

# Basis Matters: Better Communication-Efficient Second Order Methods for Federated Learning\*



INSTITUT  
POLYTECHNIQUE  
DE PARIS



جامعة الملك عبد الله  
للعلوم والتقنية  
King Abdullah University of  
Science and Technology

**Rustem Islamov**

Institut Polytechnique de Paris (IP Paris)  
King Abdullah University of Science and Technology

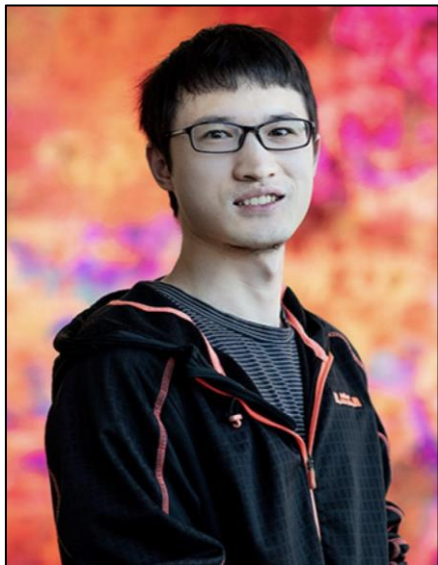


\*Xun Qian, Rustem Islamov, Mher Safaryan, and Peter Richtárik

**Basis Matters: Better Communication-Efficient Second Order Methods for Federated Learning**

arXiv preprint arXiv: 2111.01847, 2021

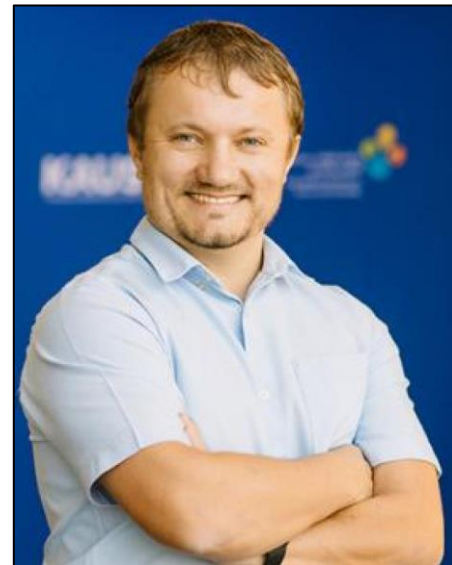
# Co-Authors



**Xun Qian**  
Postdoctoral Fellow  
KAUST



**Mher Safaryan**  
Postdoctoral Fellow  
KAUST



**Peter Richtárik**  
Professor  
KAUST




# The Problem

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^n f_i(x) \right\}$$

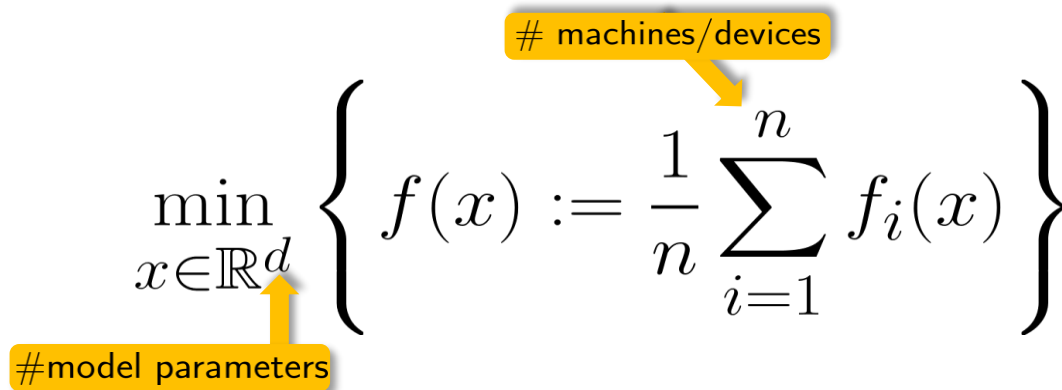
# The Problem

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^n f_i(x) \right\}$$

#model parameters



# The Problem



The diagram illustrates a distributed optimization problem. It features a mathematical expression for minimizing a function  $f(x)$  over a parameter space  $x \in \mathbb{R}^d$ . The function  $f(x)$  is defined as the average of  $n$  local functions  $f_i(x)$ . Annotations in yellow boxes with arrows provide context: 

- An arrow points from the label "#model parameters" to the  $d$  in  $\mathbb{R}^d$ .
- An arrow points from the label "# machines/devices" to the  $n$  in the summation.

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^n f_i(x) \right\}$$

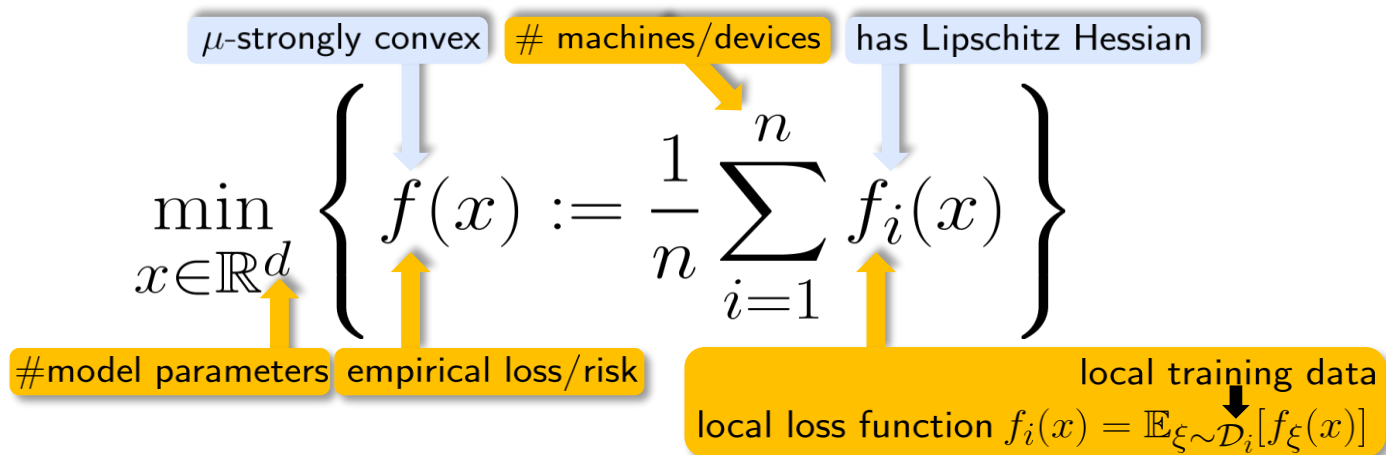
# The Problem

The diagram illustrates the optimization problem with several annotations:

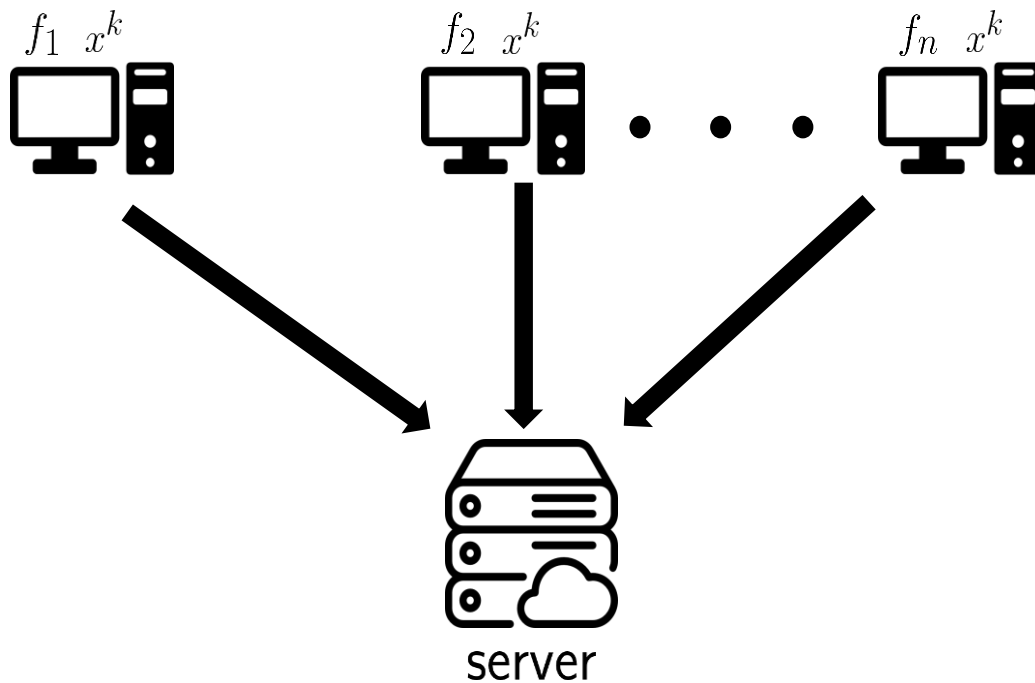
- A light blue box labeled  $\mu$ -strongly convex has a blue arrow pointing down to the function  $f(x)$  in the objective.
- A yellow box labeled # machines/devices has a yellow arrow pointing down to the summation index  $n$  in the objective.
- A yellow box labeled #model parameters has a yellow arrow pointing up to the domain  $x \in \mathbb{R}^d$ .
- A yellow box labeled empirical loss/risk has a yellow arrow pointing up to the function  $f(x)$  in the objective.

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^n f_i(x) \right\}$$

# The Problem

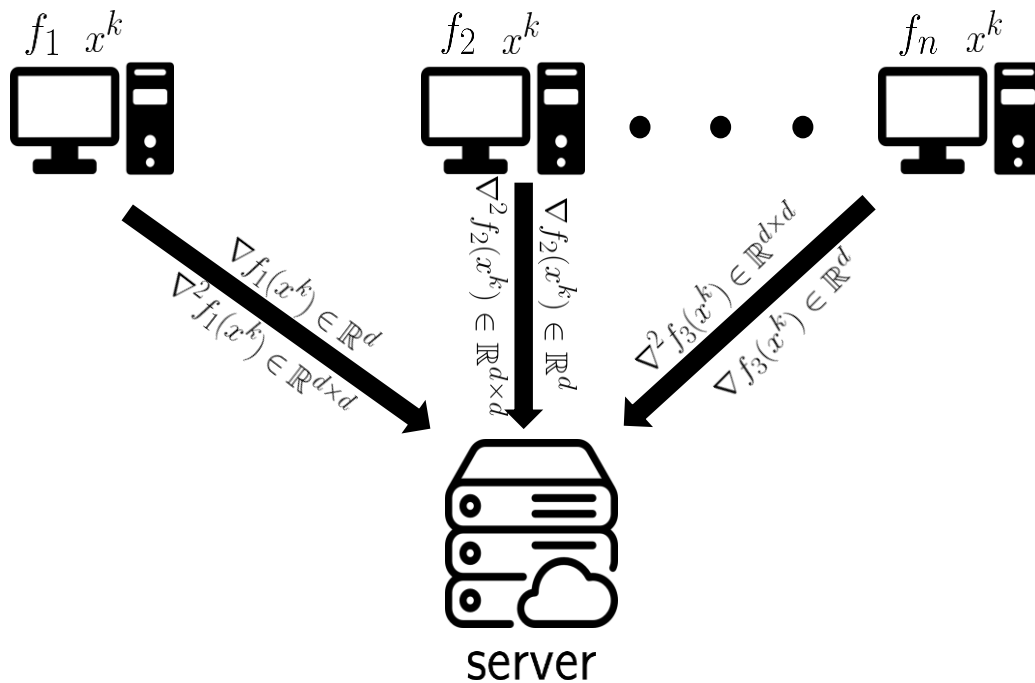


# Centralized Setting

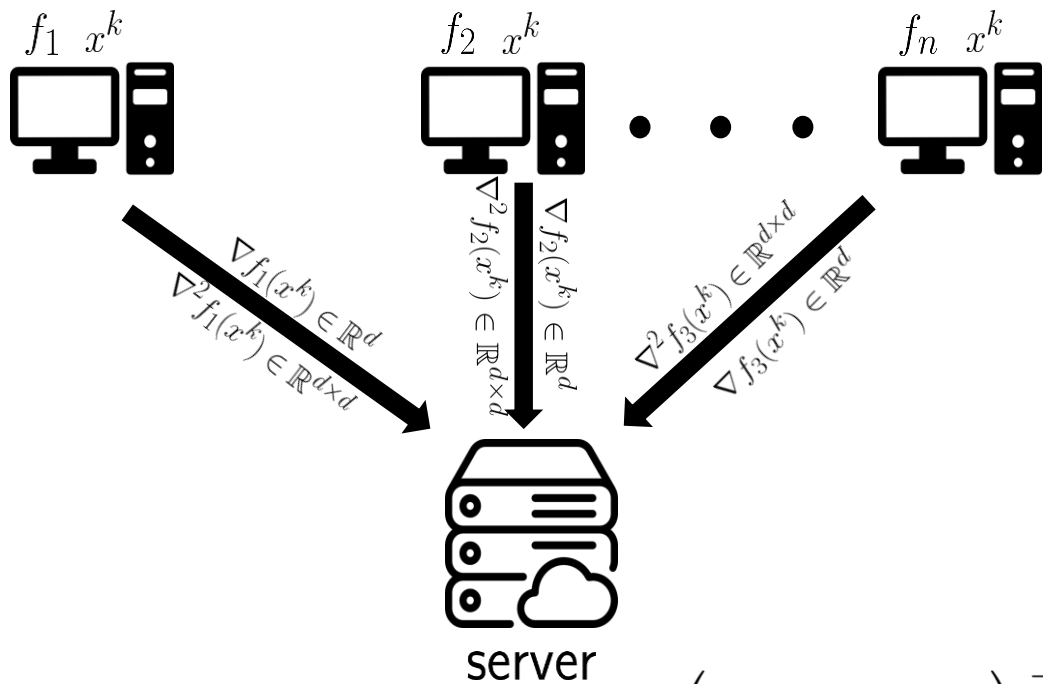




# Centralized Setting

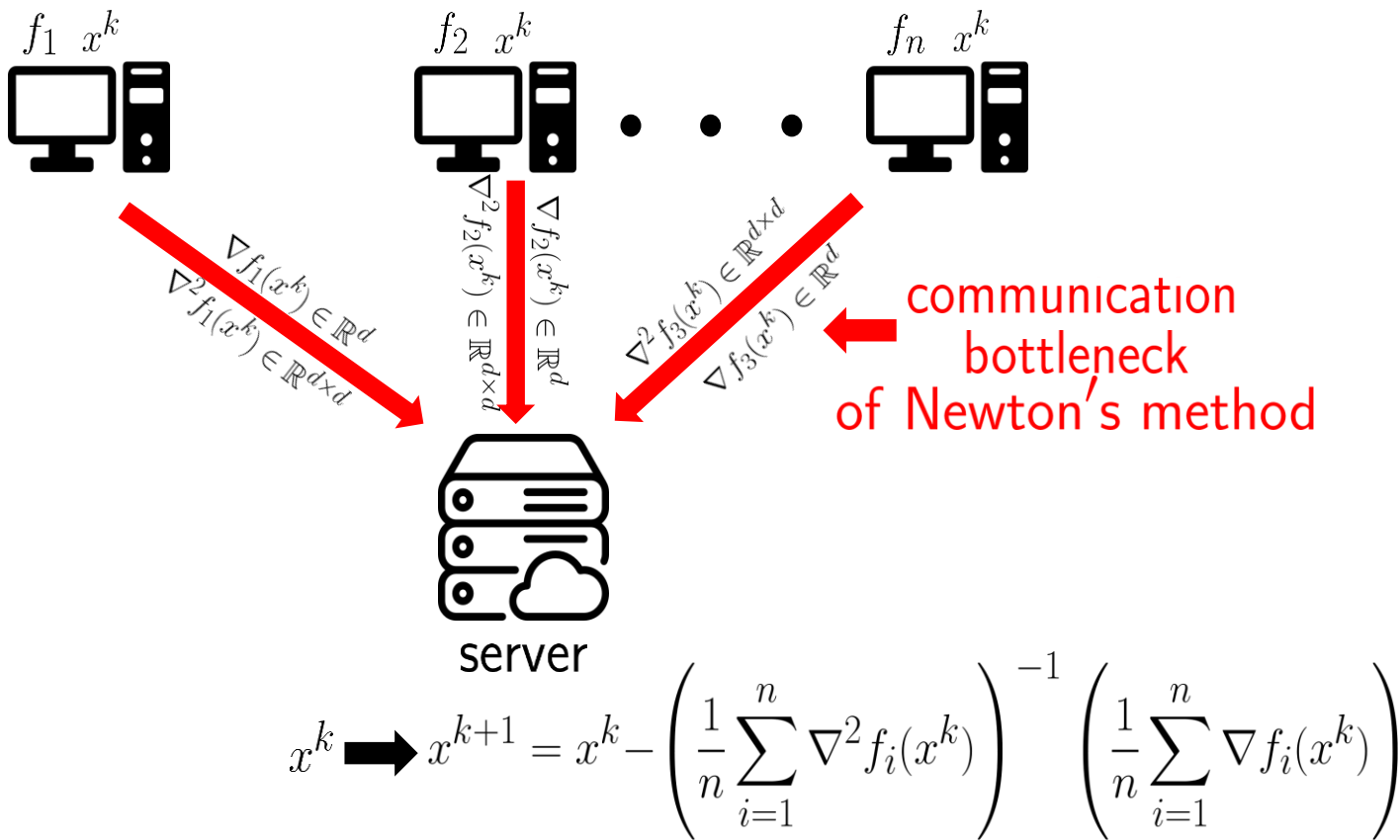


# Centralized Setting



$$x^k \rightarrow x^{k+1} = x^k - \left( \frac{1}{n} \sum_{i=1}^n \nabla^2 f_i(x^k) \right)^{-1} \left( \frac{1}{n} \sum_{i=1}^n \nabla f_i(x^k) \right)$$

# Centralized Setting



# Existing approaches and their disadvantages

## First order methods

- ✗ Rates depend on the condition number
- ✗ Hard to find optimal stepsizes

## Second order methods

- ✗ Rates depend on the condition number
- ✗ Communication cost is high

# Existing approaches and their disadvantages

## First order methods

- ✗ Rates depend on the condition number
- ✗ Hard to find optimal stepsizes

## Second order methods

- ✗ Rates depend on the condition number
- ✗ Communication cost is high

## GOAL

Develop a communication-efficient distributed Newton-type method whose (local) convergence rate is independent of the condition number

- ✓ Can provably benefit from communication compression
- ✓ Rate independent of the condition number
- ✓ Supports bidirectional compression
- ✗ Good rate for local convergence only

# Related Works

Table 1: Theoretical comparison of 7 second order methods (including ours). Advantages are written in **green**, while limitations are colored in **red**.

Method	Problem	Assumptions	CC <sup>1</sup>	Rate	Comments
GIANT [Wang et al., 2018]	GLM <sup>3</sup>	LipC <sup>2</sup> Hessian, convex + $l_2$ reg., $\approx$ i.i.d. data	$\mathcal{O}(d)$	Local $\kappa$ -dependent linear. Global $\mathcal{O}(\log \kappa/\epsilon)$ , quadratics	Big data regime (#data $\gg d$ )
DINGO [Crane and Roosta, 2019]	GFS <sup>4</sup>	Moral Smoothness <sup>5</sup> , $\approx$ strong convexity <sup>6</sup>	$\mathcal{O}(d)$	Global linear rate. No fast local rate.	Operates full gradients, Hessian-vector products, Hessian pseudo-inverse and vector products.
DAN [Zhang et al., 2020]	GFS	LipC Hessian, strong convexity	$\mathcal{O}(nd^2)$	Global quadratic rate after $\mathcal{O}(L/\mu^2)$ iterations.	Operates full gradients and Hessian matrices.
DAN-LA [Zhang et al., 2020]	GFS	LipC Hessian, LipC gradient, strong convexity	$\mathcal{O}(nd)$	Asymptotic and implicit global superlinear rate.	$\lim_{k \rightarrow \infty} \frac{\ x_{k+1} - x^*\ }{\ x_k - x^*\ } = 0$ Independent of $\kappa$ ? Better non-asymptotic complexity over linear rate ?
NL [Islamov et al., 2021]	GLM	LipC Hessian, convex + $l_2$ reg.	$\mathcal{O}(d)$	Local superlinear rate independent of $\kappa$ , but dependent on #data. Global linear rate.	reveals local data to server
Quantized Newton [Alimisis et al., 2021]	GFS	LipC Hessian, LipC gradient, strong convexity <sup>6</sup>	$\tilde{\mathcal{O}}(d^2)$	Local (fixed) linear rate. No global rate.	Operates full gradients and Hessian matrices.
FedNL [Safaryan et al., 2021]	GFS	LipC Hessian, strong convexity	$\mathcal{O}(d)$	Local (fixed) linear rate. Local superlinear rate independent of $\kappa$ , independent of #data. Global linear rate.	Operates full gradients and Hessian matrices. Supports contractive Hessian compression. Extensions <sup>†</sup>
Basis Learn (this work)	GFS	LipC Hessian, strong convexity	$\mathcal{O}(d)$	Local (fixed) linear rate. Local superlinear rate independent of $\kappa$ , independent of #data.	Operates full gradients and Hessian matrices. Supports contractive Hessian compression. Extensions <sup>†</sup>

<sup>1</sup> CC = Communication Cost per iteration.

<sup>2</sup> LipC = Lipschitz Continuous.

<sup>3</sup> GLM = Generalized Linear Model, e.g.  $\text{loss}_j(x; a_j) = \phi_j(a_j^\top x) + \lambda \|x\|^2$ .

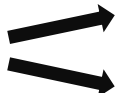
<sup>4</sup> GFS = General Finite Sum.

<sup>5</sup> Moral Smoothness:  $\|\nabla^2 f(x) \nabla f(x) - \nabla^2 f(y) \nabla f(y)\| \leq L \|x - y\|$ .

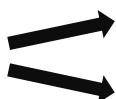
<sup>6</sup> Applies to local loss functions for all clients.

<sup>†</sup> Partial Participation and Bidirectional Compression.

# Motivation: utilizing the data structure

Naïve Implementation:  $\nabla f_i(x^k) \in \mathbb{R}^d, \nabla^2 f_i(x^k) \in \mathbb{R}^{d \times d}$   Expensive communication  $\mathcal{O}(d^2)$   
Local Quadratic Rate

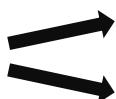
# Motivation: utilizing the data structure

Naïve Implementation:  $\nabla f_i(x^k) \in \mathbb{R}^d, \nabla^2 f_i(x^k) \in \mathbb{R}^{d \times d}$   Expensive communication  $\mathcal{O}(d^2)$   
Local Quadratic Rate

$$\min_x \left\{ f(x) := \frac{1}{n} \sum_{i=1}^n f_i(x) \right\}, \quad f_i(x) = \frac{1}{m} \sum_{j=1}^m \varphi_{ij}(a_{ij}^\top x)$$



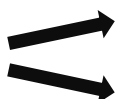
# Motivation: utilizing the data structure

Naïve Implementation:  $\nabla f_i(x^k) \in \mathbb{R}^d, \nabla^2 f_i(x^k) \in \mathbb{R}^{d \times d}$   Expensive communication  $\mathcal{O}(d^2)$   
Local Quadratic Rate

$$\min_x \left\{ f(x) := \frac{1}{n} \sum_{i=1}^n f_i(x) \right\}, \quad f_i(x) = \frac{1}{m} \sum_{j=1}^m \varphi_{ij}(a_{ij}^\top x)$$

Assumption:  $\{a_{ij} \in \mathbb{R}^d : j \in [m]\} \subseteq G_i, \quad \dim G_i = r \Rightarrow$  Fix basis  $\{v_{it}\}_{t=1}^r$

# Motivation: utilizing the data structure

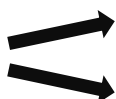
Naïve Implementation:  $\nabla f_i(x^k) \in \mathbb{R}^d, \nabla^2 f_i(x^k) \in \mathbb{R}^{d \times d}$   Expensive communication  $\mathcal{O}(d^2)$   
Local Quadratic Rate

$$\min_x \left\{ f(x) := \frac{1}{n} \sum_{i=1}^n f_i(x) \right\}, \quad f_i(x) = \frac{1}{m} \sum_{j=1}^m \varphi_{ij}(a_{ij}^\top x)$$

Assumption:  $\{a_{ij} \in \mathbb{R}^d : j \in [m]\} \subseteq G_i, \quad \dim G_i = r \xrightarrow{\quad} \text{Fix basis } \{v_{it}\}_{t=1}^r$

New data representation:  $a_{ij} = \sum_{t=1}^r \alpha_{ijt} v_{it}, j \in [m]$

# Motivation: utilizing the data structure

Naïve Implementation:  $\nabla f_i(x^k) \in \mathbb{R}^d, \nabla^2 f_i(x^k) \in \mathbb{R}^{d \times d}$   Expensive communication  $\mathcal{O}(d^2)$   
Local Quadratic Rate

$$\min_x \left\{ f(x) := \frac{1}{n} \sum_{i=1}^n f_i(x) \right\}, \quad f_i(x) = \frac{1}{m} \sum_{j=1}^m \varphi_{ij}(a_{ij}^\top x)$$

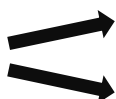
Assumption:  $\{a_{ij} \in \mathbb{R}^d : j \in [m]\} \subseteq G_i, \quad \dim G_i = r \Rightarrow$  Fix basis  $\{v_{it}\}_{t=1}^r$

New data representation:  $a_{ij} = \sum_{t=1}^r \alpha_{ijt} v_{it}, j \in [m]$

New Hessian  
representation:

$$\begin{aligned} \nabla^2 f_i(x) &= \frac{1}{m} \sum_{j=1}^m \varphi''_{ij}(a_{ij}^\top x) a_{ij} a_{ij}^\top = \frac{1}{m} \sum_{j=1}^m \varphi''_{ij}(a_{ij}^\top x) \sum_{t,l=1}^r \alpha_{ijt} \alpha_{ijl} v_{it} v_{il}^\top \\ &= \sum_{t,l=1}^r \left[ \frac{1}{m} \sum_{j=1}^m \varphi''_{ij}(a_{ij}^\top x) \alpha_{ijt} \alpha_{ijl} \right] v_{it} v_{il}^\top \end{aligned}$$

# Motivation: utilizing the data structure

Naïve Implementation:  $\nabla f_i(x^k) \in \mathbb{R}^d, \nabla^2 f_i(x^k) \in \mathbb{R}^{d \times d}$   Expensive communication  $\mathcal{O}(d^2)$   
Local Quadratic Rate

$$\min_x \left\{ f(x) := \frac{1}{n} \sum_{i=1}^n f_i(x) \right\}, \quad f_i(x) = \frac{1}{m} \sum_{j=1}^m \varphi_{ij}(a_{ij}^\top x)$$

Assumption:  $\{a_{ij} \in \mathbb{R}^d : j \in [m]\} \subseteq G_i, \quad \dim G_i = r \Rightarrow$  Fix basis  $\{v_{it}\}_{t=1}^r$

New data representation:  $a_{ij} = \sum_{t=1}^r \alpha_{ijt} v_{it}, j \in [m]$

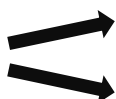
$$\nabla^2 f_i(x) = \frac{1}{m} \sum_{j=1}^m \varphi_{ij}''(a_{ij}^\top x) a_{ij} a_{ij}^\top = \frac{1}{m} \sum_{j=1}^m \varphi_{ij}''(a_{ij}^\top x) \sum_{t,l=1}^r \alpha_{ijt} \alpha_{ijl} v_{it} v_{il}^\top$$

New Hessian  
representation:

$$= \sum_{t,l=1}^r \left[ \frac{1}{m} \sum_{j=1}^m \varphi_{ij}''(a_{ij}^\top x) \alpha_{ijt} \alpha_{ijl} \right] v_{it} v_{il}^\top$$

 New basis in the space  
of matrices

# Motivation: utilizing the data structure

Naïve Implementation:  $\nabla f_i(x^k) \in \mathbb{R}^d, \nabla^2 f_i(x^k) \in \mathbb{R}^{d \times d}$   Expensive communication  $\mathcal{O}(d^2)$   
Local Quadratic Rate

$$\min_x \left\{ f(x) := \frac{1}{n} \sum_{i=1}^n f_i(x) \right\}, \quad f_i(x) = \frac{1}{m} \sum_{j=1}^m \varphi_{ij}(a_{ij}^\top x)$$

Assumption:  $\{a_{ij} \in \mathbb{R}^d : j \in [m]\} \subseteq G_i, \quad \dim G_i = r \Rightarrow$  Fix basis  $\{v_{it}\}_{t=1}^r$

New data representation:  $a_{ij} = \sum_{t=1}^r \alpha_{ijt} v_{it}, j \in [m]$

$$\begin{aligned} \nabla^2 f_i(x) &= \frac{1}{m} \sum_{j=1}^m \varphi''_{ij}(a_{ij}^\top x) a_{ij} a_{ij}^\top = \frac{1}{m} \sum_{j=1}^m \varphi''_{ij}(a_{ij}^\top x) \sum_{t,l=1}^r \alpha_{ijt} \alpha_{ijl} v_{it} v_{il}^\top \\ &= \sum_{t,l=1}^r \left[ \frac{1}{m} \sum_{j=1}^m \varphi''_{ij}(a_{ij}^\top x) \alpha_{ijt} \alpha_{ijl} \right] v_{it} v_{il}^\top \end{aligned}$$

New Hessian  
representation:

New representation can be  
derived for gradients too

# Motivation: utilizing the data structure

Naïve Implementation:  $\nabla f_i(x^k) \in \mathbb{R}^d, \nabla^2 f_i(x^k) \in \mathbb{R}^{d \times d}$   $\begin{matrix} \nearrow \text{Expensive communication } \mathcal{O}(d^2) \\ \searrow \text{Local Quadratic Rate} \end{matrix}$

**New Hessian and gradient representations requires  $\mathcal{O}(r^2)$  floats. In the extreme cases of  $r = \mathcal{O}(1)$  we run Newton's method with  $\mathcal{O}(1)$  cost per iteration!**

New data representation:  $a_{ij} = \sum_{t=1}^m \alpha_{ijt} v_{it}, j \in [m]$

New Hessian representation:

$$\begin{aligned} \nabla^2 f_i(x) &= \frac{1}{m} \sum_{j=1}^m \varphi''_{ij}(a_{ij}^\top x) a_{ij} a_{ij}^\top = \frac{1}{m} \sum_{j=1}^m \varphi''_{ij}(a_{ij}^\top x) \sum_{t,l=1}^r \alpha_{ijt} \alpha_{ijl} v_{it} v_{il}^\top \\ &= \sum_{t,l=1}^r \left[ \frac{1}{m} \sum_{j=1}^m \varphi''_{ij}(a_{ij}^\top x) \alpha_{ijt} \alpha_{ijl} \right] v_{it} v_{il}^\top \end{aligned}$$

# Newton Star

$$x^{k+1} = x^k - \left( \frac{1}{n} \sum_{i=1}^n \nabla^2 f_i(x^*) \right)^{-1} \left( \frac{1}{n} \sum_{i=1}^n \nabla f_i(x^k) \right)$$



Rustem Islamov, Xun Qian and Peter Richtárik  
**Distributed Second Order Methods with Fast Rates  
and Compressed Communication,**  
*ICML 2021.*

# Newton Star

$$x^{k+1} = x^k - \left( \underbrace{\frac{1}{n} \sum_{i=1}^n \nabla^2 f_i(x^*)}_{\nabla^2 f(x^*)} \right)^{-1} \left( \frac{1}{n} \sum_{i=1}^n \nabla f_i(x^k) \right)$$

$x^* := \arg \min_x f(x)$

We assume this is known  $\rightarrow \nabla^2 f(x^*)$

The diagram illustrates the Newton Star algorithm. The main equation is  $x^{k+1} = x^k - \left( \frac{1}{n} \sum_{i=1}^n \nabla^2 f_i(x^*) \right)^{-1} \left( \frac{1}{n} \sum_{i=1}^n \nabla f_i(x^k) \right)$ . An orange box at the top defines  $x^* := \arg \min_x f(x)$ , with an arrow pointing to the  $x^*$  in the Hessian term of the equation. Another orange box at the bottom left states 'We assume this is known', with an arrow pointing to a box containing  $\nabla^2 f(x^*)$ . A bracket under the Hessian term in the equation points down to this box, indicating that the average of the Hessians is assumed to be known.



# Newton Star

$$x^{k+1} = x^k - \left( \underbrace{\frac{1}{n} \sum_{i=1}^n \nabla^2 f_i(x^*)}_{\substack{\text{We assume this is} \\ \text{known}}} \right)^{-1} \left( \frac{1}{n} \sum_{i=1}^n \nabla f_i(x^k) \right)$$

$x^* := \arg \min_x f(x)$

$\nabla^2 f(x^*)$

Can be computed by machine  $i$

The diagram illustrates the Newton Star algorithm equation. The equation is  $x^{k+1} = x^k - \left( \frac{1}{n} \sum_{i=1}^n \nabla^2 f_i(x^*) \right)^{-1} \left( \frac{1}{n} \sum_{i=1}^n \nabla f_i(x^k) \right)$ . Annotations include: a yellow box at the top defining  $x^* := \arg \min_x f(x)$  with an arrow pointing to  $x^*$  in the Hessian term; a yellow box at the bottom left stating 'We assume this is known' with an arrow pointing to the Hessian term; a yellow box at the bottom center containing  $\nabla^2 f(x^*)$  with a bracket pointing to the Hessian term; and a pink box at the bottom right stating 'Can be computed by machine  $i$ ' with an arrow pointing to the gradient term  $\nabla f_i(x^k)$ .

# Newton Star

$$x^{k+1} = x^k - \left( \frac{1}{n} \sum_{i=1}^n \nabla^2 f_i(x^*) \right)^{-1} \left( \frac{1}{n} \sum_{i=1}^n \nabla f_i(x^k) \right)$$

✓ Local quadratic convergence rate independent of the condition number

✓ Cheap  $O(d)$  communication cost

✗ The Hessian at the optimum is unknown

Hessian Lipschitz constant

$$\|x^{k+1} - x^*\| \leq \frac{H}{2\mu} \|x^k - x^*\|^2$$

Strong convexity constant

# Basis Learn: Idea

Fix basis:  $\{\mathbf{B}_i^{jl} : j, l \in [d]\} \longrightarrow \forall \mathbf{A} \in \mathbb{R}^{d \times d} \hookrightarrow \mathbf{A} = \sum_{j,l} h_{j,l}^i(\mathbf{A}) \mathbf{B}_i^{jl}$

# Basis Learn: Idea

Fix basis:  $\{\mathbf{B}_i^{jl} : j, l \in [d]\} \longrightarrow \forall \mathbf{A} \in \mathbb{R}^{d \times d} \hookrightarrow \mathbf{A} = \sum_{j,l} h_{j,l}^i(\mathbf{A}) \mathbf{B}_i^{jl}$

Define:  $h^i(\mathbf{A}) \in \mathbb{R}^{d \times d} : h^i(\mathbf{A})_{j,l} = h_{j,l}^i(\mathbf{A})$

# Basis Learn: Idea

Fix basis:  $\{\mathbf{B}_i^{jl} : j, l \in [d]\} \longrightarrow \forall \mathbf{A} \in \mathbb{R}^{d \times d} \hookrightarrow \mathbf{A} = \sum_{j,l} h_{j,l}^i(\mathbf{A}) \mathbf{B}_i^{jl}$

Define:  $h^i(\mathbf{A}) \in \mathbb{R}^{d \times d} : h^i(\mathbf{A})_{j,l} = h_{j,l}^i(\mathbf{A})$

## Wish list:

- $\mathbf{H}_i^k \rightarrow \nabla^2 f_i(x^*)$  as  $k \rightarrow \infty$
- Local rate independent of the condition number

# Basis Learn: Idea

Fix basis:  $\{\mathbf{B}_i^{jl} : j, l \in [d]\} \longrightarrow \forall \mathbf{A} \in \mathbb{R}^{d \times d} \hookrightarrow \mathbf{A} = \sum_{j,l} h_{j,l}^i(\mathbf{A}) \mathbf{B}_i^{jl}$

Define:  $h^i(\mathbf{A}) \in \mathbb{R}^{d \times d} : h^i(\mathbf{A})_{j,l} = h_{j,l}^i(\mathbf{A})$

## Wish list:

- $\mathbf{H}_i^k \rightarrow \nabla^2 f_i(x^*)$  as  $k \rightarrow \infty \iff \mathbf{L}_i^k \rightarrow h^i(\nabla^2 f_i(x^*))$
- Local rate independent of the condition number

# Basis Learn: Idea

Fix basis:  $\{\mathbf{B}_i^{jl} : j, l \in [d]\} \longrightarrow \forall \mathbf{A} \in \mathbb{R}^{d \times d} \hookrightarrow \mathbf{A} = \sum_{j,l} h_{j,l}^i(\mathbf{A}) \mathbf{B}_i^{jl}$

Define:  $h^i(\mathbf{A}) \in \mathbb{R}^{d \times d} : h^i(\mathbf{A})_{j,l} = h_{j,l}^i(\mathbf{A})$

## Wish list:

- $\mathbf{H}_i^k \rightarrow \nabla^2 f_i(x^*)$  as  $k \rightarrow \infty \longleftrightarrow \mathbf{L}_i^k \rightarrow h^i(\nabla^2 f_i(x^*))$
- Local rate independent of the condition number

**Desire:**  
Communication-efficient  
learning mechanism



Rustem Islamov, Xun Qian and Peter Richtárik  
**Distributed Second Order Methods with Fast  
Rates and Compressed Communication,**  
ICML 2021.

# BL1: Basis Learn with Bidirectional Compression

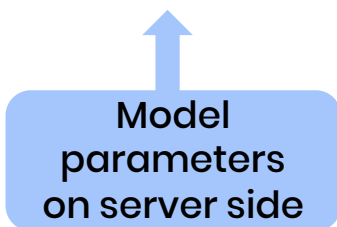
**Parameters:** Hessian learning rate  $\alpha \geq 0$ , model learning rate  $\eta \geq 0$ ,  
gradient compression probability  $p \in (0, 1]$



# BL1: Basis Learn with Bidirectional Compression

**Parameters:** Hessian learning rate  $\alpha \geq 0$ , model learning rate  $\eta \geq 0$ ,  
gradient compression probability  $p \in (0, 1]$

**Initialization:**  $x^0 = w^0 = z^0$



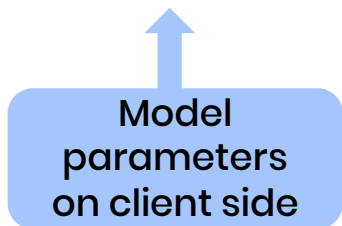
Model  
parameters  
on server side

A light blue rounded rectangle containing the text 'Model parameters on server side'. A light blue arrow points upwards from the top center of this rectangle towards the initialization equation  $x^0 = w^0 = z^0$  in the block above.

# BL1: Basis Learn with Bidirectional Compression

**Parameters:** Hessian learning rate  $\alpha \geq 0$ , model learning rate  $\eta \geq 0$ ,  
gradient compression probability  $p \in (0, 1]$


**Initialization:**  $x^0 = w^0 = z^0$



# BL1: Basis Learn with Bidirectional Compression

**Parameters:** Hessian learning rate  $\alpha \geq 0$ , model learning rate  $\eta \geq 0$ ,  
gradient compression probability  $p \in (0, 1]$

**Initialization:**  $x^0 = w^0 = z^0$



Model parameters  
when gradient was  
computed last time

# BL1: Basis Learn with Bidirectional Compression

**Parameters:** Hessian learning rate  $\alpha \geq 0$ , model learning rate  $\eta \geq 0$ ,  
gradient compression probability  $p \in (0, 1]$

**Initialization:**  $x^0 = w^0 = z^0$ ,  $\mathbf{L}_i^0 \in \mathbb{R}^{d \times d}$ ,  $\mathbf{H}_i^0 = \sum_{j,l} (\mathbf{L})_{j,l} \mathbf{B}_i^{j,l}$

# BL1: Basis Learn with Bidirectional Compression

**Parameters:** Hessian learning rate  $\alpha \geq 0$ , model learning rate  $\eta \geq 0$ ,  
gradient compression probability  $p \in (0, 1]$

**Initialization:**  $x^0 = w^0 = z^0$ ,  $\mathbf{L}_i^0 \in \mathbb{R}^{d \times d}$ ,  $\mathbf{H}_i^0 = \sum_{j,l} (\mathbf{L})_{j,l} \mathbf{B}_i^{j,l}$   
 $\mathbf{H}^0 = \frac{1}{n} \sum_{i=1}^n \mathbf{H}_i^0$ ,  $\xi^0 = 1$



If one, we compute gradients  
If zero, we use gradients from previous  
iteration

# Learning mechanism

$$\mathbf{L}_i^{k+1} = \mathbf{L}_i^k + \alpha \mathcal{C}_i^k (h^i(\nabla^2 f_i(z^k)) - \mathbf{L}_i^k)$$

Compressing the update inspired  
(by first-order method DIANA)



Konstantin Mishchenko, Eduard Gorbunov,  
Martin Takáč and Peter Richtárik,  
**Distributed learning with compressed  
gradient differences**, arXiv:1901.09269,  
2019.

# Learning mechanism

$$\mathbf{L}_i^{k+1} = \mathbf{L}_i^k + \alpha \mathcal{C}_i^k (h^i(\nabla^2 f_i(z^k)) - \mathbf{L}_i^k)$$

Compression operator

# Learning mechanism

$$\mathbf{L}_i^{k+1} = \mathbf{L}_i^k + \alpha \mathcal{C}_i^k (h^i(\nabla^2 f_i(z^k)) - \mathbf{L}_i^k)$$

Compression operator

**Contractive compressor**

$$\|\mathcal{C}(\mathbf{M})\|_F \leq \|\mathbf{M}\|_F$$

$$\|\mathcal{C}(\mathbf{M}) - \mathbf{M}\|_F^2 \leq (1 - \delta) \|\mathbf{M}\|_F^2 \quad \forall \mathbf{M} \in \mathbb{R}^{d \times d}$$

**Unbiased compressor**

$$\mathbb{E}[\mathcal{C}(\mathbf{M})] = \mathbf{M}$$

$$\mathbb{E} \left[ \|\mathcal{C}(\mathbf{M}) - \mathbf{M}\|_F^2 \right] \leq \omega \|\mathbf{M}\|_F^2 \quad \forall \mathbf{M} \in \mathbb{R}^{d \times d}$$



# Learning mechanism

$$\mathbf{L}_i^{k+1} = \mathbf{L}_i^k + \alpha \mathcal{C}_i^k (h^i(\nabla^2 f_i(z^k)) - \mathbf{L}_i^k)$$

Compression operator

**Contractive compressor**

$$\|\mathcal{C}(\mathbf{M})\|_F \leq \|\mathbf{M}\|_F$$

$$\|\mathcal{C}(\mathbf{M}) - \mathbf{M}\|_F^2 \leq (1 - \delta) \|\mathbf{M}\|_F^2 \quad \forall \mathbf{M} \in \mathbb{R}^{d \times d}$$

**Unbiased compressor**

$$\mathbb{E}[\mathcal{C}(\mathbf{M})] = \mathbf{M}$$

$$\mathbb{E} \left[ \|\mathcal{C}(\mathbf{M}) - \mathbf{M}\|_F^2 \right] \leq \omega \|\mathbf{M}\|_F^2 \quad \forall \mathbf{M} \in \mathbb{R}^{d \times d}$$

Don't need Error Feedback  
Mechanism

# Learning mechanism

$$\mathbf{L}_i^{k+1} = \mathbf{L}_i^k + \alpha C_i^k (h^i(\nabla^2 f_i(z^k)) - \mathbf{L}_i^k)$$



Stepsize depends only  
on the compression

# BL1: Algorithm. Client Side

**for** each device  $i = 1, \dots, n$  in parallel **do**

**if**  $\xi^k = 1$

$w^{k+1} = z^k$ , compute local gradient  $\nabla f_i(z^k)$  and send to the server

**if**  $\xi^k = 0$

$w^{k+1} = w^k$

  Compute local Hessian  $\nabla^2 f_i(z^k)$  and send  $\mathbf{S}_i^k := \mathcal{C}_i^k(h^i(\nabla^2 f_i(z^k))) - \mathbf{L}_i^k$  to the server

  Update local Hessian shifts  $\mathbf{L}_i^{k+1} = \mathbf{L}_i^k + \alpha \mathbf{S}_i^k$ ,  $\mathbf{H}_i^{k+1} = \mathbf{H}_i^k + \alpha \sum_{jl} (\mathbf{S}_i^k)_{jl} \mathbf{B}_i^{jl}$

**end for**

# BL1: Algorithm. Server Side

**on** server

**if**  $\xi^k = 1$

$$w^{k+1} = z^k, \quad g^k = \nabla f(z^k)$$

**if**  $\xi^k = 0$

$$w^{k+1} = w^k, \quad g^k = [\mathbf{H}^k]_{\mu} (z^k - w^k) + \nabla f(w^k)$$

$$x^{k+1} = z^k - [\mathbf{H}^k]_{\mu}^{-1} g^k$$

$$\mathbf{H}^{k+1} = \mathbf{H}^k + \frac{\alpha}{n} \sum_{i=1}^n \sum_{jl} (\mathbf{S}_i^k)_{jl} \mathbf{B}_i^{jl}$$

Send  $v^k := \mathcal{Q}^k(x^{k+1} - z^k)$  to all devices  $i \in [n]$

Update the model  $z^{k+1} = z^k + \eta v^k$

Send  $\xi^{k+1} \sim \text{Bernoulli}(p)$  to all devices  $i \in [n]$

**for** each device  $i = 1, \dots, n$  in parallel **do**

Update the model  $z^{k+1} = z^k + \eta v^k$

# Basis Learn: Assumptions

**Assumption 4.3.** (i)  $\mathcal{Q}^k$  ( $\mathcal{Q}_i^k$ ) is an unbiased compressor with parameter  $\omega_M$  and  $0 < \eta \leq 1/(\omega_M+1)$ . (ii) For all  $j \in [d]$ ,  $(z^k)_j$  in Algorithm 1 is a convex combination of  $\{(x^t)_j\}_{t=0}^k$  for  $k \geq 0$ .

**Assumption 4.4.** (i)  $\mathcal{Q}^k$  ( $\mathcal{Q}_i^k$ ) is a contraction compressor with parameter  $\delta_M$  and  $\eta = 1$ . (ii)  $\mathcal{Q}^k$  ( $\mathcal{Q}_i^k$ ) is deterministic, i.e.,  $\mathbb{E}[\mathcal{Q}^k(x)] = \mathcal{Q}^k(x)$  for any  $x \in \mathbb{R}^d$ .

**Assumption 4.5.** (i)  $\mathcal{C}_i^k$  is an unbiased compressor with parameter  $\omega$  and  $0 < \alpha \leq 1/(\omega+1)$ . (ii) For all  $i \in [n]$  and  $j, l \in [d]$ ,  $(\mathbf{L}_i^k)_{jl}$  is a convex combination of  $\{h^i(\nabla^2 f_i(z^t))_{jl}\}_{t=0}^k$  in Algorithm 1

**Assumption 4.6.** (i)  $\mathcal{C}_i^k$  is a contraction compressor with parameter  $\delta$  and  $\alpha = 1$ . (ii)  $\mathcal{C}_i^k$  is deterministic, i.e.,  $\mathbb{E}[\mathcal{C}_i^k(\mathbf{A})] = \mathcal{C}_i^k(\mathbf{A})$  for any  $\mathbf{A} \in \mathbb{R}^{d \times d}$ .

**Assumption 4.7.** We have  $\|\nabla^2 f_i(x) - \nabla^2 f_i(y)\| \leq H\|x - y\|$ ,  $\|\nabla^2 f_i(x) - \nabla^2 f_i(y)\|_F \leq H_1\|x - y\|$ ,  $\|h^i(\nabla^2 f_i(x)) - h^i(\nabla^2 f_i(y))\|_F \leq M_1\|x - y\|$ ,  $\max_{j,l}\{|h^i(\nabla^2 f_i(x))_{jl} - h^i(\nabla^2 f_i(y))_{jl}|\} \leq M_2\|x - y\|$ ,  $\max_{j,l}\{\|\mathbf{B}_i^{jl}\|_F\} \leq R$  for any  $x, y \in \mathbb{R}^d$  and  $i \in [n]$ .

# Local Convergence Theory

$$\mathbb{E}[\Phi_2^k] \leq \left(1 - \frac{\min\{4A, A_M\}}{4}\right)^k \Phi_2^0$$

$$\mathbf{E} \left[ \frac{\|x^{k+1} - x^*\|^2}{\|x^k - x^*\|^2} \right] \leq \left(1 - \frac{\min\{4A, A_M\}}{4}\right)^k \left( \frac{A_M H^2}{8B M_1^2 \mu^2} + \frac{2N_B R^2}{\mu^2} \right) \Phi_2^0$$

# Local Convergence Theory

$$\mathbb{E}[\Phi_2^k] \leq \left(1 - \frac{\min\{4A, A_M\}}{4}\right)^k \Phi_2^0$$

Local linear rate

$$\mathbf{E} \left[ \frac{\|x^{k+1} - x^*\|^2}{\|x^k - x^*\|^2} \right] \leq \left(1 - \frac{\min\{4A, A_M\}}{4}\right)^k \left( \frac{A_M H^2}{8B M_1^2 \mu^2} + \frac{2N_B R^2}{\mu^2} \right) \Phi_2^0$$

Local superlinear rate

# Local Convergence Theory

Lyapunov function

$$\Phi_2^k := \frac{1}{n} \sum_{i=1}^n \|\mathbf{L}_i^k - \mathbf{L}_i^*\|_F^2 + \frac{4BM_1^2}{A_M} \|x^k - x^*\|^2$$

$$\mathbb{E}[\Phi_2^k] \leq \left(1 - \frac{\min\{4A, A_M\}}{4}\right)^k \Phi_2^0$$

Local linear rate

$$\mathbf{E} \left[ \frac{\|x^{k+1} - x^*\|^2}{\|x^k - x^*\|^2} \right] \leq \left(1 - \frac{\min\{4A, A_M\}}{4}\right)^k \left( \frac{A_M H^2}{8BM_1^2 \mu^2} + \frac{2N_B R^2}{\mu^2} \right) \Phi_2^0$$

Local superlinear rate



# Local Convergence Theory

Lyapunov function

$$\Phi_2^k := \frac{1}{n} \sum_{i=1}^n \|\mathbf{L}_i^k - \mathbf{L}_i^*\|_F^2 + \frac{4BM_1^2}{A_M} \|x^k - x^*\|^2$$

$$\mathbb{E}[\Phi_2^k] \leq \left(1 - \frac{\min\{4A, A_M\}}{4}\right)^k \Phi_2^0$$

Local linear rate

Constants depending on  
the choice of the  
compressor and stepsize

$$A_M = \begin{cases} \eta & \text{if Asm. 4.3(i) holds} \\ \frac{\delta_M}{4} & \text{if Asm. 4.4(i) holds} \end{cases}$$

$$A = \begin{cases} \alpha & \text{if Asm. 4.5(i) holds} \\ \frac{\delta}{4} & \text{if Asm. 4.6(i) holds} \end{cases}$$

$$\mathbb{E} \left[ \frac{\|x^{k+1} - x^*\|^2}{\|x^k - x^*\|^2} \right] \leq \left(1 - \frac{\min\{4A, A_M\}}{4}\right)^k \left( \frac{A_M H^2}{8BM_1^2 \mu^2} + \frac{2N_B R^2}{\mu^2} \right) \Phi_2^0$$

Local superlinear rate

# Local Convergence Theory

Lyapunov function

$$\Phi_2^k := \frac{1}{n} \sum_{i=1}^n \|\mathbf{L}_i^k - \mathbf{L}_i^*\|_F^2 + \frac{4BM_1^2}{A_M} \|x^k - x^*\|^2$$

Provably learn the Hessian  
at the optimum

$$\mathbb{E}[\Phi_2^k] \leq \left(1 - \frac{\min\{4A, A_M\}}{4}\right)^k \Phi_2^0$$

Local linear rate

Constants depending on  
the choice of the  
compressor and stepsize

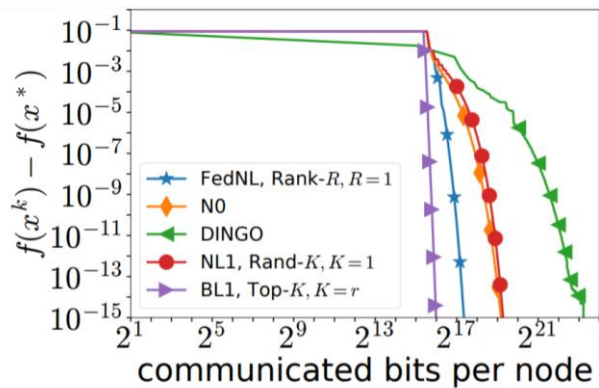
$$A_M = \begin{cases} \eta & \text{if Asm. 4.3(i) holds} \\ \frac{\delta_M}{4} & \text{if Asm. 4.4(i) holds} \end{cases}$$

$$A = \begin{cases} \alpha & \text{if Asm. 4.5(i) holds} \\ \frac{\delta}{4} & \text{if Asm. 4.6(i) holds} \end{cases}$$

$$\mathbb{E} \left[ \frac{\|x^{k+1} - x^*\|^2}{\|x^k - x^*\|^2} \right] \leq \left(1 - \frac{\min\{4A, A_M\}}{4}\right)^k \left( \frac{A_M H^2}{8BM_1^2 \mu^2} + \frac{2N_B R^2}{\mu^2} \right) \Phi_2^0$$

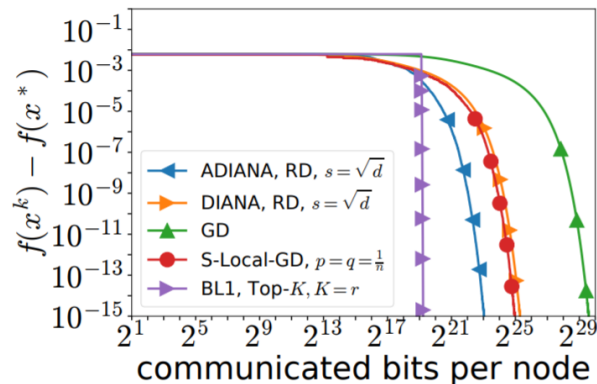
Local superlinear rate

# Experiments: Logistic Regression



(a) **covtype**,  $\lambda = 10^{-3}$

$$r \sim 0.44d$$



(b) **w2a**,  $\lambda = 10^{-4}$

$$r \sim 0.2d$$

# The End

For more details:



Xun Qian, Rustem Islamov, Mher Safaryan, and Peter Richtárik

**Basis Matters: Better Communication-Efficient Second Order Methods for Federated Learning**

arXiv preprint arXiv: 2106.02969, 2021