Wangni 2018]
- **Error feedback**
  [Stich 2018b] [Karimireddy 2019]
  ✓ Can handle bias
  ✓ Comparable convergence rate with vanilla SGD

Server:

Randomly choose m workers

Broadcast $x(t)$

Each chosen worker:

Query stochastic gradient $g_i(t) = \mathrm{g}_i(x(t))$

Compress $y_i(t) = \mathcal{C}(g_i(t))$

Upload $y_i(t)$

Server:

Decompress and aggregate

$\hat{g}(t) = \mathcal{U}(\{y_i(t)\})$

Update $x(t+1) = x(t) - \eta\hat{g}(t)$

Communication-Efficient SGD

Local SGD/FedAvg | **Gradient Compression**

- Quantization & sparsification are nonlinear
- First decompress, then aggregate
- Harder to control the error $\|\hat{g}(t) - \frac{1}{m}\sum_i g_i(t)\|$
- Error-feedback requires **full participation** of workers for each iteration.

Server:

Randomly choose m workers

Broadcast $x(t)$

Each chosen worker:

Query stochastic gradient $g_i(t) = \mathrm{g}_i(x(t))$

Compress $y_i(t) = \mathcal{C}(g_i(t))$

Upload $y_i(t)$

Server:

Decompress and aggregate

$$\hat{g}(t) = \frac{1}{m}\sum_i \mathcal{U}(y_i(t))$$

Update $x(t+1) = x(t) - \eta \hat{g}(t)$

Communication-Efficient SGD

Local SGD/FedAvg | **Gradient Compression**

- **Count Sketch**
  [Ivkin 2019] [Rothchild 2020]
  - $\mathcal{C}$ is a linear operator
  - $\mathcal{U}$ recovers the top-K entries of $\frac{1}{m}\sum_i g_i(t)$
  - Incorporates error feedback
  - Replies on **approximate sparsity** of (error-corrected) aggregated SG

Server:

Randomly choose m workers

Broadcast $x(t)$

Each chosen worker:

Query stochastic gradient $g_i(t) = \mathrm{g}_i(x(t))$

Compress $y_i(t) = \mathcal{C}(g_i(t))$

Upload $y_i(t)$

Server:

Aggregate and decompress

$$\hat{g}(t) = \mathcal{U}\left(\frac{1}{m}\sum_i y_i(t)\right)$$

Update $x(t+1) = x(t) - \eta\hat{g}(t)$

Communication-Efficient SGD

Local SGD/FedAvg

Gradient Compression

- **Count Sketch**
  [Ivkin 2019] [Rothchild 2020]

❖ First aggregate, then decompress

❖ Error feedback carried out by the server

❖ Allows partial participation of workers

❖ **Inconsistency** in its theoretical foundation

Server:

Randomly choose m workers

Broadcast $x(t)$

Each chosen worker:

Query stochastic gradient $g_i(t) = \mathrm{g}_i(x(t))$

Compress $y_i(t) = \mathcal{C}(g_i(t))$

Upload $y_i(t)$

Server:

Aggregate and decompress

$$\hat{g}(t) = \mathcal{U}\left(\frac{1}{m}\sum_i y_i(t)\right)$$

Update $x(t+1) = x(t) - \eta\hat{g}(t)$

- Motivation & Problem Setup
- Literature Review
- **Algorithm Design & Convergence Guarantees**
- Numerical Experiments
- Summary & Future Directions

Preliminaries on Compressed Sensing

- Undetermined noisy linear measurement

$$y = \Phi x$$

measurement
Q entries

$Q \times d$ sensing matrix
$Q < d$

approximately sparse
d entries

- How to design
  - sensing matrix Φ
  - reconstruction algorithm

  to recover the original signal x from y and Φ ?

- Two schemes: **for-each** and **for-all**

Two Schemes of Compressed Sensing

Preliminaries

Algorithm Design

Convergence

For-each scheme

- Construct a probability distribution $\mathcal{D}$ over $Q \times d$ sensing matrices
- Sample a new $\Phi \sim \mathcal{D}$ every time a new signal x is to be measured and reconstructed
- Theoretical guarantees of reconstruction algorithms:

> Given Q and d, suppose $K \leq O(Q/\log d)$. Then there exist $\epsilon > 0$ and $\alpha > 0$ depending on K, Q and d, such that for any $x \in \mathbb{R}^d$ that is **deterministic/independent of Φ**,
>
> $$\mathbb{P}_{\Phi\sim\mathcal{D}}(\|\mathcal{A}(y;\Phi) - x\|_2 \leq (1+\epsilon)\|x - x^{[K]}\|_2) \geq 1 - O(d^{-\alpha})$$
>
> reconstructed signal; best K-sparse approximation of x

Two Schemes of Compressed Sensing

For-each scheme

- Construct a probability distribution $\mathcal{D}$ over $Q \times d$ sensing matrices
- Sample a new $\Phi \sim \mathcal{D}$ every time a new signal x is to be measured and reconstructed
- Theoretical guarantees of reconstruction algorithms:

> Given Q and d, suppose $K \leq O(Q/\log d)$. Then there exist $\epsilon > 0$ and $\alpha > 0$ depending on K, Q and d, such that for any $x \in \mathbb{R}^d$ that is **deterministic/independent of** Φ,
>
> $$\mathbb{P}_{\Phi\sim\mathcal{D}}\Big(\underbrace{\|\mathcal{A}(y;\Phi) - x\|_2}_{\text{reconstruction error}} \leq (1+\epsilon)\underbrace{\|x - x^{[K]}\|_2}_{\text{best } K\text{-sparse approximation error}}\Big) \geq \underbrace{1 - O(d^{-\alpha})}_{\text{w.h.p.}}$$

- Examples: Count Sketch [Charikar 2002], Count-min Sketch [Cormode 2005]

Two Schemes of Compressed Sensing

For-all scheme

- Construct a single $\Phi \in \mathbb{R}^{Q\times d}$ that satisfies **restricted isometry property**
- Use this sensing matrix for measuring and reconstructing **all** possible x

A matrix $\Phi \in \mathbb{R}^{Q\times d}$ is said to satisfy (K, δ_K)-**restricted isometry property (RIP)** for some $K < d$ and $\delta_K \in (0,1)$, if

$$(1-\delta_K)\|x\|_2^2 \le \|\Phi x\|_2^2 \le (1+\delta_K)\|x\|_2^2$$

for any x that has at most K nonzero entries.

$(2K, \delta_{2K})$-RIP → $\|\Phi u - \Phi v\|_2 \ge \sqrt{1-\delta_{2K}}\|u-v\|_2$ for any u, v that have at most K nonzero entries.

↓

linear measurement $x \mapsto \Phi x$ can discriminate sparse signals

Two Schemes of Compressed Sensing

For-all scheme

- Construct a single $\Phi \in \mathbb{R}^{Q \times d}$ that satisfies **restricted isometry property**
- Use this sensing matrix for measuring and reconstructing **all** possible x

> A matrix $\Phi \in \mathbb{R}^{Q \times d}$ is said to satisfy (K, δ_K)-**restricted isometry property (RIP)** for some $K < d$ and $\delta_K \in (0, 1)$, if
>
> $$(1 - \delta_K)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta_K)\|x\|_2^2$$
>
> for any x that has at most K nonzero entries.

- How to generate RIP matrices?
  - ✓ Randomized methods (will be explained later)

Two Schemes of Compressed Sensing

Preliminaries

Algorithm Design

Convergence

For-all scheme

- Construct a single $\Phi \in \mathbb{R}^{Q \times d}$ that satisfies **restricted isometry property**
- Use this sensing matrix for measuring and reconstructing **all** possible x

> A matrix $\Phi \in \mathbb{R}^{Q \times d}$ is said to satisfy (K, δ_K)-**restricted isometry property (RIP)** for some $K < d$ and $\delta_K \in (0, 1)$, if
>
> $$(1 - \delta_K)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta_K)\|x\|_2^2$$
>
> for any x that has at most K nonzero entries.

- Examples: ℓ_1 minimization [Candès 2005], CoSaMP [Needell 2009], Fast Iterative Hard Thresholding [Wei 2014]

Approximately solve $\min_{z \in \mathbb{R}^d} \quad \|z\|_0 \quad \text{s.t.} \quad y = \Phi z$

or $\min_{z \in \mathbb{R}^d} \quad \frac{1}{2}\|y - \Phi z\|_2^2 \quad \text{s.t.} \quad \|z\|_0 \leq K$

Two Schemes of Compressed Sensing

Preliminaries

Algorithm Design

Convergence

For-all scheme

- Construct a single $\Phi \in \mathbb{R}^{Q \times d}$ that satisfies **restricted isometry property**
- Use this sensing matrix for measuring and reconstructing **all** possible x

> A matrix $\Phi \in \mathbb{R}^{Q \times d}$ is said to satisfy (K, δ_K)-**restricted isometry property (RIP)** for some $K < d$ and $\delta_K \in (0, 1)$, if
>
> $$(1 - \delta_K)\|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta_K)\|x\|_2^2$$
>
> for any x that has at most K nonzero entries.

- Examples: ℓ_1 minimization [Candès 2005], CoSaMP [Needell 2009], Fast Iterative Hard Thresholding [Wei 2014]

✓ Theoretical guarantees on reconstruction error when Φ satisfies RIP.

Metric of Sparsity

➢ How to quantify the **sparsity** of a signal x?

- ℓ_0 norm: $\|x\|_0 :=$ number of nonzero entries of x
  - Not continuous, not robust to small perturbations
  - Cannot characterize **approximate sparsity**

- An alternative metric [Lopes 2016]:

$$\mathrm{sp}(x) := \frac{\|x\|_1^2}{\|x\|_2^2 \cdot d} \in (0,1)$$

  - Continuous, robust to small perturbations
  - **Schur concave**: If $\|u\|_1 = \|v\|_1$ and $\|u - u^{[K]}\|_1 \leq \|v - v^{[K]}\|_1$ for all $K = 1, \ldots, d$, then $\mathrm{sp}(u) \leq \mathrm{sp}(v)$
  - Can characterize **approximate sparsity**

Preliminaries on Compressed Sensing

For-each scheme

- Construct a probability distribution $\mathcal{D}$ over $Q \times d$ sensing matrices
- Sample a new $\Phi \sim \mathcal{D}$ every time a new signal x is to be measured and reconstructed

For-all scheme

- Construct a single $\Phi \in \mathbb{R}^{Q \times d}$ that satisfies **restricted isometry property**
- Use this sensing matrix for measuring and reconstructing **all** possible x

Sparsity metric

$$\mathrm{sp}(x) := \frac{\|x\|_1^2}{\|x\|_2^2 \cdot d}$$

- Continuous & **Schur concave**
- Can characterize **approximate sparsity**

Algorithm Design

Server:

- Randomly choose m workers
- Broadcast $x(t)$

Each chosen worker:

- Query SG $g_i(t) = \mathrm{g}_i(x(t))$
- Compress $y_i(t) = \mathcal{C}(g_i(t))$
- Upload $y_i(t)$

Server:

- Aggregate and decompress

  $\hat{g}(t) = \mathcal{U}(\{y_i(t)\})$
- Update $x(t+1) = x(t) - \eta \hat{g}(t)$

Algorithm Design

Server generates $\Phi \in \mathbb{R}^{Q \times d}$ and broadcasts it to all workers

Server:

- Randomly choose m workers
- Broadcast $x(t)$

Each chosen worker:

- Query SG $g_i(t) = \mathrm{g}_i(x(t))$
- Compress $y_i(t) = \Phi g_i(t)$
- Upload $y_i(t)$

Server:

- Aggregate and decompress

$$\hat{g}(t) = \mathcal{A}\left(\frac{1}{m}\sum_i y_i(t); \Phi\right)$$

- Update $x(t+1) = x(t) - \eta \hat{g}(t)$

- $\mathcal{A}$: reconstruction algorithm
- Why can we average before reconstruction?
  - ✓ Compression is **linear**
  - ✓ $\hat{g}(t) \approx \frac{1}{m}\sum_i g_i(t)$

Algorithm Design

Server generates $\Phi \in \mathbb{R}^{Q\times d}$ and broadcasts it to all workers

Server:

Randomly choose m workers
Broadcast $x(t)$

Each chosen worker:

Query SG $g_i(t) = \mathrm{g}_i(x(t))$
Compress $y_i(t) = \Phi g_i(t)$
Upload $y_i(t)$

Server:

Aggregate and decompress

$$\hat{g}(t) = \mathcal{A}\left(\frac{1}{m}\sum_i y_i(t); \Phi\right)$$

Update $x(t+1) = x(t) - \eta\hat{g}(t)$

- A single Φ for all iterations
  ✓ **For-all** scheme

❖ **Inconsistency** in the work [Rothchild 2020]:

A **single** Φ for compression and reconstruction in **all** iterations

↕

Count Sketch for generation of Φ and reconstruction $\mathcal{A}$ (**for-each** scheme)

Algorithm Design

Preliminaries

Algorithm Design

Convergence

Server generates $\Phi \in \mathbb{R}^{Q\times d}$ and broadcasts it to all workers

Server:

Randomly choose m workers

Broadcast $x(t)$

Each chosen worker:

Query SG $g_i(t) = \mathrm{g}_i(x(t))$

Compress $y_i(t) = \Phi g_i(t)$

Upload $y_i(t)$

Server:

Aggregate and decompress

$$\hat{g}(t) = \mathcal{A}\left(\frac{1}{m}\sum_i y_i(t); \Phi\right)$$

Update $x(t+1) = x(t) - \eta\hat{g}(t)$

- A single Φ for all iterations

  ✓ **For-all** scheme

❖ **Our algorithm**

Φ : Subsampled Fourier matrix

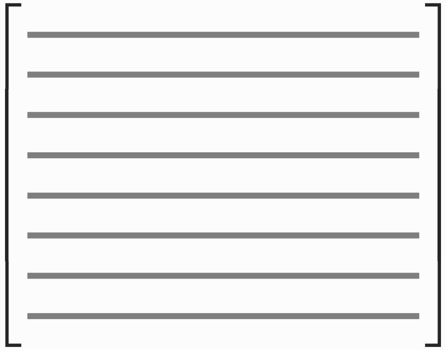
$\mathcal{A}$: Fast Iterative Hard Thresholding (FIHT)

Algorithm Design: Sensing Matrix

Φ: Subsampled Fourier matrix

$\mathcal{A}$: Fast Iterative Hard Thresholding (FIHT)

1. Let B be the $d \times d$ discrete cosine transform (**DCT**) matrix or Walsh-Hadamard transform (**WHT**) matrix
   - B is orthogonal
   - $|B_{ij}| \leq O(1/\sqrt{d})$
   - Bu and $B^\top v$ for any u and v can be computed by $O(d \log d)$ algorithms

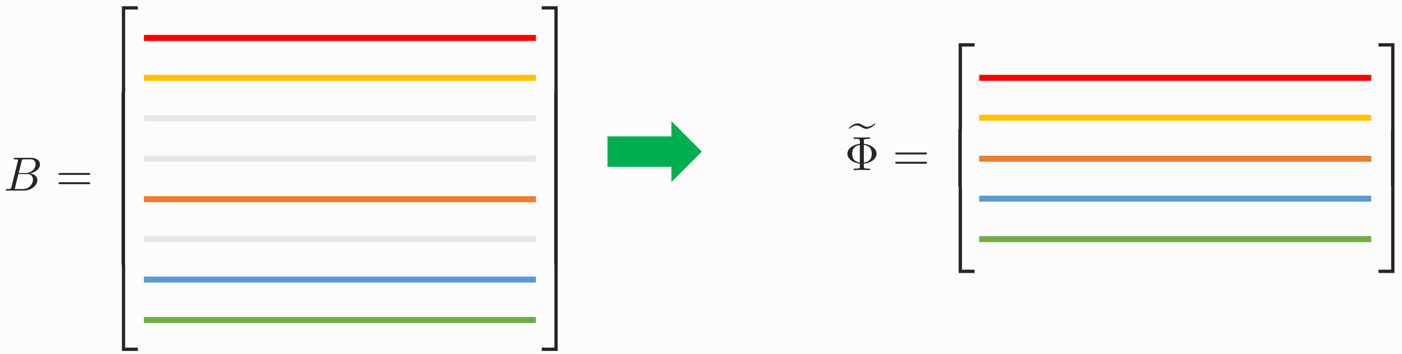
$B =$ 

Algorithm Design: Sensing Matrix

Φ: Subsampled Fourier matrix

$\mathcal{A}$: Fast Iterative Hard Thresholding (FIHT)

1. Let B be the $d \times d$ discrete cosine transform (**DCT**) matrix or Walsh-Hadamard transform (**WHT**) matrix
2. Randomly choose Q rows of B to form a $Q \times d$ submatrix $\widetilde{\Phi}$

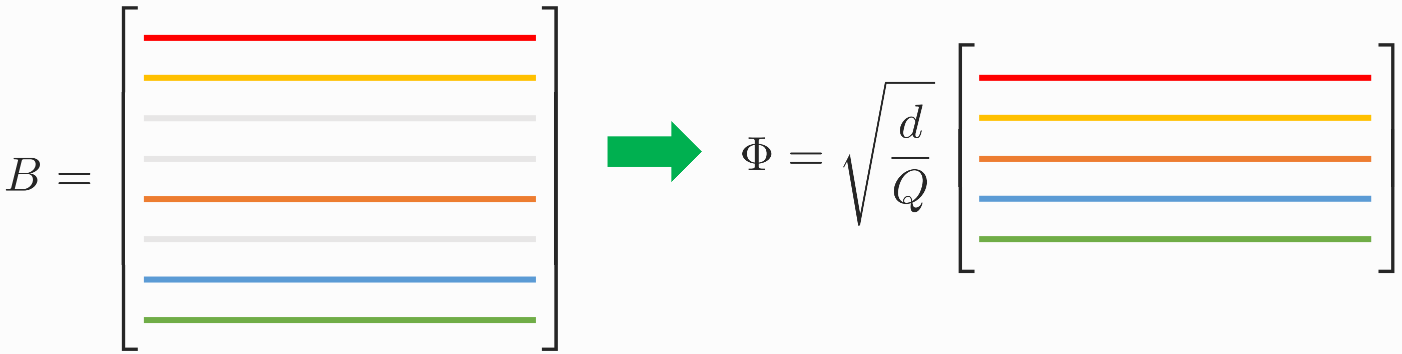


Algorithm Design: Sensing Matrix

Φ: Subsampled Fourier matrix

$\mathcal{A}$: Fast Iterative Hard Thresholding (FIHT)

1. Let B be the $d \times d$ discrete cosine transform (**DCT**) matrix or Walsh-Hadamard transform (**WHT**) matrix
2. Randomly choose Q rows of B to form a $Q \times d$ submatrix $\widetilde{\Phi}$
3. Normalize by $\Phi = \sqrt{\frac{d}{Q}} \cdot \widetilde{\Phi}$



Algorithm Design: Sensing Matrix

Φ: Subsampled Fourier matrix

$\mathcal{A}$: Fast Iterative Hard Thresholding (FIHT)

1. Let B be the $d \times d$ discrete cosine transform (**DCT**) matrix or Walsh-Hadamard transform (**WHT**) matrix
2. Randomly choose Q rows of B to form a $Q \times d$ submatrix $\widetilde{\Phi}$
3. Normalize by $\Phi = \sqrt{\frac{d}{Q}} \cdot \widetilde{\Phi}$

Theorem. [Haviv 2017] Φ satisfies (K, δ_K)-RIP with high probability when $Q \geq \tilde{O}(K \log^2 K \log d \cdot \delta_K^{-2})$

✓ Broadcasting Φ is easy: Just send the row indices of B

✓ Matrix-vector multiplications Φu and $\Phi^\top v$ are fast

Φ: Subsampled Fourier matrix

$\mathcal{A}$: Fast Iterative Hard Thresholding (FIHT)

Fast Iterative Hard Thresholding (FIHT) [Wei 2014]

- Greedy algorithm that approximately solves

$$\min_{z \in \mathbb{R}^d} \quad \frac{1}{2}\|y - \Phi z\|_2^2 \quad \text{s.t.} \quad \|z\|_0 \leq K$$

- Returns a sparse vector with at most K nonzero entries (K tunable)
- Theoretical guarantees on the reconstruction error if Φ satisfies $(4K, \delta_{4K})$-RIP.
- Empirically, it achieves a good balance between reconstruction error and computation time.

Algorithm Design

Server generates $\Phi \in \mathbb{R}^{Q \times d}$ and broadcasts it to all workers

Server:

- Randomly choose m workers
- Broadcast $x(t)$

Each chosen worker:

- Query SG $g_i(t) = \mathrm{g}_i(x(t))$
- Compress $y_i(t) = \Phi g_i(t)$
- Upload $y_i(t)$

Server:

- Aggregate and decompress

$$\hat{g}(t) = \mathcal{A}\left(\frac{1}{m}\sum_i y_i(t); \Phi\right)$$

- Update $x(t+1) = x(t) - \eta \hat{g}(t)$

- A single Φ for all iterations
  - ✓ **For-all** scheme

❖ **Our algorithm**

Φ : Subsampled Fourier matrix

$\mathcal{A}$: Fast Iterative Hard Thresholding (FIHT)

- Reconstruction by $\mathcal{A}$ is **biased**
  - ✓ Incorporate **error-feedback**

Algorithm Design: Error-Feedback

Preliminaries

Algorithm Design

Convergence

Error-feedback [Stich 2018b] [Karimireddy 2019]

$$g(t) = \mathrm{g}(x(t))$$
$$\hat{g}(t) = \mathcal{A}(\Phi g(t); \Phi)$$
$$x(t+1) = x(t) - \eta \hat{g}(t)$$

$$g(t) = \mathrm{g}(x(t))$$
$$p(t) = \eta g(t) + e(t)$$ ➢ error feedback
$$\Delta(t) = \mathcal{A}(\Phi p(t); \Phi)$$
$$x(t+1) = x(t) - \Delta(t)$$
$$e(t+1) = p(t) - \Delta(t)$$ ➢ error update

Suppose there exists $\gamma < 1$ such that $\|\Delta(t) - p(t)\|_2 \le \gamma \|p(t)\|_2$ for all t. Then SGD with error-feedback converges with rate

$$\frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\|\nabla f(x(t))\|_2^2] \le \frac{C_1}{\sqrt{T}} + \frac{C_2(\gamma)}{T}$$

where C_1 does not depend on γ.

✓ Leading term is **not** affected by compression

Algorithm Outline

Preliminaries

Algorithm Design

Convergence

Server generates $\Phi \in \mathbb{R}^{Q \times d}$ as a **subsampled Fourier matrix** and broadcasts it to all workers

Server:

- Randomly choose m workers
- Broadcast $x(t)$

Each chosen worker:

- Query stochastic gradient $g_i(t) = \mathrm{g}_i(x(t))$
- Compress $y_i(t) = \Phi g_i(t)$
- Upload $y_i(t) \in \mathbb{R}^Q$

Server:

- $y(t) = \frac{1}{m} \sum_i y_i(t)$ ➤ aggregation
- $z(t) = \eta\, y(t) + \varepsilon(t)$ ➤ error feedback
- $\Delta(t) = \mathcal{A}(z(t); \Phi)$ ➤ reconstruction by **FIHT**
- $x(t+1) = x(t) - \Delta(t)$ ➤ SGD update
- $\varepsilon(t+1) = z(t) - \Phi\Delta(t)$ ➤ error update

Convergence Guarantees

Preliminaries

Algorithm Design

Convergence

T : # of iterations η: step size K : # of nonzero entries in the output of FIHT

$p(t)$: error-corrected aggregated SG $\eta \cdot \frac{1}{m} \sum_i g_i(t) + e(t)$

Suppose that Φ satisfies $(4K, \delta_{4K})$-RIP for sufficiently small δ_{4K}, and that

$$\mathrm{sp}(p(t)) \leq O\left(\frac{K}{d}\right)$$

for all t. Then for sufficiently large T, by choosing $\eta = O(1/\sqrt{T})$, we have

$$(f \text{ is smooth}) \qquad \frac{1}{T} \sum_{t=1}^{T} \mathbb{E}[\|\nabla f(x(t))\|_2^2] \leq \frac{C}{\sqrt{T}} + O\left(\frac{1}{T}\right)$$

$$(f \text{ is smooth \& convex}) \qquad f(x(t)) - f^* \leq \frac{C'}{\sqrt{T}} + O\left(\frac{1}{T}\right)$$

Is that the end of the story? ***No***

Convergence Guarantees

Preliminaries

Algorithm Design

Convergence

T: # of iterations η: step size K: # of nonzero entries in the output of FIHT

$p(t)$: error-corrected aggregated SG $\eta \cdot \frac{1}{m}\sum_i g_i(t) + e(t)$

Suppose that Φ satisfies $(4K, \delta_{4K})$-RIP for sufficiently small δ_{4K}, and that

$$\mathrm{sp}(p(t)) \leq O\left(\frac{K}{d}\right)$$

for all t. Then for sufficiently large T, by choosing $\eta = O(1/\sqrt{T})$, we have

$$(f \text{ is smooth}) \qquad \frac{1}{T}\sum_{t=1}^{T} \mathbb{E}[\|\nabla f(x(t))\|_2^2] \leq \frac{C}{\sqrt{T}} + O\left(\frac{1}{T}\right)$$

$$(f \text{ is smooth \& convex}) \qquad f(x(t)) - f^* \leq \frac{C'}{\sqrt{T}} + O\left(\frac{1}{T}\right)$$

Issues with the condition:

- Hard to check
- Rarely holds in practice
- Empirically, $\mathrm{sp}(g(t)) \leq O(K/d)$ seems to be sufficient

- Motivation & Problem Setup
- Literature Review
- Algorithm Design & Convergence Guarantees
- **Numerical Experiments**
- Summary & Future Directions

Numerical Experiments

Federated Learning with CIFAR-10 Dataset

- **Model**: ResNet with $d = 668426$ parameters

- **Setting 1**: i.i.d. local datasets, 100 workers
  - Server queries local gradients from all workers
- **Setting 2**: non-i.i.d. local datasets, 10000 workers
  - Server queries local gradients from 1% of all workers

- We test two algorithms

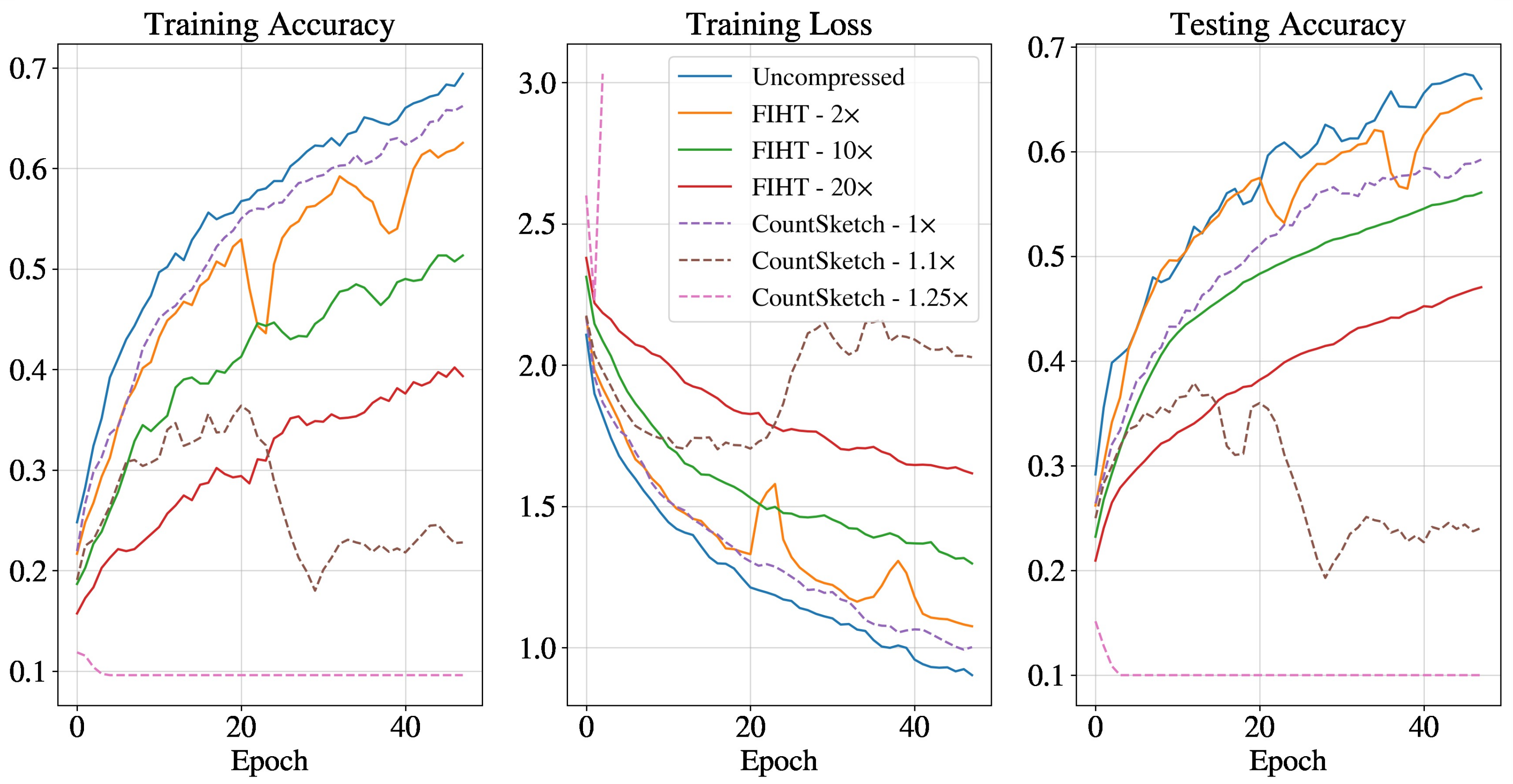
  1. our algorithm, FIHT + error-feedback
  2. Count Sketch + error-feedback
     (the algorithm in [Rothchild 2020] without momentum)

  for different compression rates d/Q

Numerical Experiments

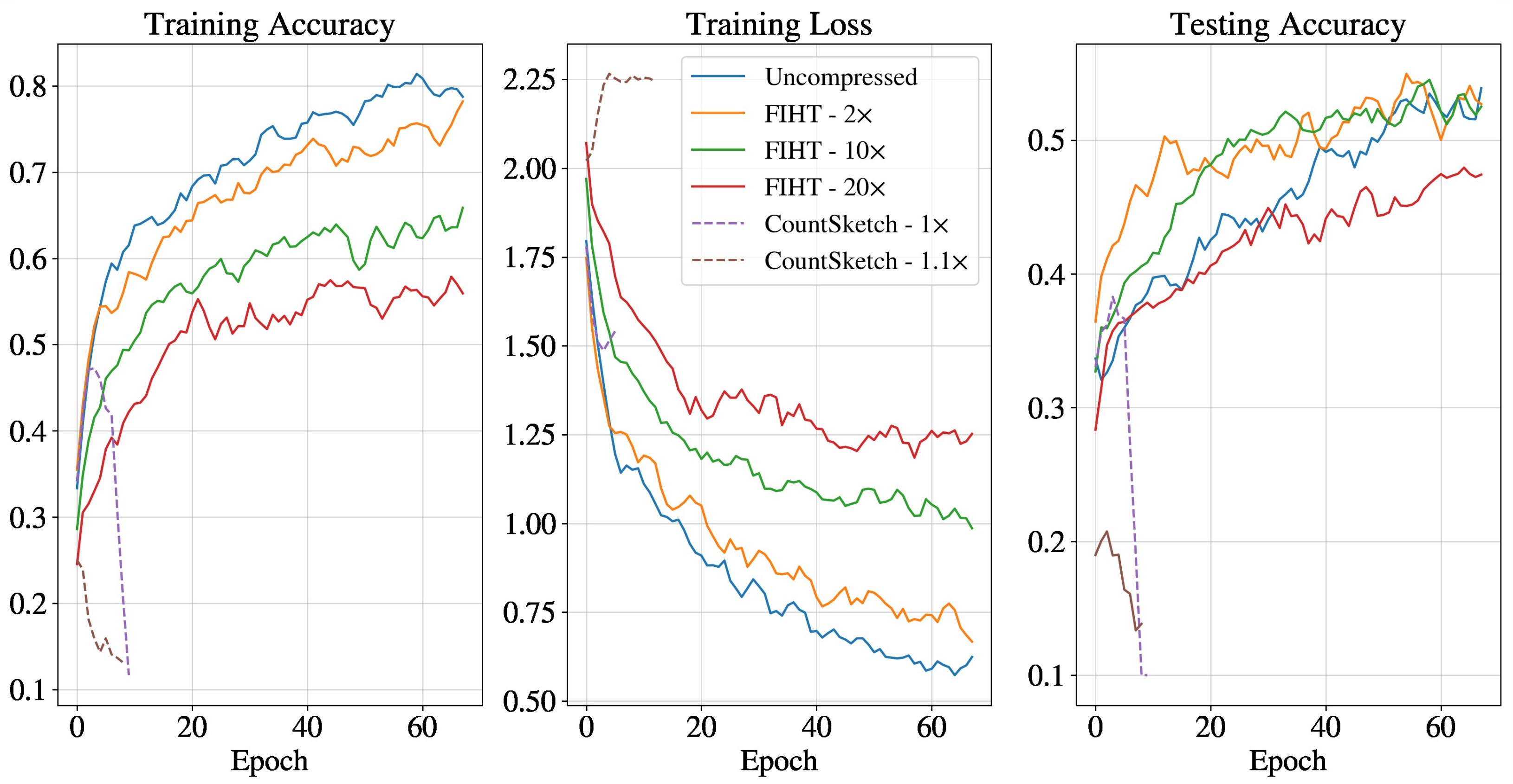
Federated Learning with CIFAR-10 Dataset

- **Setting 1**: i.i.d. local datasets, 100 workers, full participating, $K = 30000$



Federated Learning with CIFAR-10 Dataset

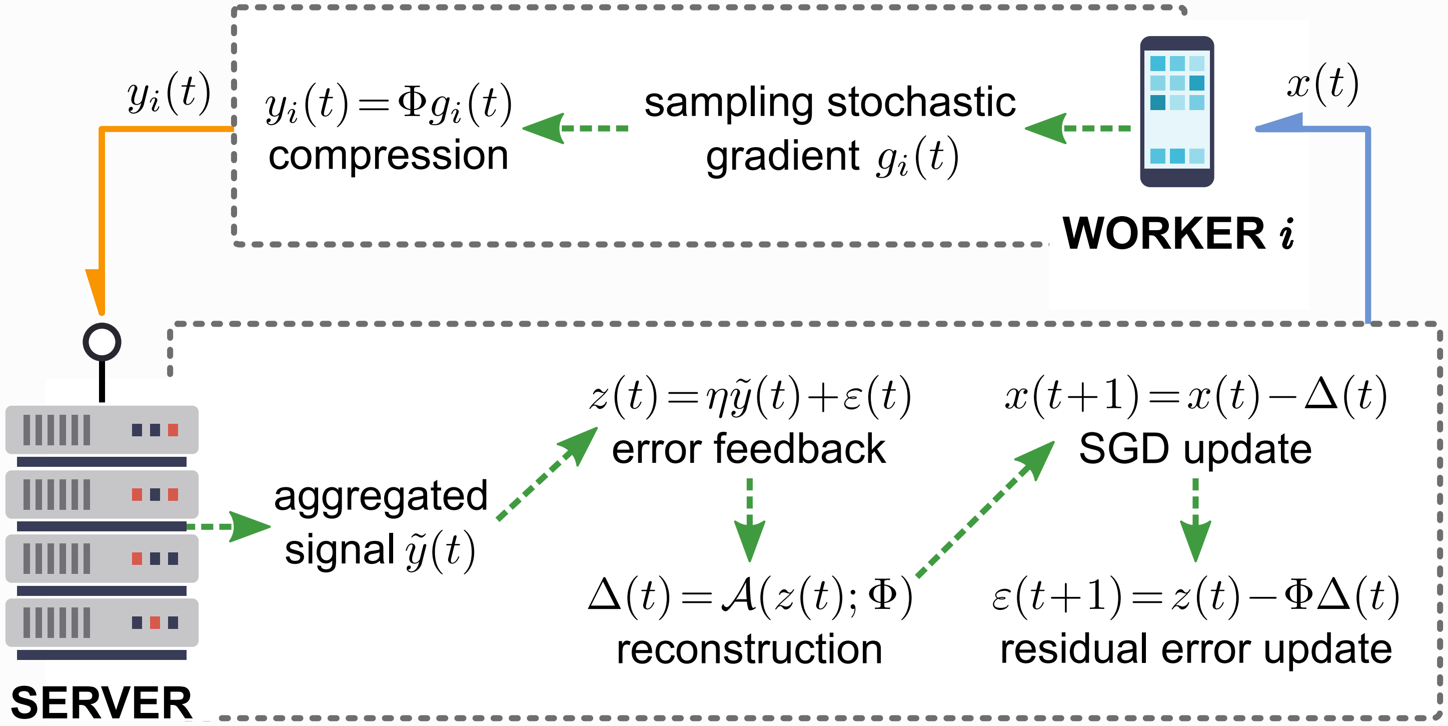
- **Setting 2:** non-i.i.d. local datasets, 10000 workers, 1% participation, $K = 30000$



- Motivation & Problem Setup
- Literature Review
- Algorithm Design & Convergence Guarantees
- Numerical Experiments
- **Summary & Future Directions**

Summary

Compress local SG by sensing matrix



Recover a sparse approximation of the aggregated gradient from the compressed local gradients

- Sensing matrix: **Subsample Fourier matrix**
- Reconstruction algorithm: **FIHT**
- **Error feedback**

Future Directions

- Improving theoretical analysis
- Estimation of sparsity of aggregated gradients
- Extension to decentralized setting
- Extension to gradient-free optimization & reinforcement learning

References

- [McMahan 2017] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pages 1273–1282,2017.
- [Stich 2018a] S. U. Stich. Local SGD converges fast and communicates little. arXiv:1805.09767, 2018.
- [Stich 2018b] S. U. Stich, J.-B. Cordonnier, and M. Jaggi. Sparsified SGD with memory. In *Advances in Neural Information Processing Systems*, volume 31, 2018.
- [Wang 2018] J. Wang and G. Joshi. Cooperative SGD: A unified framework for the design and analysis of communication-efficient SGD algorithms. arXiv:1808.07576, 2018.
- [Yu 2019] H. Yu, S. Yang, and S. Zhu. Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5693–5700, 2019.
- [Li 2018] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith. Federated optimization in heterogeneous networks. arXiv:1812.06127, 2018.
- [Khaled 2019] A. Khaled, K. Mishchenko, and P. Richtárik. First analysis of local GD on heterogeneous data. arXiv:1909.04715, 2019.
- [Li 2019] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang. On the convergence of FedAvg on non-IID data. arXiv:1907.02189, 2019.
- [Wang 2020] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor. Tackling the objective inconsistency problem in heterogeneous federated optimization. In *Advances in Neural Information Processing Systems*, volume 33, pages 7611–7623, 2020.
- [Seide 2014] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu. 1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs. In *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.
- [Alistarh 2017] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic. QSGD: Communication-efficient SGD via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, volume 30, 2017.
- [Alistarh 2018] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli. The convergence of sparsified gradient methods. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018

References

- [Bernstein 2018] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar. signSGD: Compressed optimisation for non-convex problems. In *Proceedings of the 35th International Conference on Machine Learning*, pages 560–569, 2018.
- [Wangni 2018] J. Wangni, J. Wang, J. Liu, and T. Zhang. Gradient sparsification for communication-efficient distributed optimization. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, page 1306–1316, 2018.
- [Karimireddy 2019] S. P. Karimireddy, Q. Rebjock, S. Stich, and M. Jaggi. Error feedback fixes signSGD and other gradient compression schemes. In *International Conference on Machine Learning*, pages 3252–3261, 2019.
- [Ivkin 2019] N. Ivkin, D. Rothchild, E. Ullah, V. braverman, I. Stoica, and R. Arora. Communication-efficient distributed SGD with sketching. In *Advances in Neural Information Processing Systems*, volume 32, 2019.
- [Rothchild 2020] D. Rothchild, A. Panda, E. Ullah, N. Ivkin, I. Stoica, V. Braverman, J. Gonzalez, and R. Arora. FetchSGD: Communication-efficient federated learning with sketching. In *International Conference on Machine Learning*, pages 8253–8265, 2020.
- [Charikar 2002] M. Charikar, K. Chen, and M. Farach-Colton. Finding frequent items in data streams. In *International Colloquium on Automata, Languages, and Programming*, pages 693–703. Springer, 2002.
- [Cormode 2005] G. Cormode and S. Muthukrishnan. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms*, 55(1):58–75, 2005.
- [Candès 2005] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.
- [Needell 2009] D. Needell and J. A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.
- [Wei 2014] K. Wei. Fast iterative hard thresholding for compressed sensing. *IEEE Signal Processing Letters*, 22(5):593–597, 2014.
- [Lopez 2016] M. E. Lopes. Unknown sparsity in compressed sensing: Denoising and inference. *IEEE Transactions on Information Theory*, 62(9):5145–5166, 2016.
- [Haviv 2017] I. Haviv and O. Regev. The restricted isometry property of subsampled Fourier matrices. In *Geometric aspects of functional analysis*, pages 163–179. Springer, 2017.