A Flexible Framework for Communication-Efficient Machine Learning

AAAI'21

Presented by,

Pavana Prakash

Sarit Khirirat[1], Sindri Magnusson[2], Arda Ayetekin[3], Mikael Johansson[1]

[1]KTH Royal Institute of Technology

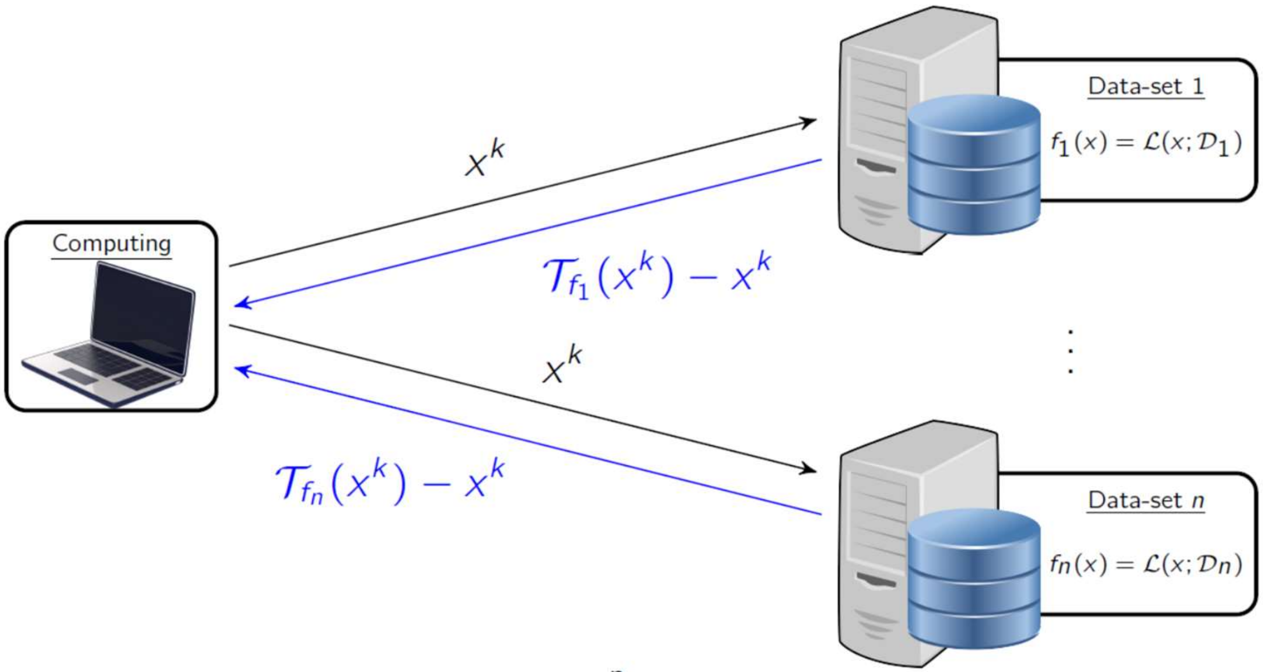
[2]Stockholm University

[3]Ericsson, Sweden.

Outline

- Motivation
- Compression: methods and justification
- Theoretical results
- Numerical experiments and discussion
- Conclusion

Distributed Machine Learning



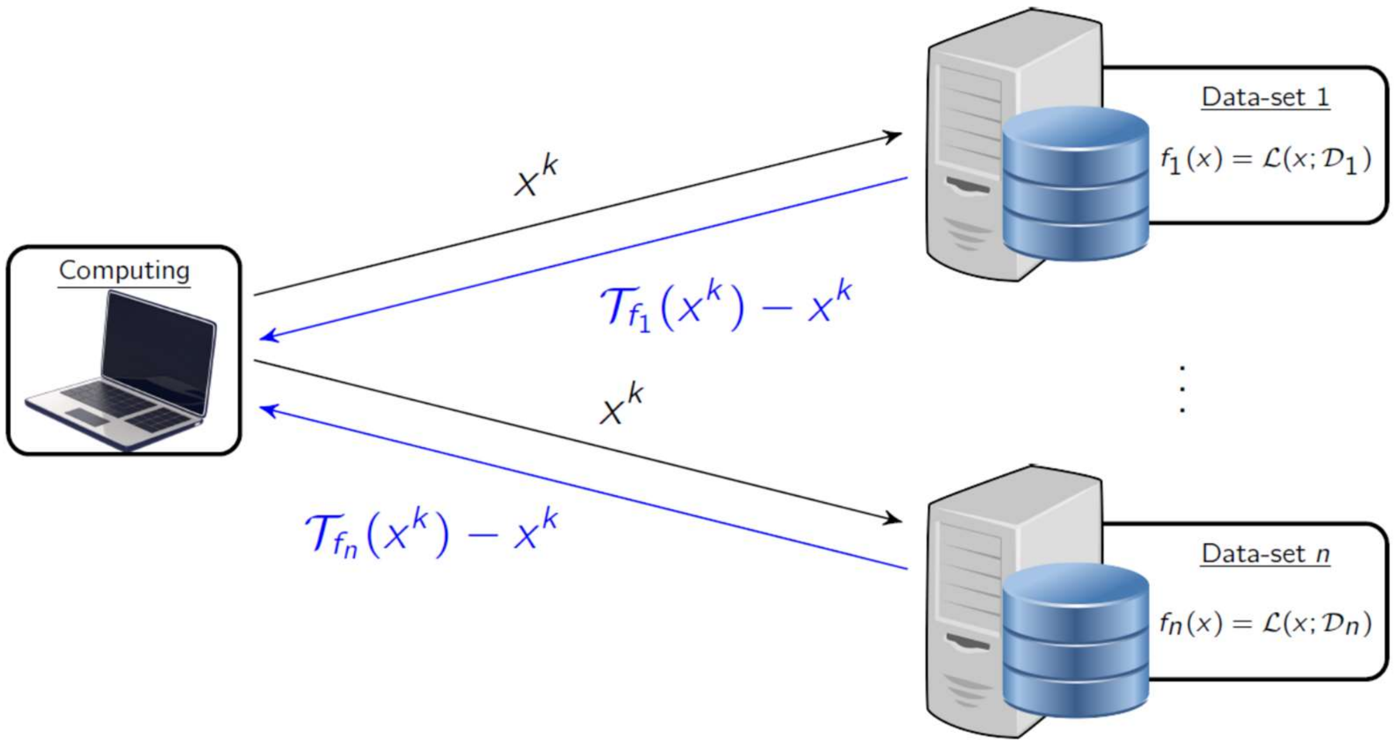
$$x^{k+1} = x^k + \sum_{i=1}^{n} [\mathcal{T}_{f_i}(x^k) - x^k].$$

- State-of-the-art problems have large dimension d (millions of parameters).
- 64 × d communicated bits per iteration per node.
- Performance bottleneck has shifted from computation to communication!
    - Neural network training: communication dominates 80% of run time

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Strategies to Reduce Communication Bottleneck

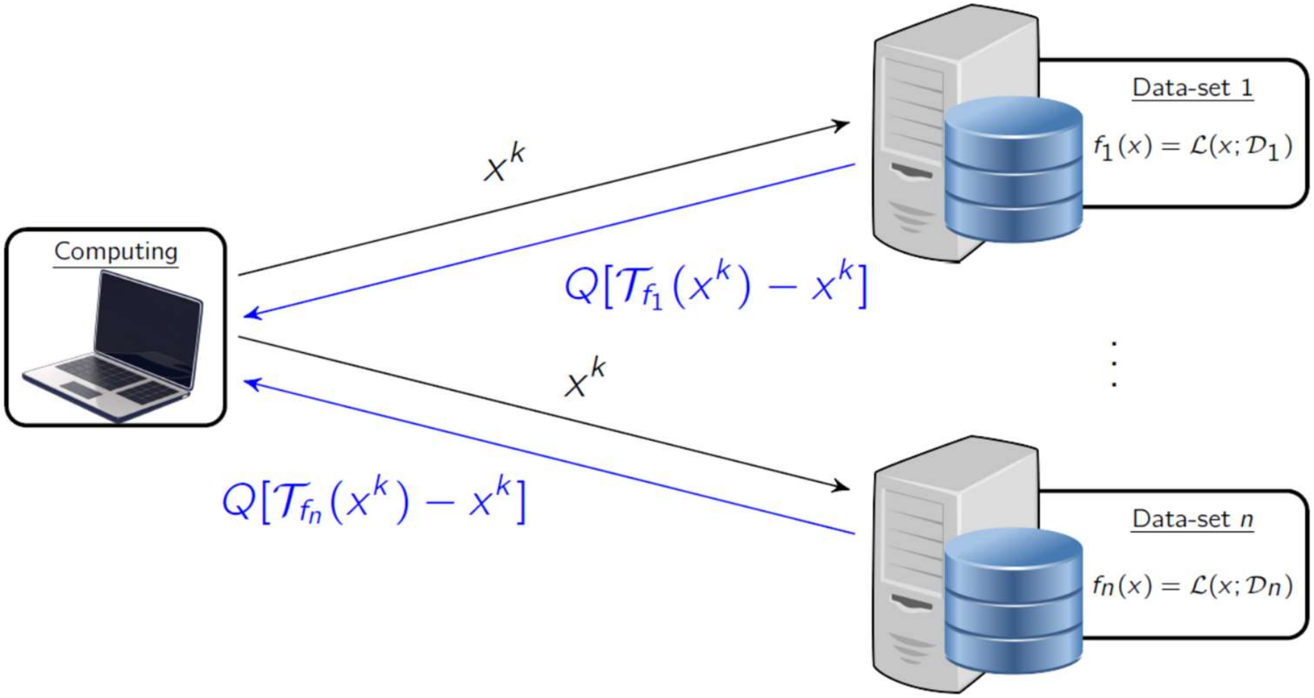


- Asynchronous computation
- Client sampling
- Communication period to update global parameters
- **Compression**

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

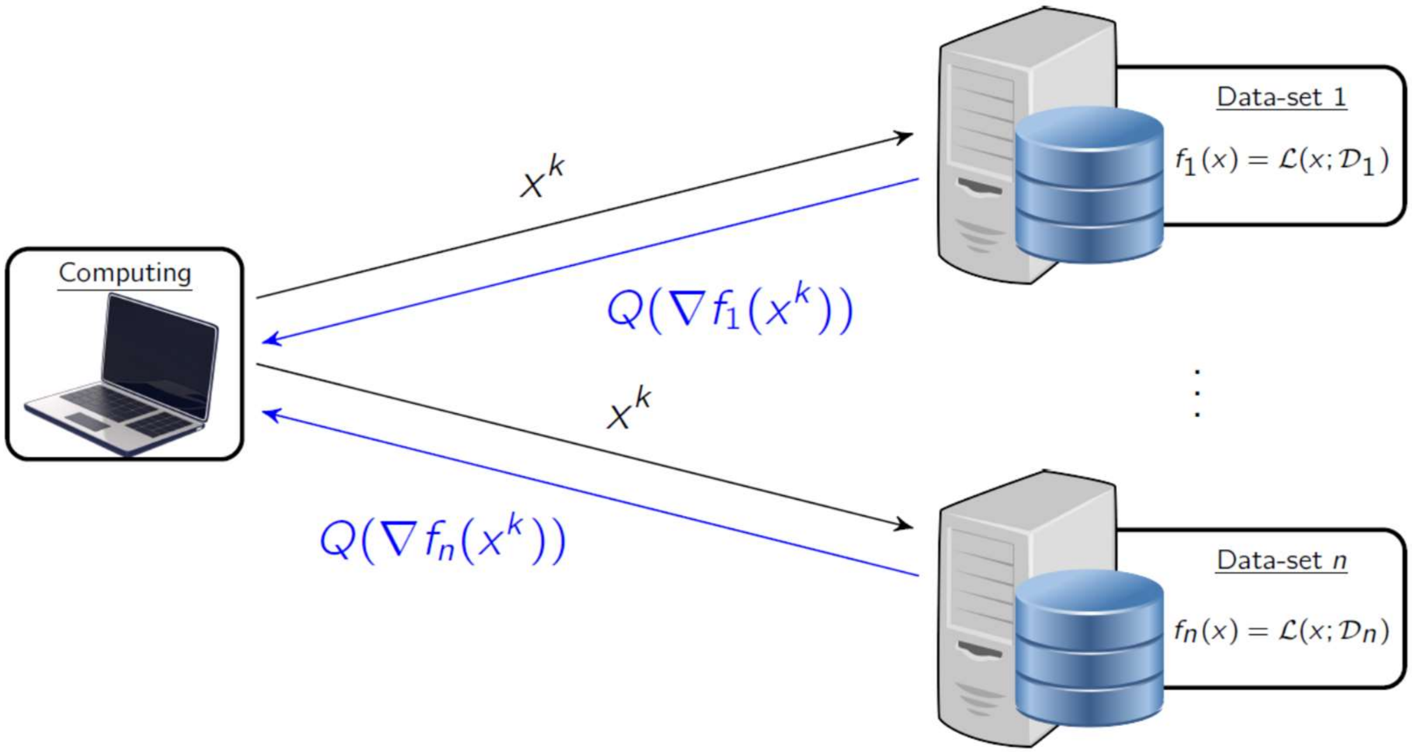
Compression Methods for Reducing Communications



$$x^{k+1} = x^k + \sum_{i=1}^{n} Q\left(\mathcal{T}_{f_i}(x^k) - x^k\right).$$

- Compression Q(·) can be
  - Sparsification: send only most important gradient elements.
  - Quantization: reduce precision on elements (e.g., sign compression).

Gradient Compression Methods



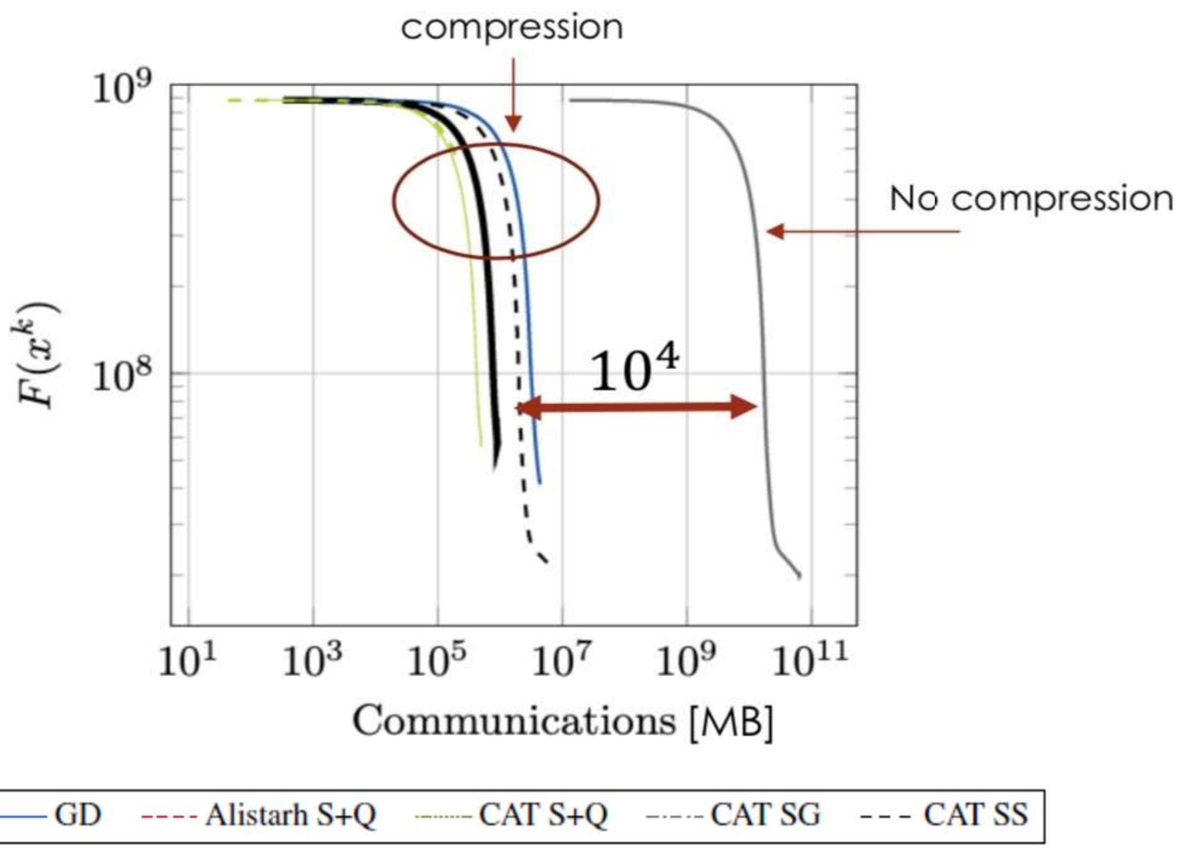
$$x^{k+1} = x^k - \gamma \sum_{i=1}^{n} Q(\nabla f_i(x^k)).$$

- Gradient compression in distributed learning

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Gradient compression works well in practice



- Distributed data: server (Ericsson Kista) 500 km away from the data (Lund).
- ×1000 communication saving, compared to full-precision algorithms.

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Lack of theoretical justification for gradient compression

Communication Complexity (Tsitsiklis & Luo, 1987)

$$\texttt{strongly convex:} \quad \underset{x \in \mathbb{R}^d}{\text{minimize}} \quad \sum_{i=1}^{n} f_i(x)$$

For every algorithm there exists $f_i(\cdot)$ such that

$$d \times \log\left(\frac{1}{\epsilon}\right) \text{ bits}$$

are need to be communicated to find an ϵ-solution.

- Communication complexity grows at least linearly with d .
- **In the worst case, compression does not improve efficiency!**

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Challenges

- Communication benefits are often realized after a careful tuning of the compression level before training
- Most existing compression schemes are agnostic of the disparate communication costs for different technologies
- No universally good compressor that works well on all problems (worst-case communication complexity of any optimization methods)
- Worst-case bounds do not explain communication efficiency improvements.
- Communication efficiency achieved by hyperparameter optimization.

Contributions

- Explain efficiency by data/problem dependent complexity bounds.
- Design adaptive compression algorithms that
  - maximize communication efficiency automatically
  - adjust to data on-line and communication technology used i.e.,
- Find a good balance between the communication savings and suboptimality guarantees of the solution
  - Focus on adaptive compression,
  - Strikes this balance by adjusting the compression level online, e.g.
  - by optimizing the transmitted bits per iteration.

Initial Setting: Sparsified Gradient Descent

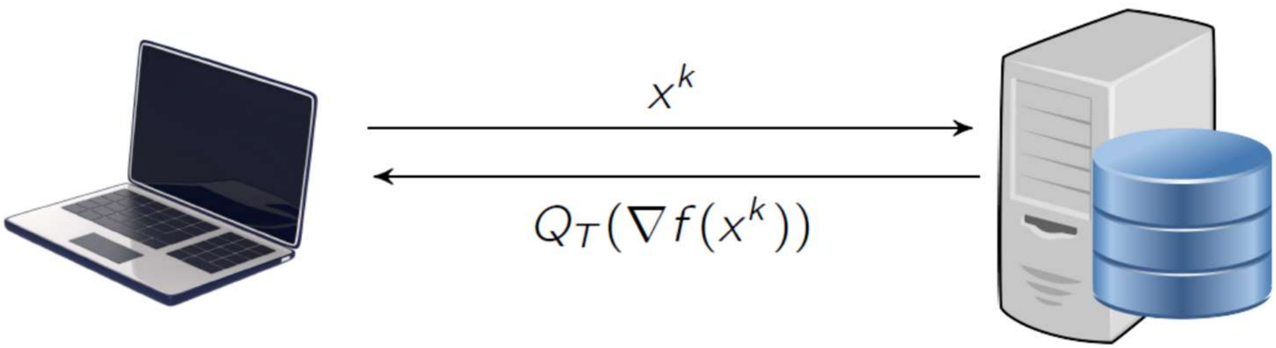
- Consider sparsified gradient descent

$$x^{k+1} = x^k - \gamma Q_T(\nabla f(x^k)),$$

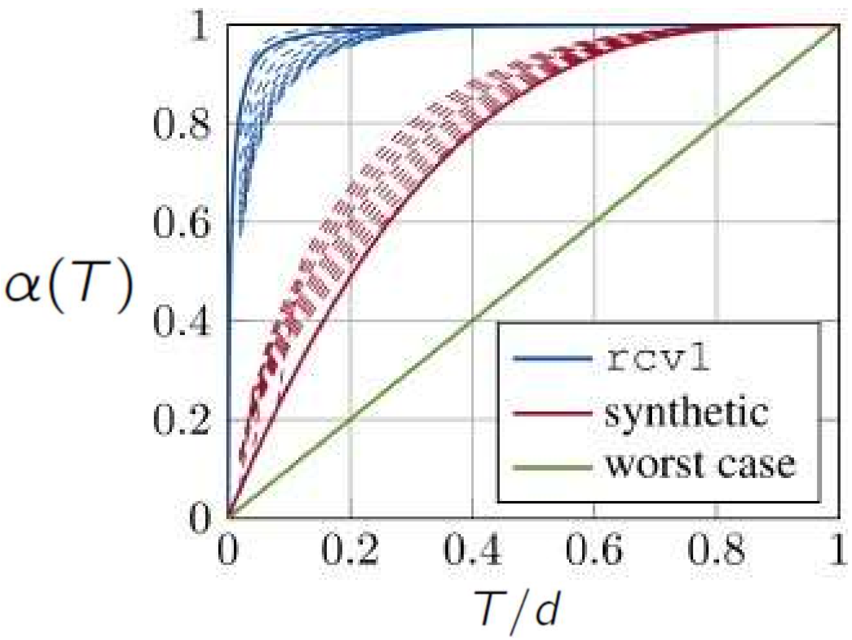
- where $Q_T(\cdot)$ with sparsity budget T is defined by

$$[Q_T(g)]_i = \begin{cases} g_i & \text{if} \quad i \in I_T(g) \\ 0 & \text{otherwise} \end{cases}$$

- where $I_T(g)$ has T indices of components with the largest absolute magnitude.



Why does sparsification improve communication efficiency?



Proportional Gradient Energy

$$\alpha(T) = \frac{||Q_T(\nabla f(x))||^2}{||\nabla f(x)||^2}$$

- **Worst Case**: Gradient energy is distributed evenly among all components.
  - $\alpha(T) = T/d$.
- **Real (sparse) Data**: Gradient energy is concentrated on few components.
  - $\alpha(T) \geq T/d$.

Sparsified Gradient Descent: Descent Lemma

Lemma (Generalized Descent Lemma)

Consider the minimization over $F(x)$ which is L-smooth and let $\gamma = 1/L$. Then for any $x, x^+ \in \mathbb{R}^d$ with

$$x^+ = x - \gamma Q_T(\nabla F(x))$$

we have

$$F(x^+) \leq F(x) - \frac{\alpha(T)}{2L}||\nabla F(x)||^2.$$

$$\alpha(T) \;=\; ||Q_T(\nabla F(x))||^2/||\nabla F(x)||^2$$

- *α(T)* implies the progress sparsification methods can make in each iteration.
- Classical gradient descent lemma when *α(T)* = 1.
- *α(T) ≥ T /d* (with equality for the worst-case energy distribution).

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Sparsified Gradient Descent: Data-dependent Complexity

- From the descent lemma, the data-dependent iteration complexities are derived

Iteration Complexity

Upper Bound	μ-convex	convex	nonconvex
No-Compression	A_ϵ^{SC}	A_ϵ^{C}	A_ϵ^{NC}
Data-Dependent	$\frac{1}{\bar{\alpha}_{\mathrm{Data}}} A_\epsilon^{\mathrm{SC}}$	$\frac{1}{\bar{\alpha}_{\mathrm{Data}}} A_\epsilon^{\mathrm{C}}$	$\frac{1}{\bar{\alpha}_{\mathrm{Data}}} A_\epsilon^{\mathrm{NC}}$
Worst-Case	$\frac{d}{T} A_\epsilon^{\mathrm{SC}}$	$\frac{d}{T} A_\epsilon^{\mathrm{C}}$	$\frac{d}{T} A_\epsilon^{\mathrm{NC}}$

where α(T) ≥ α¯Data.

Communicated bits to reach ϵ-accuracy (single precision)

- No compression: $32 A_\epsilon \times d$
- Worst case : $32 A_\epsilon \times d$
- Data-dependent : $32 A_\epsilon \times \frac{T}{\bar{\alpha}_{\mathrm{Data}}}$

$$\mathrm{SpeedUp}(T) = \frac{d}{T} \Big/ \frac{1}{\bar{\alpha}_T} = \frac{\bar{\alpha}_T}{T/d}$$

$$A_\epsilon^{\mathrm{SC}} = \kappa \log\left(\frac{F(x^0)-F^\star}{\epsilon}\right), \quad A_\epsilon^{\mathrm{C}} = \frac{2L\|x^0-x^\star\|^2}{\epsilon}, \quad A_\epsilon^{\mathrm{NC}} = \frac{2L(F(x^0)-F^\star)}{\epsilon}.$$

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Sparsified Gradient Descent: Data-dependent Complexity

- From the descent lemma, the data-dependent iteration complexities are derived

Iteration Complexity

Upper Bound	μ-convex	convex	nonconvex
No-Compression	A_ϵ^{SC}	A_ϵ^{C}	A_ϵ^{NC}
Data-Dependent	$\frac{1}{\bar{\alpha}_{\mathtt{Data}}} A_\epsilon^{\mathrm{SC}}$	$\frac{1}{\bar{\alpha}_{\mathtt{Data}}} A_\epsilon^{\mathrm{C}}$	$\frac{1}{\bar{\alpha}_{\mathtt{Data}}} A_\epsilon^{\mathrm{NC}}$
Worst-Case	$\frac{d}{T} A_\epsilon^{\mathrm{SC}}$	$\frac{d}{T} A_\epsilon^{\mathrm{C}}$	$\frac{d}{T} A_\epsilon^{\mathrm{NC}}$

where α(T) ≥ α¯Data.

Communicated bits to reach ϵ-accuracy (single precision)

- No compression: $32A_\epsilon \times d$
- Worst case : $32A_\epsilon \times d$
- Data-dependent : $32A_\epsilon \times \frac{T}{\bar{\alpha}_{\mathtt{Data}}}$

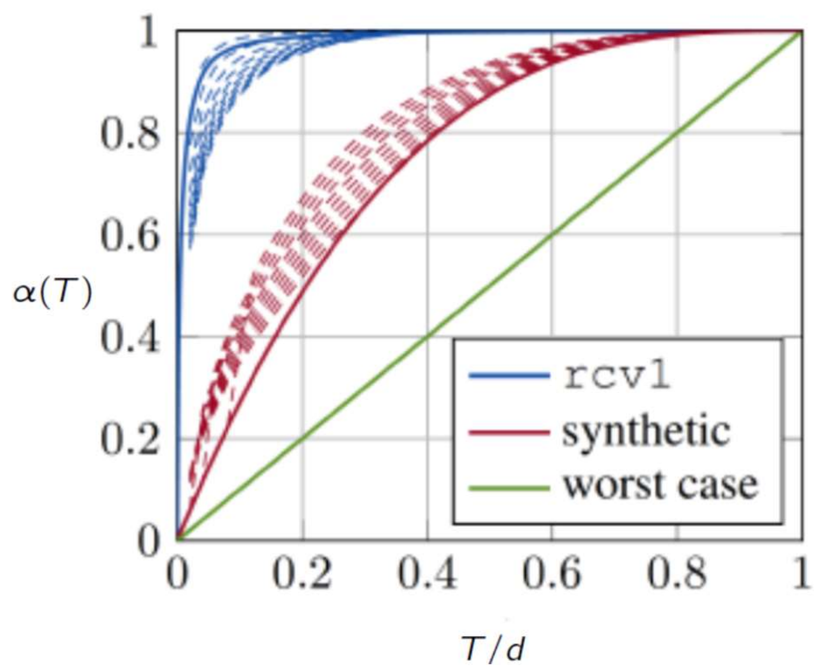
$$\mathtt{SpeedUp}(T) = \frac{d}{T} \Big/ \frac{1}{\bar{\alpha}_T} = \frac{\bar{\alpha}_T}{T/d}$$

$$A_\epsilon^{\mathrm{SC}} = \kappa \log\left(\frac{F(x^0) - F^\star}{\epsilon}\right), \quad A_\epsilon^{\mathrm{C}} = \frac{2L\|x^0 - x^\star\|^2}{\epsilon}, \quad A_\epsilon^{\mathrm{NC}} = \frac{2L(F(x^0) - F^\star)}{\epsilon}.$$

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Why does sparsification improve communication efficiency?



For RCV1 (and many real data-sets)

$$\frac{1}{\bar{\alpha}_{\mathrm{Data}}} << \frac{d}{T}$$

Data dependent bound: 1000 fold communication improvement!!

Adaptive gradient compression

- **Main Idea**: Find sparsity budget *T* that maximizes descent per iteration, i.e.

$$T = \underset{T=1,2,\ldots,d}{\operatorname{argmax}} \ \text{Efficiency}(T) := \underset{T=1,2,\ldots,d}{\operatorname{argmax}} \frac{\alpha(T)}{\text{Cost}(T)}$$

- *α(T)/Cost(T)* attains its minimum over T = 1, 2, . . . , d.
- *α(T)* and *Cost(T)* easily measured online.
  - *α(T)* adapted to compression used.
  - *Cost(T)* adapted to technology or application.

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Communication Cost: Bits, Packets, Energy and Beyond

- *Cost(T)* →

communication overhead

$$\boxed{C(T) = c_1 \times \lceil P(T)/P_{\max} \rceil + c_0}$$

cost of transmitting a single payload byte

number of payload bits per packet

payload

$$\boxed{\begin{aligned} P^{\mathrm{S}}(T) &= T \times (\lceil \log_2(d) \rceil + \mathrm{FPP}) \text{ bits} \\ P^{\mathrm{SQ}}(T) &= \mathrm{FPP} + T \times \lceil \log_2(d) \rceil \text{ bits,} \end{aligned}}$$

- For example, if we just count transmitted bits (**$c_1 = 1$**), then a single UDP packet transmitted over the Ethernet requires an overhead of,

  $c_0 = 54 \times 8$ bits and can have a payload of up to 1472 bytes.

Communication Adaptive Tuning (CAT) Algorithms

- At iteration $k = 0, 1, 2, \ldots$
- **Step 1** (Adaptive tuning):
  - tune T adaptively to optimize the communication efficiency
  - hyper-parameter optimization not required

$$\boxed{T^k = \operatorname*{argmax}_{T=1,2,\ldots,d} \frac{\alpha^k(T)}{\mathrm{Cost}(T)}}$$

- **Step 2** (Compressed gradient):

$$\boxed{x^{k+1} = x^k - \gamma Q_{T^k}(\nabla f(x^k))}$$

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Extensions of Communication Adaptive Tuning (CAT)

- Sparsification and Quantization (S+Q).

$$[Q_T(g)]_i = \begin{cases} \|g\|\mathtt{sign}(g_i) & \text{if } i \in I_i(g) \\ 0 & \text{otherwise.} \end{cases}$$

- Stochastic Sparsification (for stochastic, multi-node optimization).

$$[Q_T(g)]_i = \frac{g_i}{p_i}\xi,$$

- where

$$\xi \sim \mathtt{Bernoulli}(p_i)$$

$$\textstyle\sum_i p_i = T$$

ANTS LAB

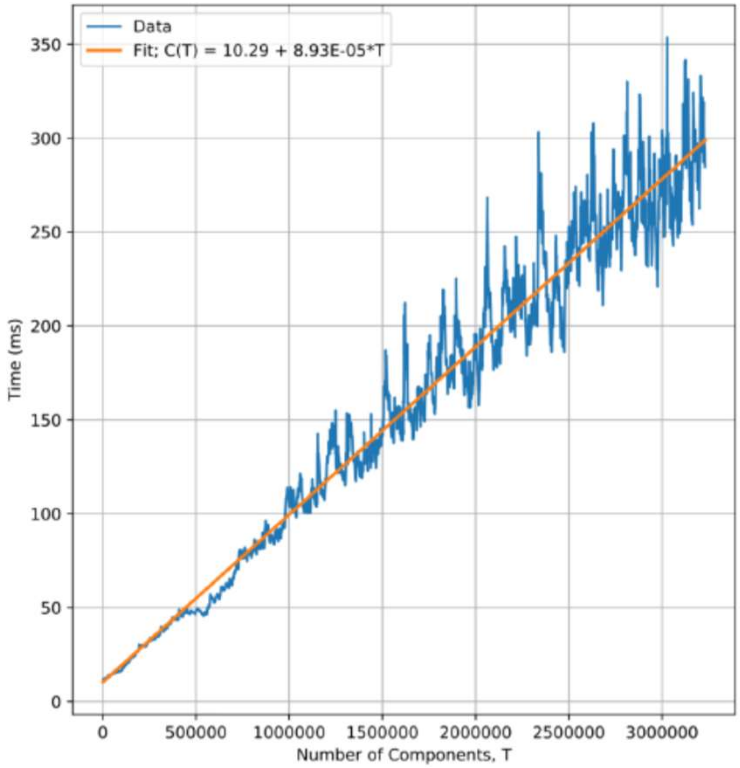
UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Communication Costs

- Many Possibilities:
  - Bits, energy, transmission time etc.
  - Build from protocols/standards
  - Build from empirical measurements

- Affine cost: Cost(T) = $c_0 + c_1 T$
  - c_0 packet header, c_1 cost per entree
  - Floats over Ethernet c_0 = 54, c1 = 4 + log2(d
  - Empirical Measurements (see figure)

- Packet cost: Ethernet $\mathtt{Cost}(T) = c_0 + c_1 \lceil \frac{P(T)}{P_{\max}} \rceil$
  - IEEE 802.15.4 (low energy wireless)

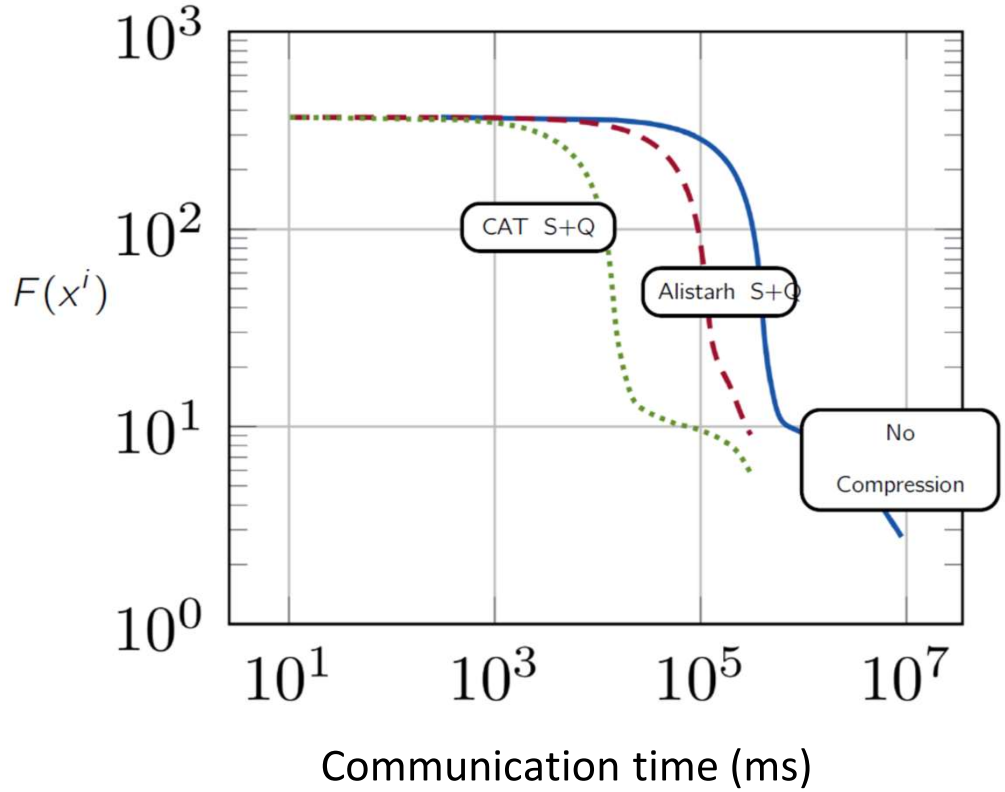


Communication times (s)

Experiment Settings

- CAT framework for dynamic sparsification and quantization (S+Q)
- Single node: single-master, single-worker setup
- URL data set with 2.4 million data points and 3.2 million features
- Distributed data: server (Ericsson Kista) 500 km away from the data (Lund)
- 1000 Mbit Internet connection using ZMQ library

Single-node Architecture

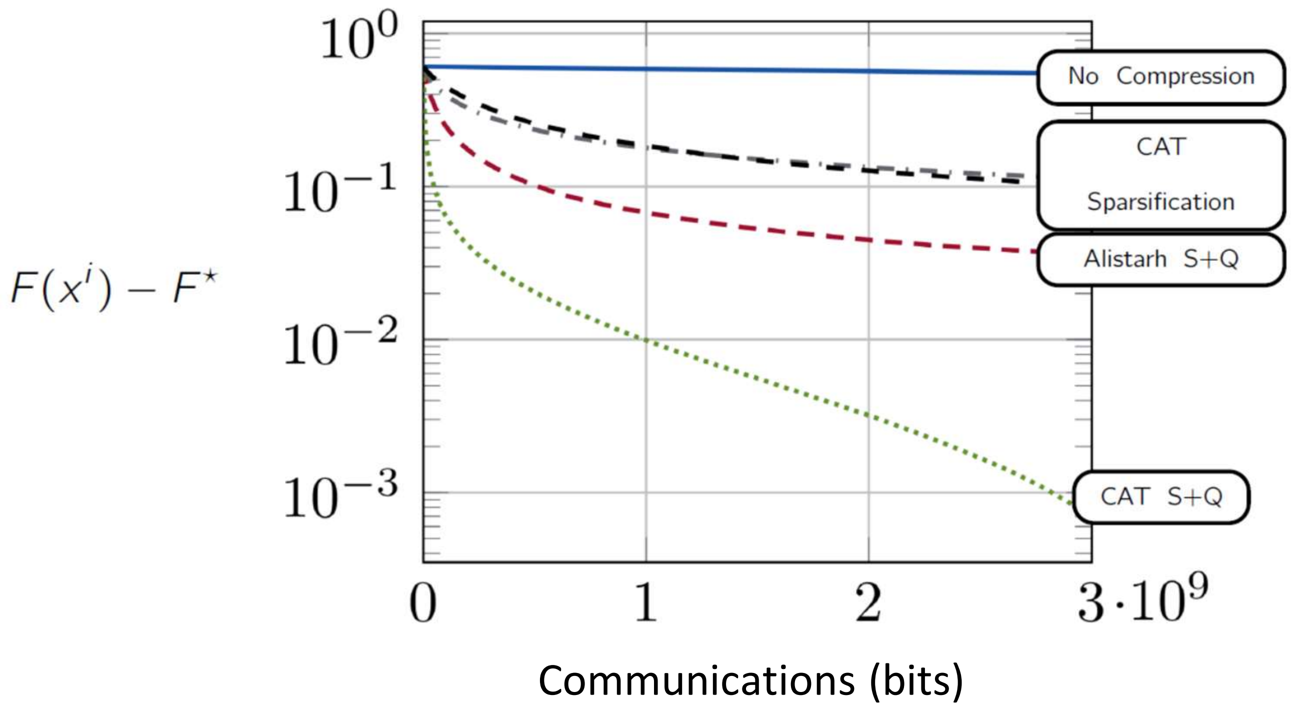


- CAT S+Q outperforms GD and Alistarh's S+Q up to two orders and one order of magnitude, respectively, in communication efficiency.

Experiment Settings – multi-node

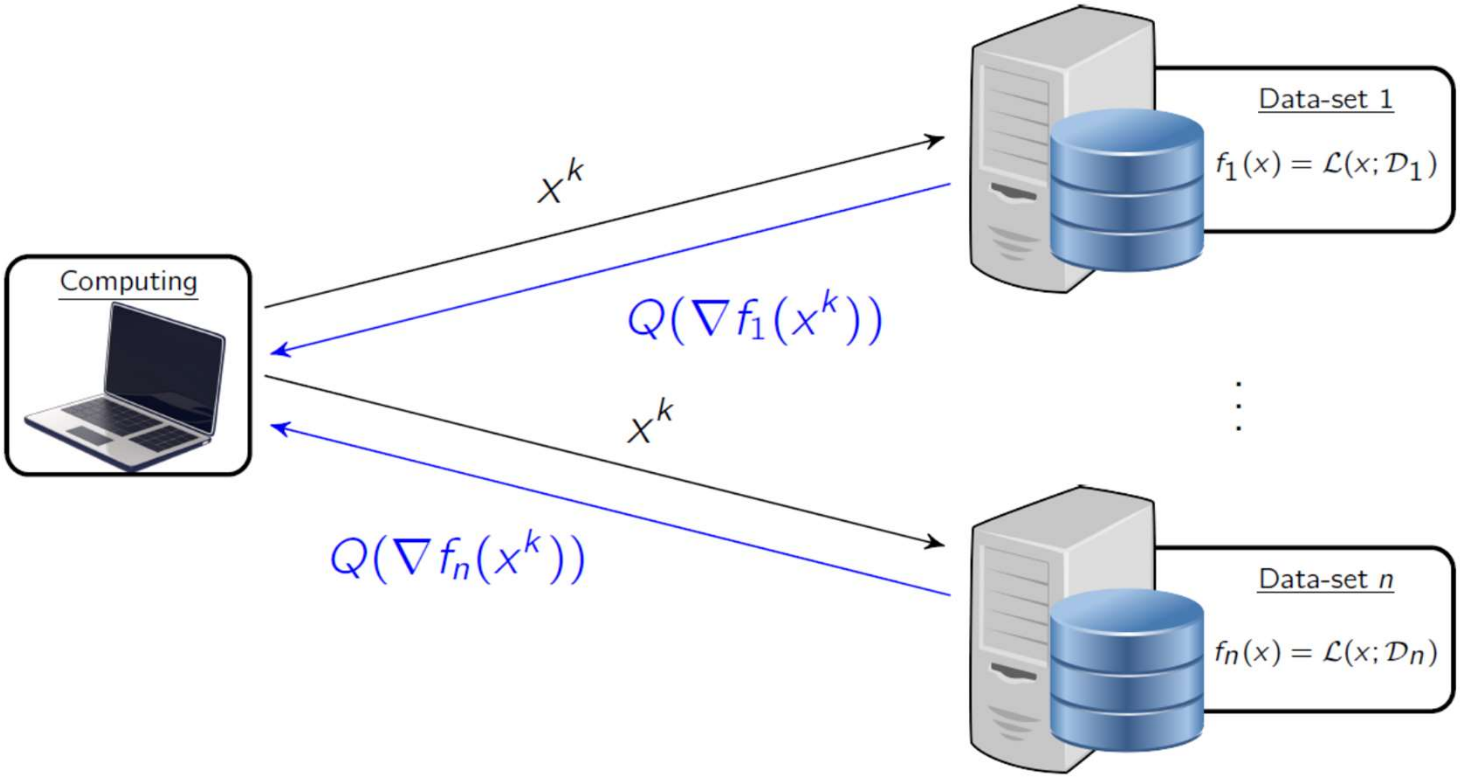
- CAT framework on
  - deterministic sparsification (SG)
  - stochastic sparsification (SS)
  - Sparsification with quantization (S+Q)
- RCV1 data set : 47,236 features, and 697,641 data points.
- Wireless communication scenario (e.g, IEEE 802.15.4) with 512 byte packets.
- Multi-node: 4 nodes using MPI, splitting the data evenly between the nodes.

Multiple-node Architecture



- CAT S+Q outperforms all other compression schemes
- CAT is roughly 6 times more communication efficient than (Alistarh et al. 2017) for the same compression scheme (compare number of bits needed to reach ϵ = 0.4).

Open Problems



- CAT for federated optimization
- CAT for error feedback, etc.

CAT Frameworks for Distributed Architectures

Lemma

Consider the minimization over $F(x) = \sum_{i=1}^{n} F_i(x)/n$, where each function $F_i(x)$ is μ-strongly convex and L-smooth. Let the sequence $\{x_k\}$ be generated by

$$x^{k+1} = x^k - \frac{\gamma^k}{n} \sum_{i=1}^{n} Q_T(\nabla F_i(x^k)),$$

where $\mathbf{E}[Q_T(v)] = v$ and $\omega(T) = \|v\|^2/\mathbf{E}\|Q_T(v)\|^2$, and let $\gamma^k = \alpha/(k+1)$ for $\alpha > 0$. Then, the communication complexity to reach $\mathbf{E}[F(x^k) - F^\star] \leq \epsilon$ is

$$\frac{\mathtt{Cost}(T)}{\omega_{\max}(T)} \cdot \min\left(\frac{B_1}{\sqrt{\epsilon}}, \frac{B_2}{\epsilon}\right),$$

for $B_1, B_2 > 0$.

- Limitations:
  - CAT with the same compression level T for all clients.

CAT Frameworks for Distributed Stochastic Sparsification

Lemma

Consider the minimization over $F(x) = \sum_{i=1}^{n} F_i(x)/n$*, where each function* $F_i(x)$ *is* μ*-strongly convex and* L*-smooth. Let the sequence* $\{x_k\}$ *be generated by*

$$x^{k+1} = x^k - \frac{\gamma}{n} \sum_{i=1}^{n} Q_T(\nabla F_i(x^k))$$

where $\mathbf{E}[Q_T(v)] = v$ *and* $\omega(T) = \|v\|^2 / \mathbf{E}\|Q_T(v)\|^2$*, and let* $\gamma_k = \alpha/(k+1)$ *for* $\alpha > 0$*. Then, the communication complexity is*

$$\frac{\mathtt{Cost}(T)}{\omega_{\max}(T)} \cdot \min\left(\frac{B_1}{\sqrt{\epsilon}}, \frac{B_2}{\epsilon}\right).$$

for $B_1, B_2 > 0$.

- **Questions**:

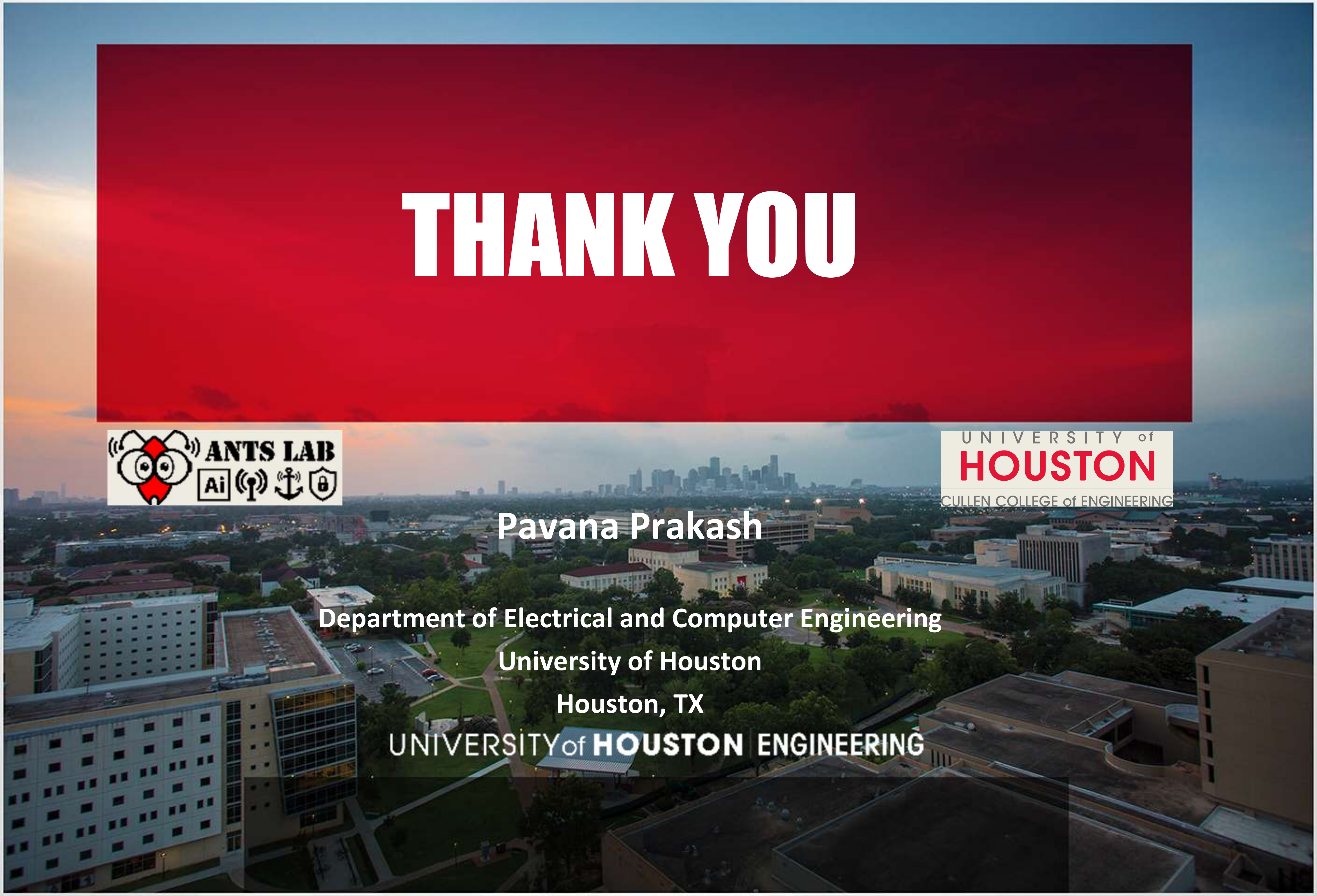
1. How to tune local compression level without synchronization?
2. How to tune deterministic compression for federated architectures?

Conclusions

- **Existing Works:**
- Worst-case bound does not explain communication efficiency by compression.
- Compressions not adapted to technology or relevant communication costs.

- **Contributions**:
- Explain improved efficiency by data/problem dependent complexity.
- Design adaptive compression that
  - optimizes overall communication efficiency automatically.
  - adjusts to data online and to any communication technology/application used.
  - leads to significant communication savings, compared to existing compression.

- **Open Problems**: CAT frameworks for distributed and federated optimization



THANK YOU

ANTS LAB

UNIVERSITY of HOUSTON
CULLEN COLLEGE of ENGINEERING

Pavana Prakash

Department of Electrical and Computer Engineering
University of Houston
Houston, TX

UNIVERSITY of HOUSTON | ENGINEERING