

# GradSkip: Communication-Accelerated Local Gradient Methods with Better Computational Complexity

**Artavazd Maranjyan**

MATHEMATICS IN ARMENIA: ADVANCES AND PERSPECTIVES

July 5, 2023

GradSkip: Communication-Accelerated  
Local Gradient Methods with  
Better Computational Complexity

Artavazd Maranjyan<sup>\*</sup> Mher Safaryan KAUST Saudi Arabia Peter Richtárik KAUST Saudi Arabia

Abstract

We study a class of distributed optimization algorithms that aim to alleviate high communication costs by allowing the clients to perform multiple local gradient-type training steps prior to communication. While methods of this type have been studied for about a decade, the empirically observed acceleration properties of local training eluded all attempts at theoretical understanding. In a recent breakthrough, Moshchuk et al. (ICML 2022) provided this local training, when properly executed, leads to provable communication acceleration, and this holds in the strongly convex regime without relying on any data similarity assumptions. However, their method, ProxDrop, requires all clients to take the same number of local training steps in each communication round. Inspired by a convex set intuition, we start our investigation by conjecturing that clients with “less important” data should be able to get away with fewer local training steps without this impacting the overall communication complexity of the method. It turns out that this intuition is correct: we managed to redesign the original ProxDrop method to achieve this. In particular, we prove that our modified method, for which only the name GradSkip, converges linearly under the same assumptions, has the same accelerated communication complexity, while the number of local gradient steps can be reduced relative to a local condition number. We further generalize our method by extending the randomness of probabilistic alternations to arbitrary unbiased compression operators and considering a generic provable regularization. This generalization, which we call GradSkip+, recovers several related methods in the literature as special cases. Finally, we present an empirical study on carefully designed key problems that confirm our theoretical claims.

## 1 Introduction

Federated Learning (FL) is an emerging distributed machine learning paradigm where diverse data holders or clients (e.g., smart watches, mobile devices, laptops, hospitals) collectively aim to train a single machine learning model without revealing local data to each other or the coordinating central server (McMahan et al., 2017; Karouze et al., 2019; Wang, 2021). Training such models amounts to solving federated optimization problems of the form

$$\min_{\theta} \left\{ f(\theta) = \frac{1}{n} \sum_{i=1}^n f_i(\theta) \right\} \quad (1)$$

<sup>\*</sup>The work of Artavazd Maranjyan was supported during a summer research fellowship in the Optimization and Machine Learning Lab at KAUST led by Peter Richtárik. Artavazd Maranjyan is a Machine Learning researcher at Yerevan State University and a Master's student at Yerevan State University, Armenia.  
<sup>†</sup>KAUST = King Abdullah University of Science and Technology.



**Artavazd Maranjyan**, Mher Safaryan, Peter Richtárik  
**GradSkip: Communication-Accelerated Local Gradient Methods with Better Computational Complexity**

*arXiv:2210.16402, 2022*

# Co-authors



**Mher Safaryan**



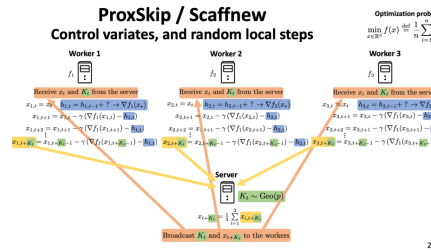
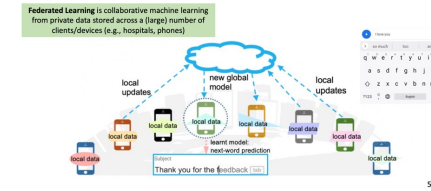
**Peter Richtárik**



# Outline of the Talk

1. What is Federated Learning?
2. What is Local Training?
3. The ProxSkip Algorithm
4. GradSkip: Algorithm
5. GradSkip: Theory
6. GradSkip: Experiments

## The First Federated Learning App: Next-Word Prediction



## GradSkip: Computational Complexity

Expected # of local steps between 2 communication =  $\frac{n_i(1+\sqrt{n_{\max}})}{n_i+\sqrt{n_{\max}}} \leq \min\{\frac{n_i}{2}, \sqrt{n_{\max}}\}$

Worker 1:  $n_1 = 10$ , Worker 2:  $n_2 = 100$ , Worker 3:  $n_3 = 1,000,000$

50 local steps, 100 local steps, 1000 local steps

ProxSkip:  $n\sqrt{n_{\max}} = 3000$

GradSkip:  $\sum_{i=1}^n \frac{n_i(1+\sqrt{n_{\max}})}{n_i+\sqrt{n_{\max}}} = 1110$

## What does Local Training do?

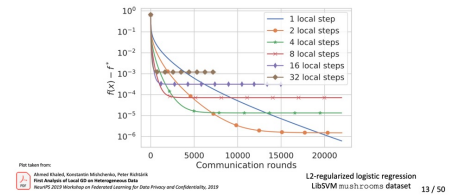


Figure 1: L2-regularized logistic regression on LIBSVM multi-class dataset. Source: [1]

## Key theoretical technique

Use random control variate

ProxSkip:  $x_{i,t+1} = x_{i,t} - \gamma(\nabla f_i(x_{i,t}) - \hat{h}_{i,t})$

with probability  $q_i$  do  $\hat{h}_{i,t+1} = h_{i,t}$

with probability  $1 - q_i$  do  $\hat{h}_{i,t+1} = \nabla f_i(x_{i,t})$

$x_{i,t+1} = x_{i,t} - \gamma(\nabla f_i(x_{i,t}) - \hat{h}_{i,t+1})$

## Large maximum smoothness constant

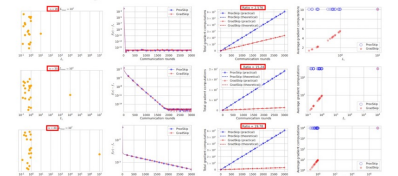


Figure 1: In the first column we show the smoothness constants for devices. In the second column we show the convergence per communication rounds. In the third column we show theoretical and practical difference between number of gradient computations. In the last column we have the average gradient computations for each device having  $L_i$  smoothness. We see that for GradSkip the device with  $L_i = L_{\max}$  does the same number of gradient computations as devices in ProxSkip. Source: [1]

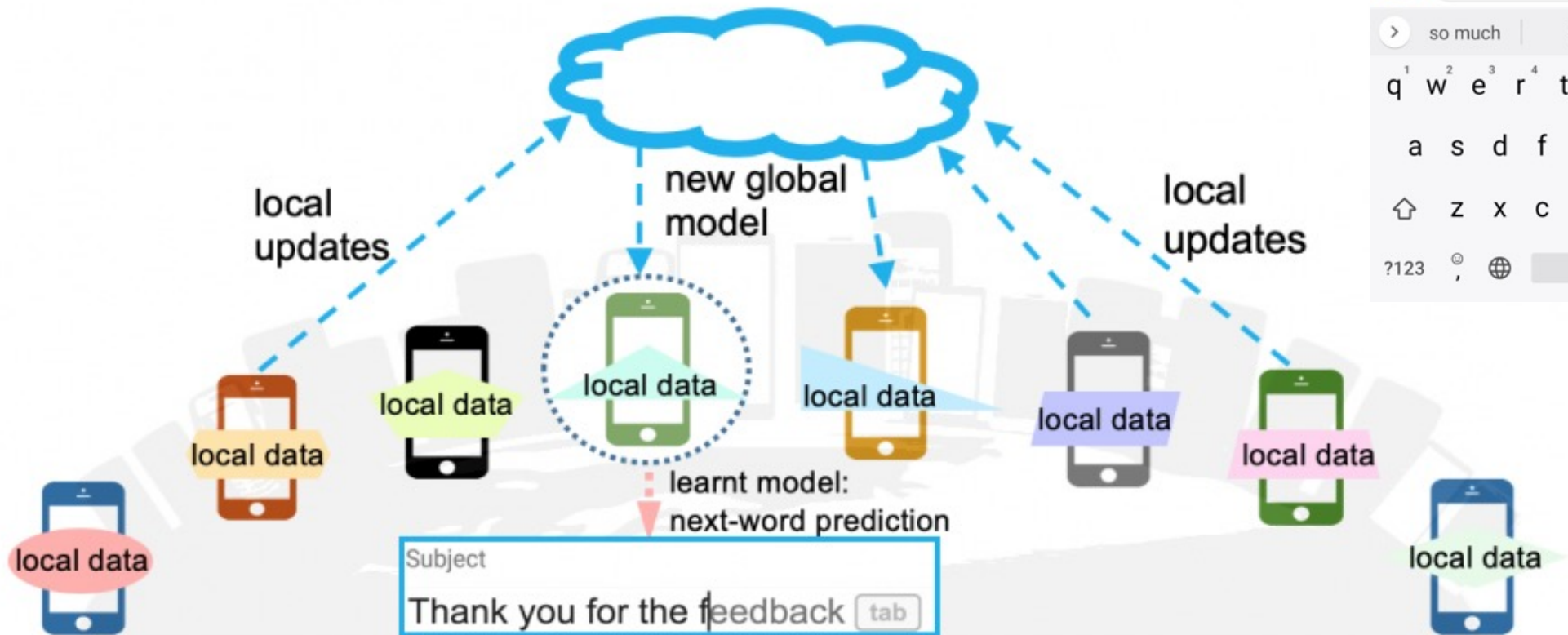


# **Part 1**

# **What is Federated Learning?**

# The First Federated Learning App: Next-Word Prediction

**Federated Learning** is collaborative machine learning from private data stored across a (large) number of clients/devices (e.g., hospitals, phones)



# Optimization Formulation of Federated Learning

$$\min_{x \in \mathbb{R}^d} f(x)$$

$\stackrel{\text{def}}{=}$

$$\frac{1}{n} \sum_{i=1}^n f_i(x)$$

# devices /  
machines

# model parameters / features

Loss on local data  $\mathcal{D}_i$  stored on device  $i$

$$f_i(x) = \mathbb{E}_{\xi \sim \mathcal{D}_i} f_{i,\xi}(x)$$

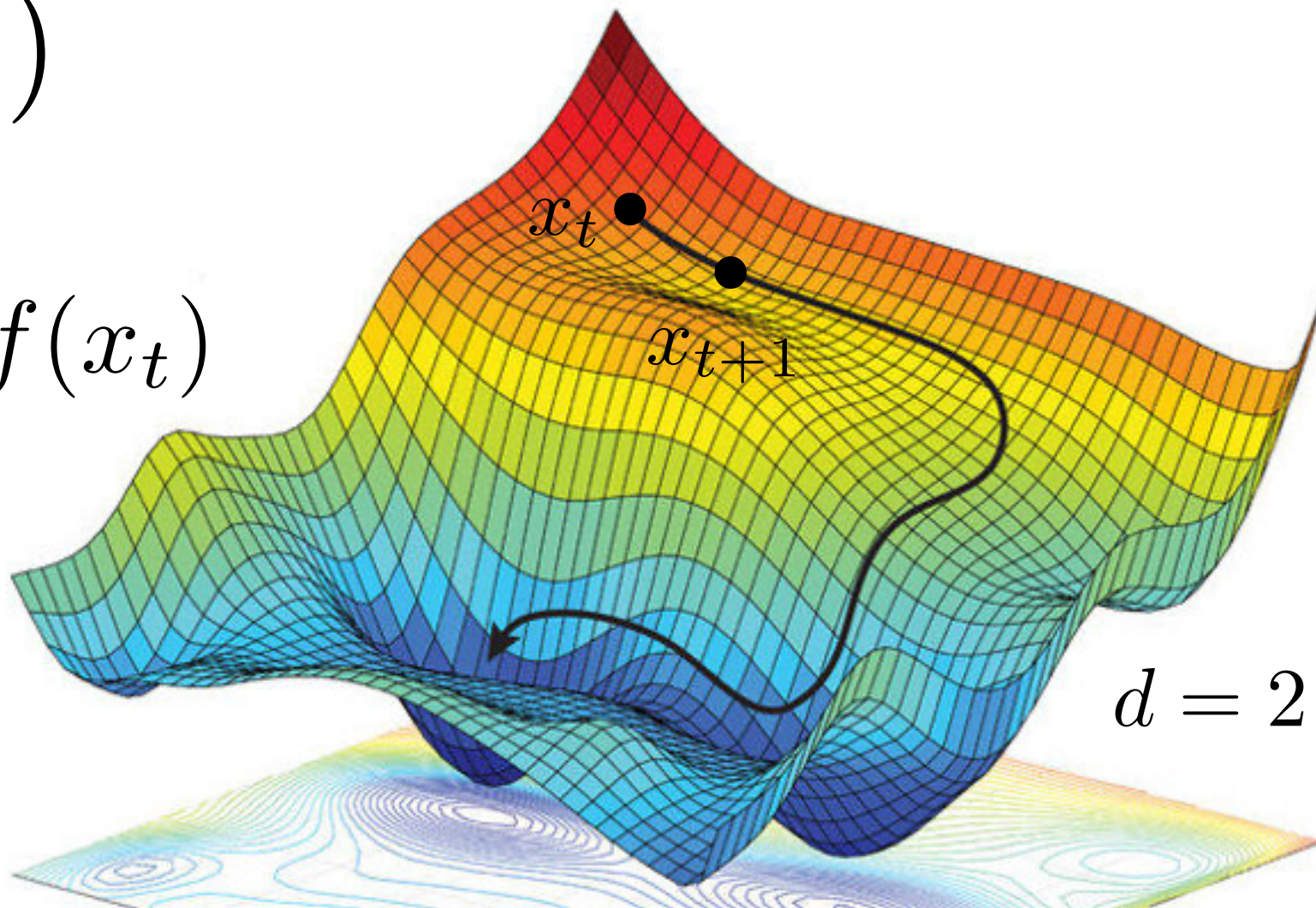
The datasets  $\mathcal{D}_1, \dots, \mathcal{D}_n$  can be arbitrarily heterogeneous

# Gradient Descent

$$\min_{x \in \mathbb{R}^d} f(x)$$

$$x_{t+1} = x_t - \gamma \nabla f(x_t)$$

Stepsize / Learning rate



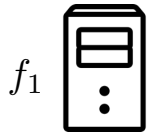
# Distributed Gradient Descent

(Each worker performs 1 GD step using its local function, and the results are averaged)

Optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n f_i(x)$$

Worker 1



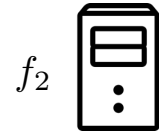
Receive  $x_t$  from the server

$$x_{1,t} = x_t$$

$$x_{1,t+1} = x_{1,t} - \gamma \nabla f_1(x_{1,t})$$

$d$ -dimensional vector  
computed by machine 1

Worker 2

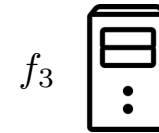


Receive  $x_t$  from the server

$$x_{2,t} = x_t$$

$$x_{2,t+1} = x_{2,t} - \gamma \nabla f_2(x_{2,t})$$

Worker 3



Receive  $x_t$  from the server

$$x_{3,t} = x_t$$

$$x_{3,t+1} = x_{3,t} - \gamma \nabla f_3(x_{3,t})$$

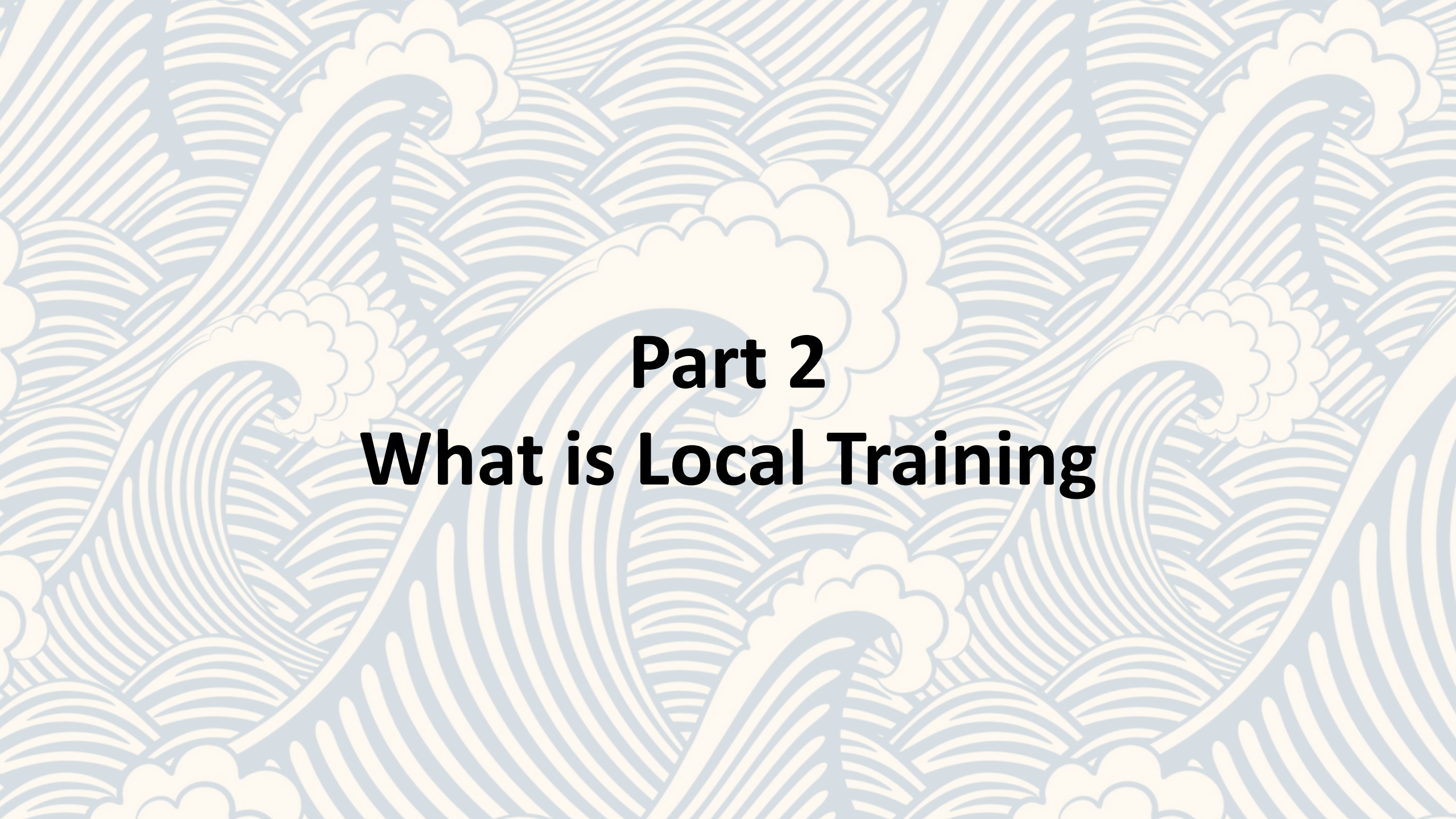
Server



$$x_{t+1} = \frac{1}{3} \sum_{i=1}^3 x_{i,t+1}$$

Broadcast  $x_{t+1}$  to the workers





# **Part 2**

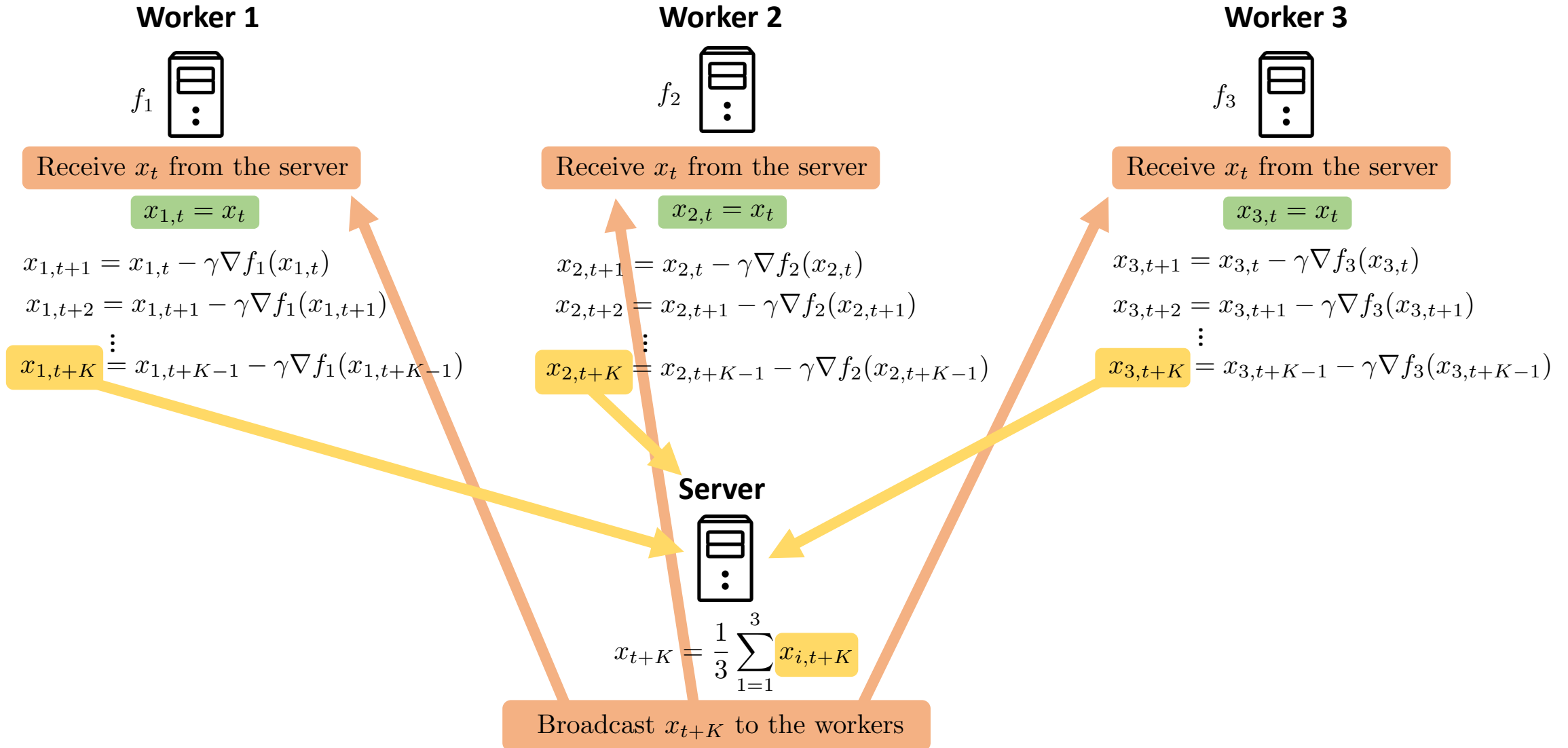
## **What is Local Training**

# Distributed Local Gradient Descent

(Each worker performs  $K$  GD steps using its local function, and the results are averaged)

Optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n f_i(x)$$



# Brief History of Local Training Methods

Table 1: Five generations of local training (LT) methods summarizing the progress made by the ML/FL community over the span of 7+ years in the understanding of the *communication acceleration properties of LT*.

Generation <sup>(a)</sup>	Theory	Assumptions	Comm. Complexity <sup>(b)</sup>	Selected Key References
1. Heuristic	✗	—	empirical results only	LocalSGD [Povey et al., 2015]
	✗	—	empirical results only	SparkNet [Moritz et al., 2016]
	✗	—	empirical results only	FedAvg [McMahan et al., 2017]
2. Homogeneous	✓	bounded gradients	sublinear	FedAvg [Li et al., 2020b]
	✓	bounded grad. diversity <sup>(c)</sup>	linear but worse than GD	LFGD [Haddadpour and Mahdavi, 2019]
3. Sublinear	✓	standard <sup>(d)</sup>	sublinear	LGD [Khaled et al., 2019]
	✓	standard	sublinear	LSGD [Khaled et al., 2020]
4. Linear	✓	standard	linear but worse than GD	Scaffold [Karimireddy et al., 2020]
	✓	standard	linear but worse than GD	S-Local-GD [Gorbunov et al., 2020a]
	✓	standard	linear but worse than GD	FedLin [Mitra et al., 2021]
5. Accelerated	✓	standard	linear & better than GD	ProxSkip/Scaffnew [Mishchenko et al., 2022]
	✓	standard	linear & better than GD	ProxSkip-VR

<sup>(a)</sup> Since client sampling (CS) and data sampling (DS) can only *worsen* theoretical communication complexity, our historical breakdown of the literature into 5 generations of LT methods focuses on the full client participation (i.e., no CS) and exact local gradient (i.e., no DS) setting. While some of the referenced methods incorporate CS and DS techniques, these are irrelevant for our purposes. Indeed, from the viewpoint of communication complexity, all these algorithms enjoy best theoretical performance in the no-CS and no-DS regime.

<sup>(b)</sup> For the purposes of this table, we consider problem (1) in the *smooth* and *strongly convex* regime only. This is because the literature on LT methods struggles to understand even in this simplest (from the point of view of optimization) regime.

<sup>(c)</sup> *Bounded gradient diversity* is a uniform bound on a specific notion of gradient variance depending on client sampling probabilities. However, this assumption (as all homogeneity assumptions) is very restrictive. For example, it is not satisfied the standard class of smooth and strongly convex functions.

<sup>(d)</sup> The notorious FL challenge of handling non-i.i.d. data by LT methods was solved by Khaled et al. [2019] (from the viewpoint of *optimization*). From generation 3 onwards, there was no need to invoke any data/gradient homogeneity assumptions. Handling non-i.i.d. data remains a challenge from the point of view of *generalization*, typically by considering *personalized FL* models.



Grigory Malinovsky, Kai Yi, Peter Richtárik

Variance Reduced ProxSkip: Algorithm, Theory and Application to Federated Learning

NeurIPS 2022

# Why treat all devices equally?

(Each worker performs  $K$  GD steps using its local function, and the results are averaged)

Optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n f_i(x)$$

**Worker 1**



Receive  $x_t$  from the server

$$x_{1,t} = x_t$$

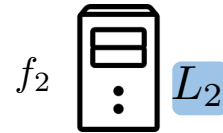
$$x_{1,t+1} = x_{1,t} - \gamma \nabla f_1(x_{1,t})$$

$$x_{1,t+2} = x_{1,t+1} - \gamma \nabla f_1(x_{1,t+1})$$

$\vdots$

$$x_{1,t+K} = x_{1,t+K-1} - \gamma \nabla f_1(x_{1,t+K-1})$$

**Worker 2**



Receive  $x_t$  from the server

$$x_{2,t} = x_t$$

$$x_{2,t+1} = x_{2,t} - \gamma \nabla f_2(x_{2,t})$$

$$x_{2,t+2} = x_{2,t+1} - \gamma \nabla f_2(x_{2,t+1})$$

$\vdots$

$$x_{2,t+K} = x_{2,t+K-1} - \gamma \nabla f_2(x_{2,t+K-1})$$

**Worker 3**



Receive  $x_t$  from the server

$$x_{3,t} = x_t$$

$$x_{3,t+1} = x_{3,t} - \gamma \nabla f_3(x_{3,t})$$

$$x_{3,t+2} = x_{3,t+1} - \gamma \nabla f_3(x_{3,t+1})$$

$\vdots$

$$x_{3,t+K} = x_{3,t+K-1} - \gamma \nabla f_3(x_{3,t+K-1})$$

**Server**



$$x_{t+K} = \frac{1}{3} \sum_{i=1}^3 x_{i,t+K}$$

Broadcast  $x_{t+K}$  to the workers



# Key insight

**GradSkip** = **ProxSkip** + **Heterogeneity Awareness**

Algorithm	Communication Complexity	Computational Complexity
<b>ProxSkip</b>	Accelerated (100 communications)	1000 GD steps per client
<b>GradSkip</b>	Accelerated (100 communications)	10,100, ... , 1000 GD steps



**Part 3**  
**The ProxSkip Algorithm**



Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich, Peter Richtárik  
**ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication  
Acceleration! Finally!** *International Conference on Machine Learning  
(ICML), 2022*

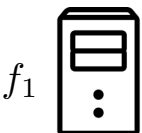
# ProxSkip / Scaffnew

## Control variates, and random local steps

Optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n f_i(x)$$

Worker 1



$f_1$

Worker 2



$f_2$

Worker 3



$f_3$

Receive  $x_t$  and  $K_t$  from the server

$$x_{1,t} = x_t \quad h_{1,t} = h_{1,t-1} + ? \rightarrow \nabla f_1(x_*)$$

$$x_{1,t+1} = x_{1,t} - \gamma (\nabla f_1(x_{1,t}) - h_{1,t})$$

$$x_{1,t+2} = x_{1,t+1} - \gamma (\nabla f_1(x_{1,t+1}) - h_{1,t})$$

$\vdots$

$$x_{1,t+K_t} = x_{1,t+K_t-1} - \gamma (\nabla f_1(x_{1,t+K_t-1}) - h_{1,t})$$

Receive  $x_t$  and  $K_t$  from the server

$$x_{2,t} = x_t \quad h_{2,t} = h_{2,t-1} + ? \rightarrow \nabla f_2(x_*)$$

$$x_{2,t+1} = x_{2,t} - \gamma (\nabla f_1(x_{2,t}) - h_{2,t})$$

$$x_{2,t+2} = x_{2,t+1} - \gamma (\nabla f_1(x_{2,t+1}) - h_{2,t})$$

$\vdots$

$$x_{2,t+K_t} = x_{2,t+K_t-1} - \gamma (\nabla f_1(x_{2,t+K_t-1}) - h_{2,t})$$

Receive  $x_t$  and  $K_t$  from the server

$$x_{3,t} = x_t \quad h_{3,t} = h_{3,t-1} + ? \rightarrow \nabla f_3(x_*)$$

$$x_{3,t+1} = x_{3,t} - \gamma (\nabla f_1(x_{3,t}) - h_{3,t})$$

$$x_{3,t+2} = x_{3,t+1} - \gamma (\nabla f_1(x_{3,t+1}) - h_{3,t})$$

$\vdots$

$$x_{3,t+K_t} = x_{3,t+K_t-1} - \gamma (\nabla f_1(x_{3,t+K_t-1}) - h_{3,t})$$

Server



$K_t \sim \text{Geo}(p)$

$$x_{t+K_t} = \frac{1}{3} \sum_{i=1}^3 x_{i,t+K_t}$$

Broadcast  $K_t$  and  $x_{t+K_t}$  to the workers





**Part 4**  
**The GradSkip Algorithm**

# GradSkip

Let workers decide how much to work

Optimization problem:

$$\min_{x \in \mathbb{R}^d} f(x) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^n f_i(x)$$

Worker 1



Worker 2



Worker 3



Receive  $x_t$  and  $K_t$  from the server

$$x_{1,t} = x_t \quad h_{1,t} = h_{1,t-1} + ? \rightarrow \nabla f_1(x_*)$$

$$M_{1,t} \sim \text{Geo}(q_1) \quad K_{1,t} = \min\{M_{1,t}, K_t\}$$

$$x_{1,t+1} = x_{1,t} - \gamma (\nabla f_1(x_{1,t}) - h_{1,t})$$

$$x_{1,t+2} = x_{1,t+1} - \gamma (\nabla f_1(x_{1,t+1}) - h_{1,t})$$

⋮

⋮

$$x_{1,t+K_{1,t}} = x_{1,t+K_{1,t}-1} - \gamma (\nabla f_1(x_{1,t+K_{1,t}-1}) - h_{1,t})$$

Receive  $x_t$  and  $K_t$  from the server

$$x_{2,t} = x_t \quad h_{2,t} = h_{2,t-1} + ? \rightarrow \nabla f_2(x_*)$$

$$M_{2,t} \sim \text{Geo}(q_2) \quad K_{2,t} = \min\{M_{2,t}, K_t\}$$

$$x_{2,t+1} = x_{2,t} - \gamma (\nabla f_2(x_{2,t}) - h_{2,t})$$

$$x_{2,t+2} = x_{2,t+1} - \gamma (\nabla f_2(x_{2,t+1}) - h_{2,t})$$

⋮

$$x_{2,t+K_{2,t}} = x_{2,t+K_{2,t}-1} - \gamma (\nabla f_2(x_{2,t+K_{2,t}-1}) - h_{2,t})$$

Server



$$K_t \sim \text{Geo}(p)$$

$$x_{t+K_t} = \frac{1}{3} \sum_{i=1}^3 x_{i,t+K_t}$$

Broadcast  $K_t$  and  $x_{t+K_t}$  to the workers

Receive  $x_t$  and  $K_t$  from the server

$$x_{3,t} = x_t \quad h_{3,t} = h_{3,t-1} + ? \rightarrow \nabla f_3(x_*)$$

$$M_{3,t} \sim \text{Geo}(q_3) \quad K_{3,t} = \min\{M_{3,t}, K_t\}$$

$$x_{3,t+1} = x_{3,t} - \gamma (\nabla f_3(x_{3,t}) - h_{3,t})$$

$$x_{3,t+2} = x_{3,t+1} - \gamma (\nabla f_3(x_{3,t+1}) - h_{3,t})$$

⋮

⋮

⋮

$$x_{3,t+K_{3,t}} = x_{3,t+K_{3,t}-1} - \gamma (\nabla f_3(x_{3,t+K_{3,t}-1}) - h_{3,t})$$

# Key theoretical technique

Use **random** control variate

ProxSkip:  $x_{i,t+1} = x_{i,t} - \gamma (\nabla f_i(x_{i,t}) - h_{i,t})$

with probability  $1 - q_i$  do

$$\hat{h}_{i,t+1} = h_{i,t}$$

with probability  $q_i$  do

$$\hat{h}_{i,t+1} = \nabla f_i(x_{i,t})$$

$$x_{i,t+1} = x_{i,t} - \gamma \left( \nabla f_i(x_{i,t}) - \hat{h}_{i,t+1} \right)$$



**Part 5**  
**GradSkip Theory**

# GradSkip: Assumptions same as in ProxSkip

## Assumptions:

$$\|\nabla f_i(x) - \nabla f_i(y)\| \leq L_i \|x - y\|$$

$$\langle \nabla f_i(x) - \nabla f_i(y), x - y \rangle \geq \mu \|x - y\|^2$$

# GradSkip: Bounding the # of Iterations

**Theorem:**

$$\gamma \leq \min_i \left\{ \frac{1}{L_i} \frac{p^2}{p^2 + q_i(1-p^2)} \right\}$$

$$t \geq \max \left\{ \frac{1}{\gamma \mu}, \frac{1}{p^2 - q_{\min}(1-p^2)} \right\} \log \frac{1}{\varepsilon} \Rightarrow \mathbb{E} [\Psi_t] \leq \varepsilon \Psi_0$$

# iterations

Lyapunov function:

$$\Psi_t := \sum_{i=1}^n \|x_{i,t} - x_{\star}\|^2 + \frac{\gamma^2}{p^2} \sum_{i=1}^n \|h_{i,t} - h_{i,\star}\|^2$$

# GradSkip: Optimal Probabilities

$$\begin{aligned}
 \kappa_i &= \frac{L_i}{\mu} \\
 \kappa_{\max} &= \frac{L_{\max}}{\mu} \\
 q_i &= \frac{\frac{1}{\kappa_i} - \frac{1}{\kappa_{\max}}}{1 - \frac{1}{\kappa_{\max}}} \\
 p^2 &= \frac{1}{\kappa_{\max}}
 \end{aligned}$$

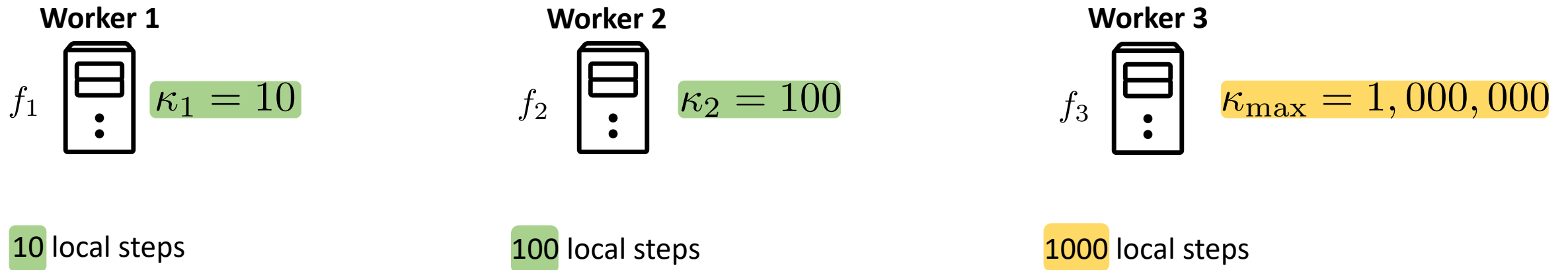
$$\gamma \leq \min_i \left\{ \frac{1}{L_i} \frac{p^2}{p^2 + q_i(1 - p^2)} \right\} = \frac{1}{L_{\max}}$$

$$t \geq \max \left\{ \frac{1}{\gamma \mu}, \frac{1}{p^2 - q_{\min}(1 - p^2)} \right\} \log \frac{1}{\varepsilon} = \kappa_{\max} \log \frac{1}{\varepsilon} \Rightarrow \mathbb{E} [\Psi_t] \leq \varepsilon \Psi_0$$

$$q_{\min} = 0$$

# GradSkip: Computational Complexity

Expected # of local steps between 2 communication =  $\frac{\kappa_i(1+\sqrt{\kappa_{\max}})}{\kappa_i+\sqrt{\kappa_{\max}}} \leq \min \{ \kappa_i, \sqrt{\kappa_{\max}} \}$



ProxSkip:  $n\sqrt{\kappa_{\max}} = 3000$

GradSkip:  $\sum_{i=1}^n \frac{\kappa_i(1+\sqrt{\kappa_{\max}})}{\kappa_i+\sqrt{\kappa_{\max}}} = 1110$

$$\frac{n\sqrt{\kappa_{\max}}}{\sum_{i=1}^n \frac{\kappa_i(1+\sqrt{\kappa_{\max}})}{\kappa_i+\sqrt{\kappa_{\max}}}} \xrightarrow{\kappa_{\max} \rightarrow \infty} n$$



# GradSkip vs ProxSkip

	GradSkip	ProxSkip
Number of iterations	$\kappa_{\max} \log \frac{1}{\varepsilon}$	$\kappa_{\max} \log \frac{1}{\varepsilon}$
Expected number of communications	$\sqrt{\kappa_{\max}} \log \frac{1}{\varepsilon}$	$\sqrt{\kappa_{\max}} \log \frac{1}{\varepsilon}$
Expected number of local steps between two communications	$\frac{\kappa_i (1 + \sqrt{\kappa_{\max}})}{\kappa_i + \sqrt{\kappa_{\max}}} \leq \min \{ \kappa_i, \sqrt{\kappa_{\max}} \}$	$\sqrt{\kappa_{\max}}$
Expected number of local steps	$\sum_{i=1}^n \frac{\kappa_i (1 + \sqrt{\kappa_{\max}})}{\kappa_i + \sqrt{\kappa_{\max}}} \approx \sqrt{\kappa_{\max}}$	$n \sqrt{\kappa_{\max}}$



# **Part 6**

## **Experiments**

# Experimental setup

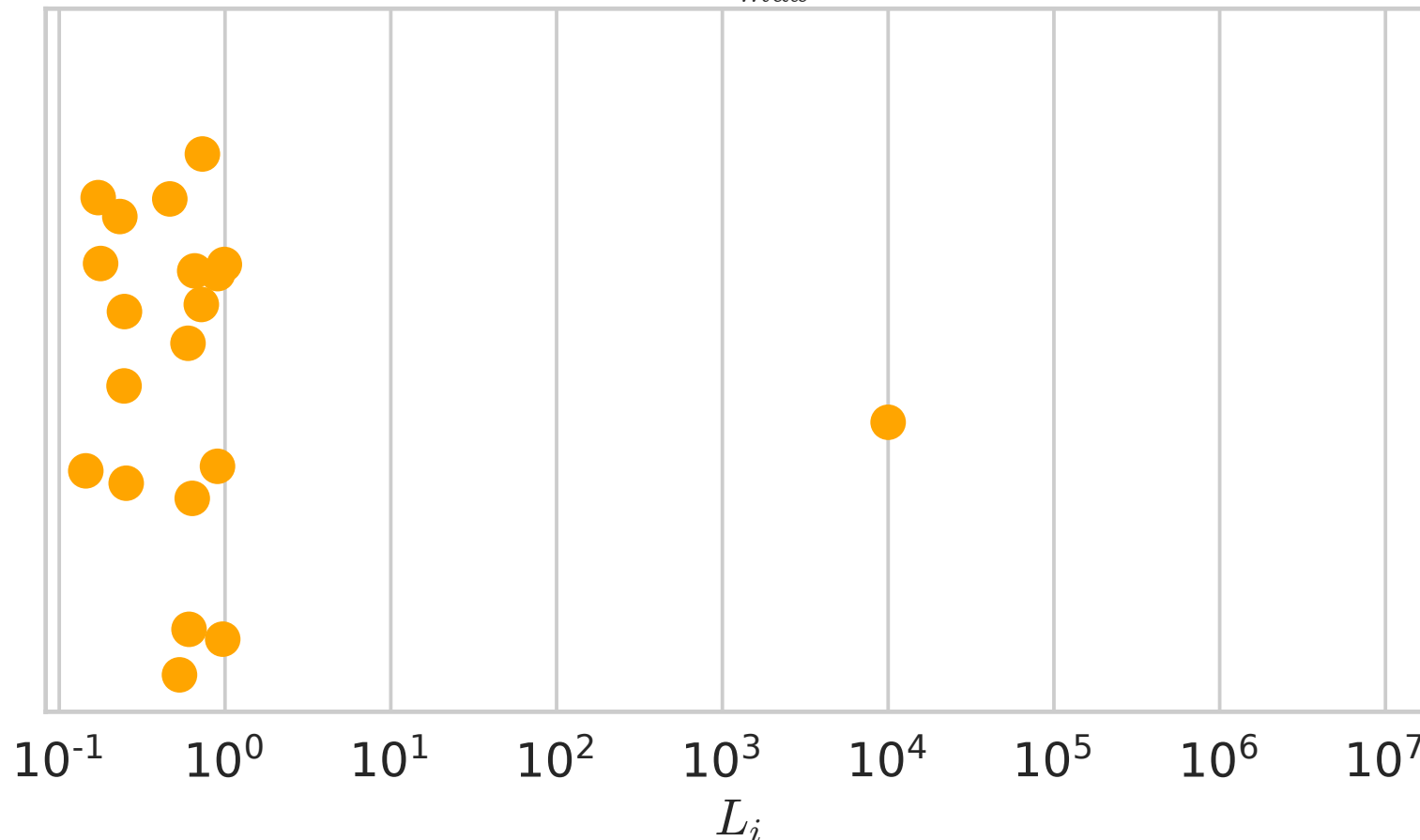
**L2-regularized logistic regression:**

$$f(x) = \frac{1}{n} \sum_{i=1}^n \log(1 + \exp(-b_i a_i^\top x)) + \frac{\lambda}{2} \|x\|^2$$

$$b_i \in \{-1, +1\}, \lambda = 0.1,$$

$$\mathbf{A}_i = \mathbf{U}_i \mathbf{L}_i \mathbf{V}_i \in \mathbb{R}^{200,300}, \sigma_{max}(\mathbf{A}_i) = L_i$$

$n = 20, L_{max} = 10^4$



# Large maximum smoothness constant

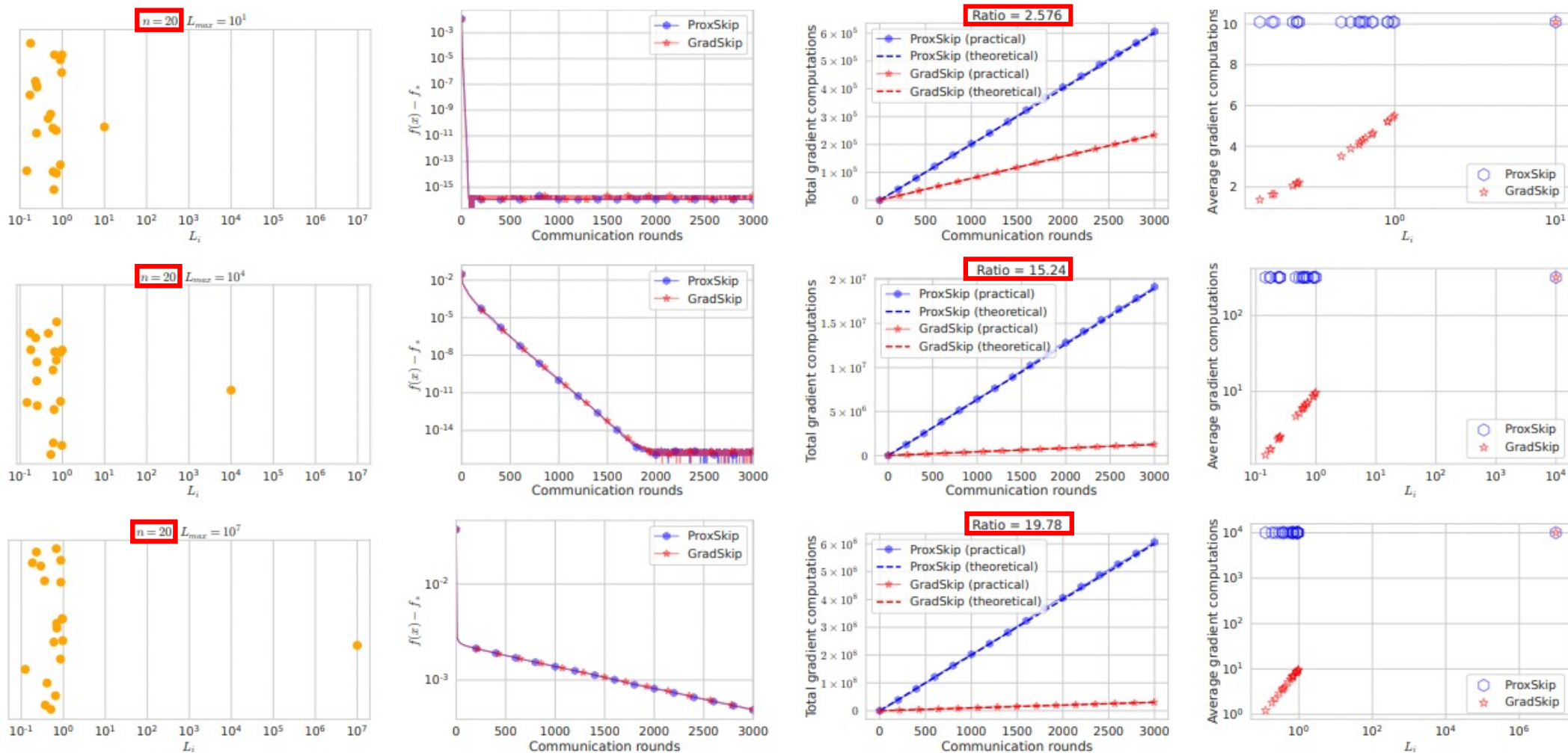


Figure 1: In the first column we show the smoothness constants for devices. In the second column we show the convergence per communication rounds. In the third column we show theoretical and practical difference between number of gradient computations. In the last column we have the average gradient computations for each device having  $L_i$  smoothness, we see that for **GradSkip** the device with  $L_i = L_{max}$  does the same number of gradient computations as devices in **ProxSkip**.

# Large number of clients

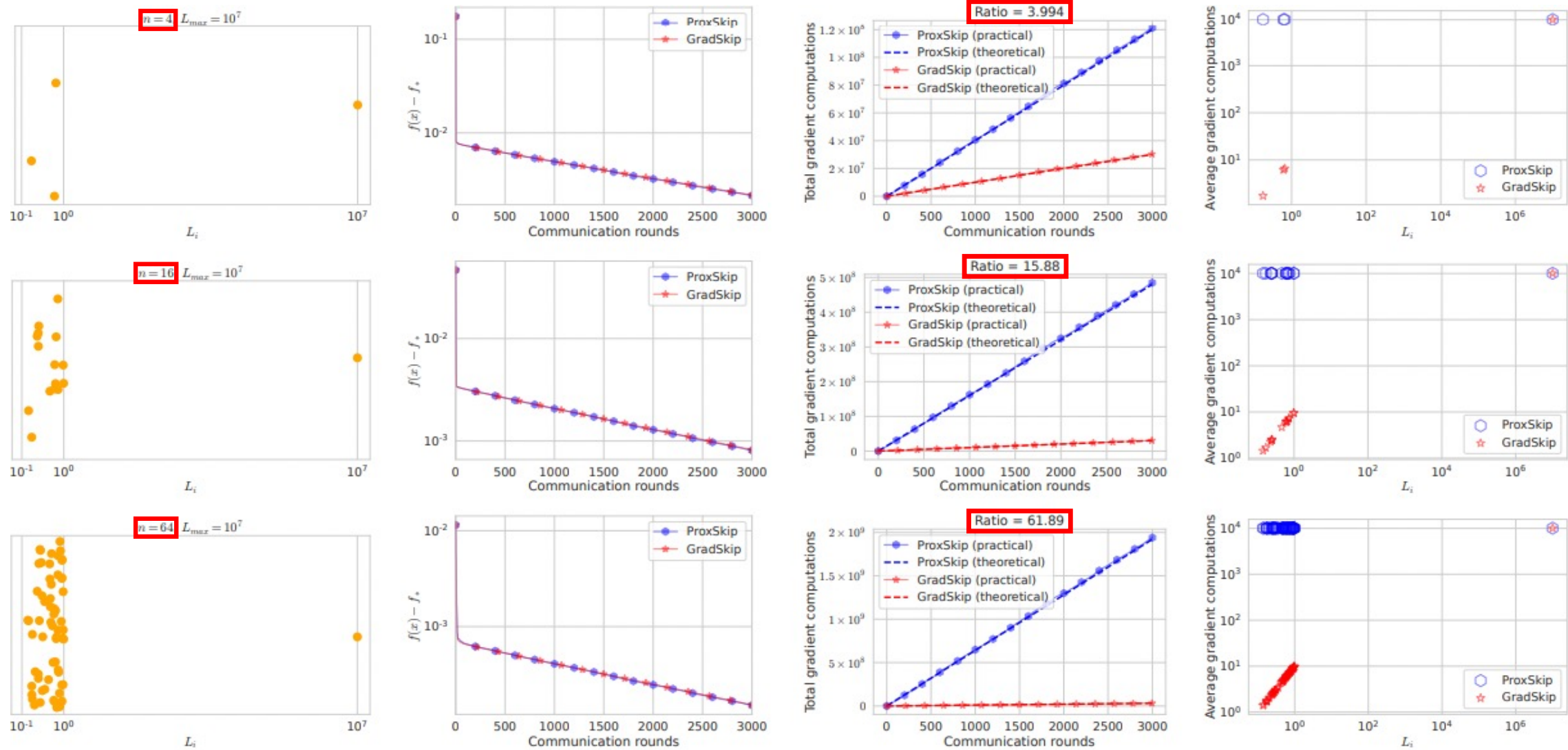


Figure 2: The columns have the same meaning as in Figure 1.



**Thank you**