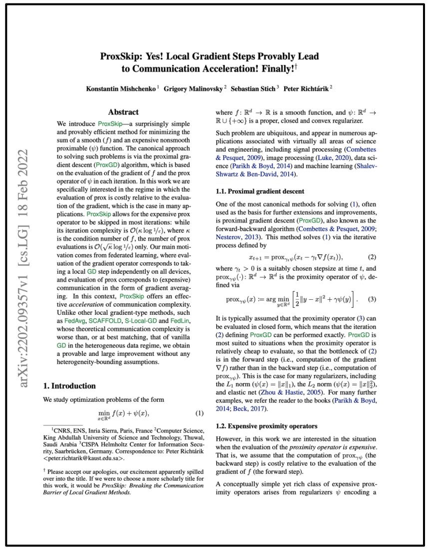
King Abdullah University of Science and Technology (KAUST) · Inria · CISPA Helmholtz Center for Information Security

ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication Acceleration! Finally!

Peter Richtárik

CrossFL Workshop @ MLSys 2022
Santa Clara & Virtual
September 1, 2022


arXiv:2202.09357v1 [cs.LG] 18 Feb 2022

ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication Acceleration! Finally![1]

Konstantin Mishchenko[1] Grigory Malinovsky[2] Sebastian Stich[3] Peter Richtárik[2]

Abstract

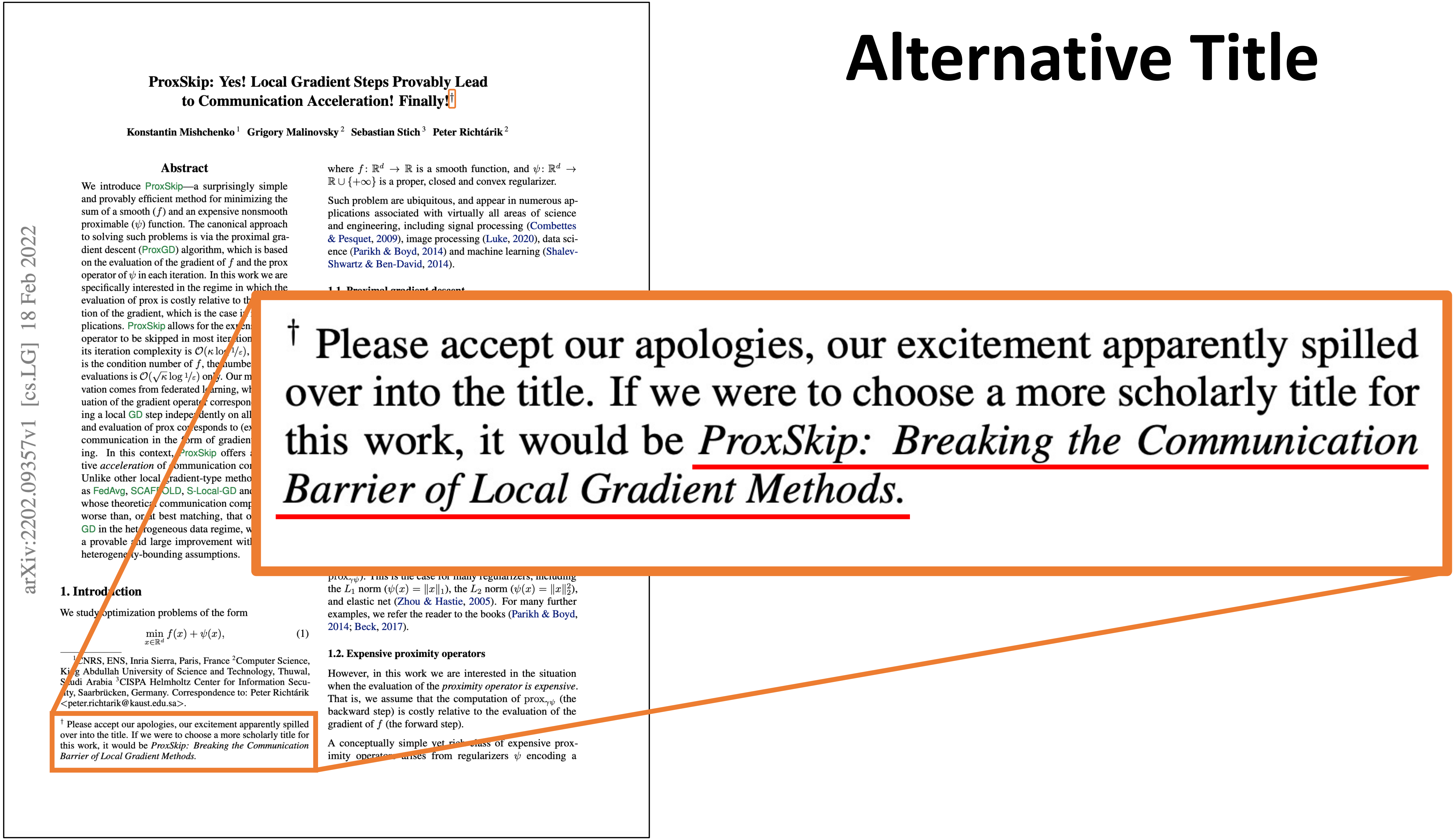
1. Introduction

1.1. Proximal gradient descent

1.2. Expensive proximity operators

Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich, Peter Richtárik
ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication Acceleration! Finally!
ICML 2022

Alternative Title



ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication Acceleration! Finally!†

Konstantin Mishchenko [1] Grigory Malinovsky [2] Sebastian Stich [3] Peter Richtárik [2]

arXiv:2202.09357v1 [cs.LG] 18 Feb 2022

Abstract

We introduce ProxSkip—a surprisingly simple and provably efficient method for minimizing the sum of a smooth (f) and an expensive nonsmooth proximable (ψ) function. The canonical approach to solving such problems is via the proximal gradient descent (ProxGD) algorithm, which is based on the evaluation of the gradient of f and the prox operator of ψ in each iteration. In this work we are specifically interested in the regime in which the evaluation of prox is costly relative to th…

where $f\colon \mathbb{R}^d \to \mathbb{R}$ is a smooth function, and $\psi\colon \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is a proper, closed and convex regularizer.

Such problem are ubiquitous, and appear in numerous applications associated with virtually all areas of science and engineering, including signal processing (Combettes & Pesquet, 2009), image processing (Luke, 2020), data science (Parikh & Boyd, 2014) and machine learning (Shalev-Shwartz & Ben-David, 2014).

1. Introduction

We study optimization problems of the form

$$\min_{x\in\mathbb{R}^d} f(x) + \psi(x), \quad (1)$$

[1] CNRS, ENS, Inria Sierra, Paris, France [2] Computer Science, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia [3] CISPA Helmholtz Center for Information Security, Saarbrücken, Germany. Correspondence to: Peter Richtárik <peter.richtarik@kaust.edu.sa>.

† Please accept our apologies, our excitement apparently spilled over into the title. If we were to choose a more scholarly title for this work, it would be *ProxSkip: Breaking the Communication Barrier of Local Gradient Methods.*

the L_1 norm ($\psi(x) = \|x\|_1$), the L_2 norm ($\psi(x) = \|x\|_2^2$), and elastic net (Zhou & Hastie, 2005). For many further examples, we refer the reader to the books (Parikh & Boyd, 2014; Beck, 2017).

1.2. Expensive proximity operators

However, in this work we are interested in the situation when the evaluation of the *proximity operator is expensive*. That is, we assume that the computation of $\mathrm{prox}_{\gamma\psi}$ (the backward step) is costly relative to the evaluation of the gradient of f (the forward step).

A conceptually simple yet rich class of expensive proximity operators arises from regularizers ψ encoding a

Coauthors

Konstantin Mishchenko

KAUST
King Abdullah University of Science and Technology

Inria

Grigory Malinovsky

KAUST
King Abdullah University of Science and Technology

Sebastian Stich

CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

Outline of the Talk

1. Introduction
2. Consensus Reformulation
3. Proximal Gradient Descent
4. **ProxSkip: Algorithm**
5. **ProxSkip: Theory**
6. Experiments
7. Extensions

Part 1
Introduction

Distributed Gradient Descent

Federated Training of a Supervised Machine Learning Model

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

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 $\mathcal{D}_1, \ldots, \mathcal{D}_n$ can be arbitrarily heterogeneous

Distributed Gradient Descent

Optimization problem:

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

$$\begin{aligned} x_{t+1} &= x_t - \gamma \nabla f(x_t) \\ &= x_t - \gamma \frac{1}{n}\sum_{i=1}^{n} \nabla f_i(x_t) \end{aligned}$$

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

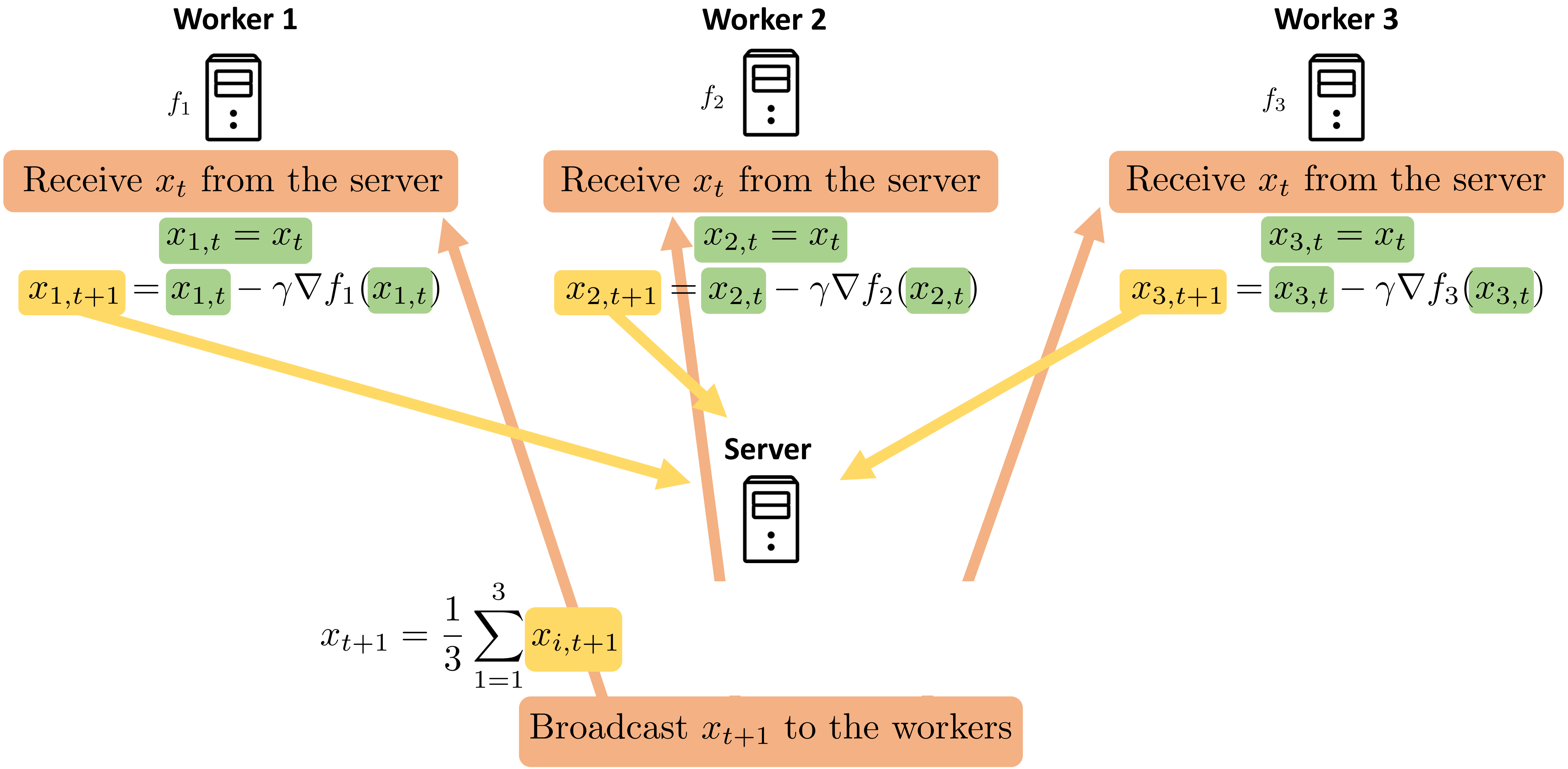
d-dimensional gradient computed by machine *i*

Distributed Gradient Descent

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

Optimization problem:

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



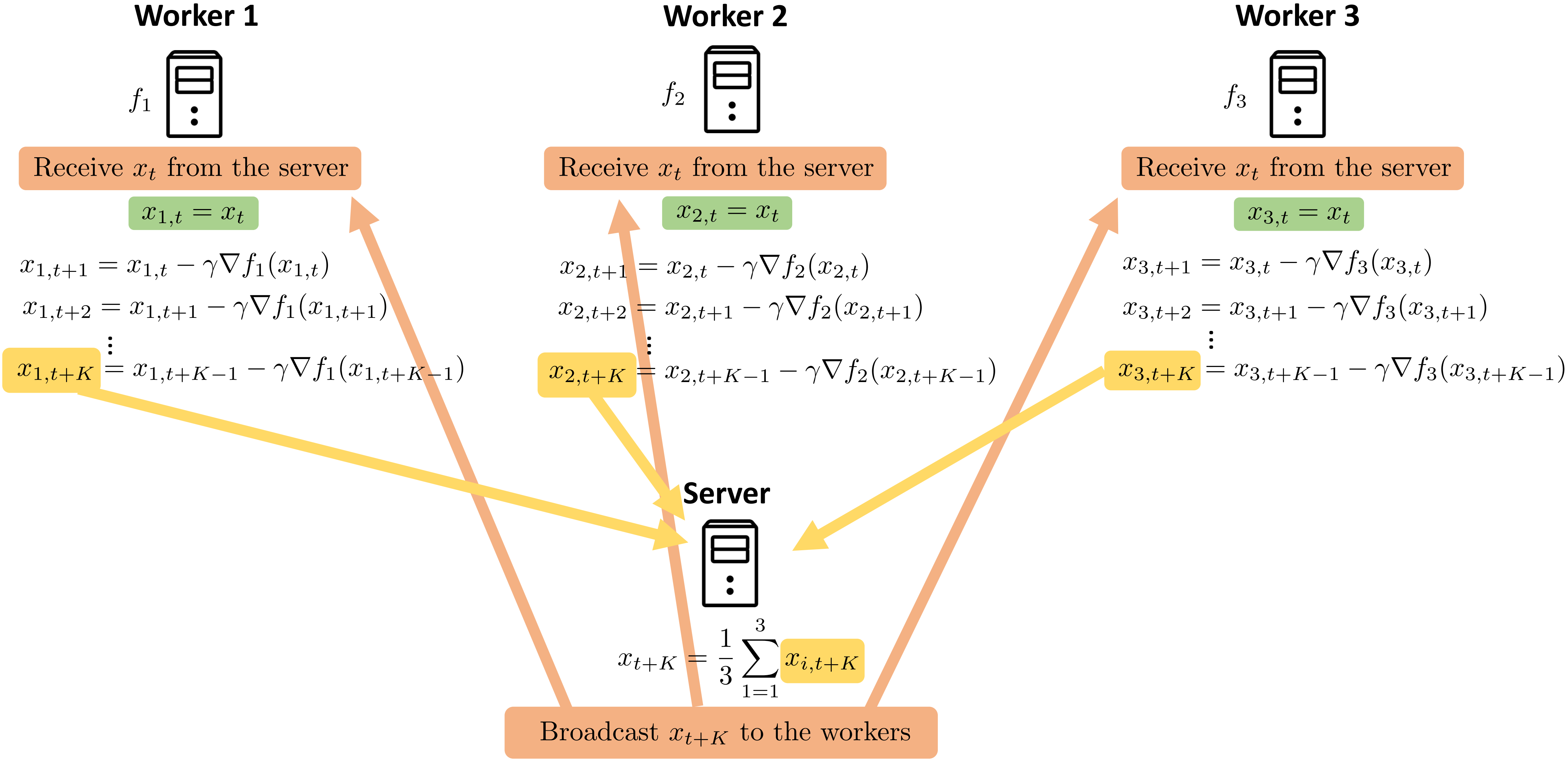
Distributed Local Gradient Descent

Distributed Local Gradient Descent

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

Optimization problem:

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



From GD to Local GD

1847

Gradient Descent (GD)

Compte Rendu à l'Académie des Sciences

(L. A. Cauchy)

1995

Local GD Proposed

Parallel Gradient Distribution in Unconstrained Optimization

(O. L. Mangasarian)

2017

Federated Averaging: Local GD Plays a Key Role in Federated Learning

Communication-efficient Learning of Deep Networks from Decentralized Data

(H. B. McMahan et al)

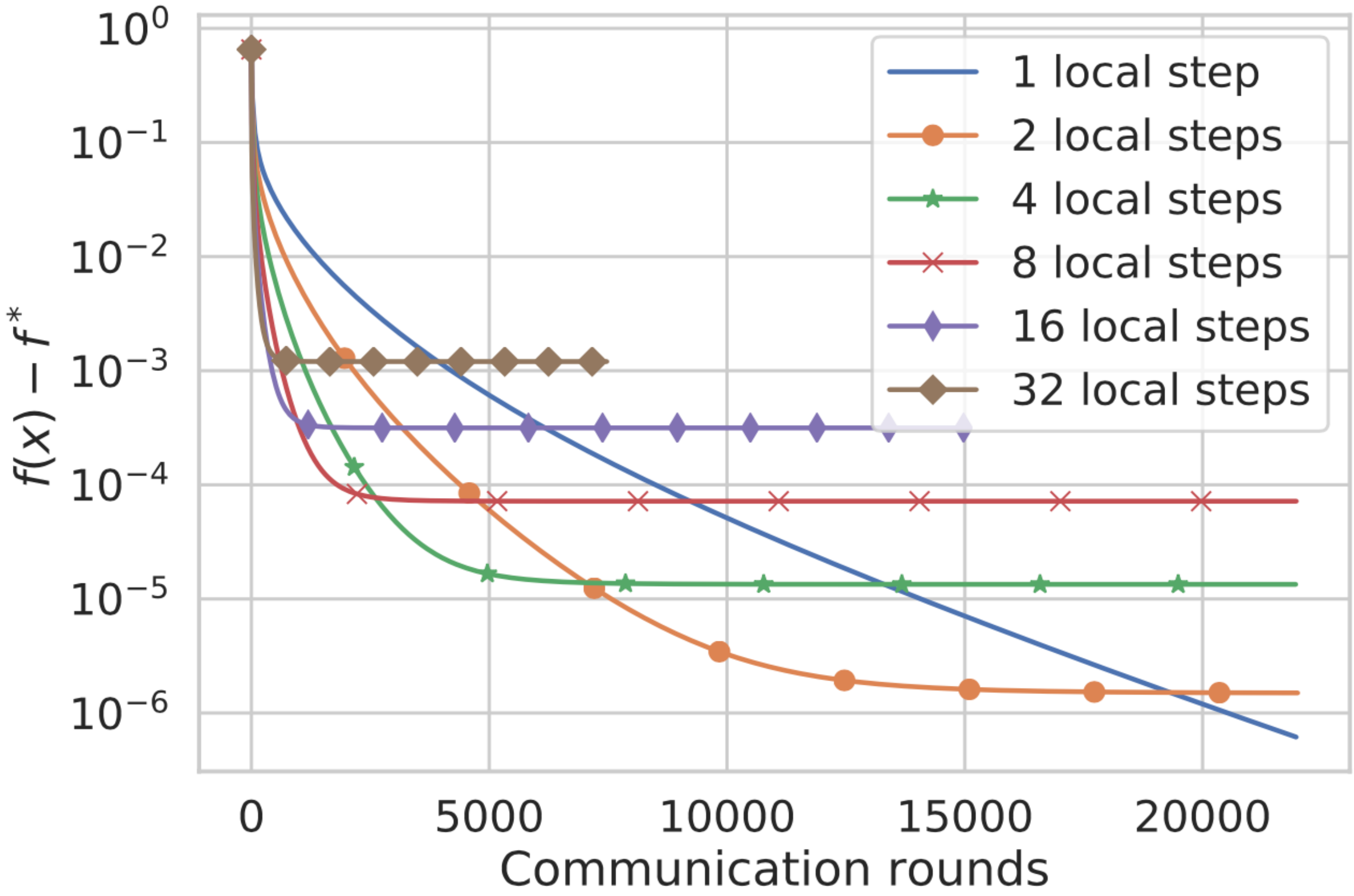
2020

First General Theory for Local GD

First Analysis of Local GD on Heterogeneous Data

(Khaled, Mishchenko & R)

What do the Local Steps do?



Plot taken from:

Ahmed Khaled, Konstantin Mishchenko, Peter Richtárik
First Analysis of Local GD on Heterogeneous Data
NeurIPS 2019 Workshop on Federated Learning for Data Privacy and Confidentiality, 2019

L2-regularized logistic regression
LibSVM `mushrooms` dataset

Linearly Converging Local GD Methods

Local GD with GD-like (=Linear) Convergence

2020

SCAFFOLD

Scaffold: Stochastic Controlled Averaging for Federated Learning

(Karimireddy, Kale, Mohri, Reddi, Stich, Suresh)

2021

S-Local-GD

Local SGD: Unified Theory and New Efficient Methods

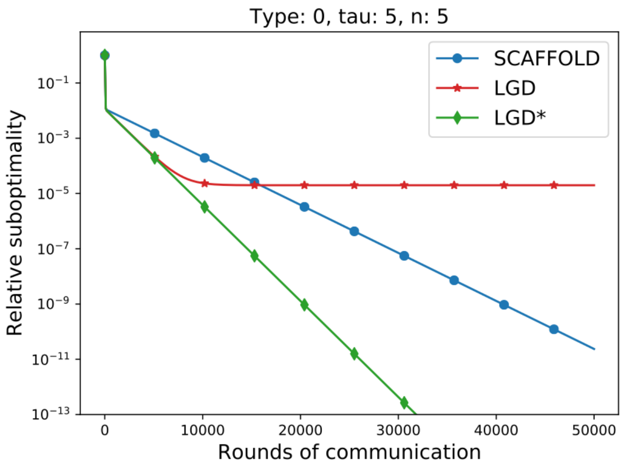
(Gorbunov, Hanzely & R)

2021

FedLin

Linear Convergence in Federated Learning...

(Mitra, Jaafar, Pappas, Hassani)



Key Theoretical Problem in Federated Learning

Local gradient steps are of key importance in FL.

In practice, local steps improve communication efficiency. But in theory*, they do not!!!

Is the situation hopeless, or can we show that (appropriately designed) local steps help?

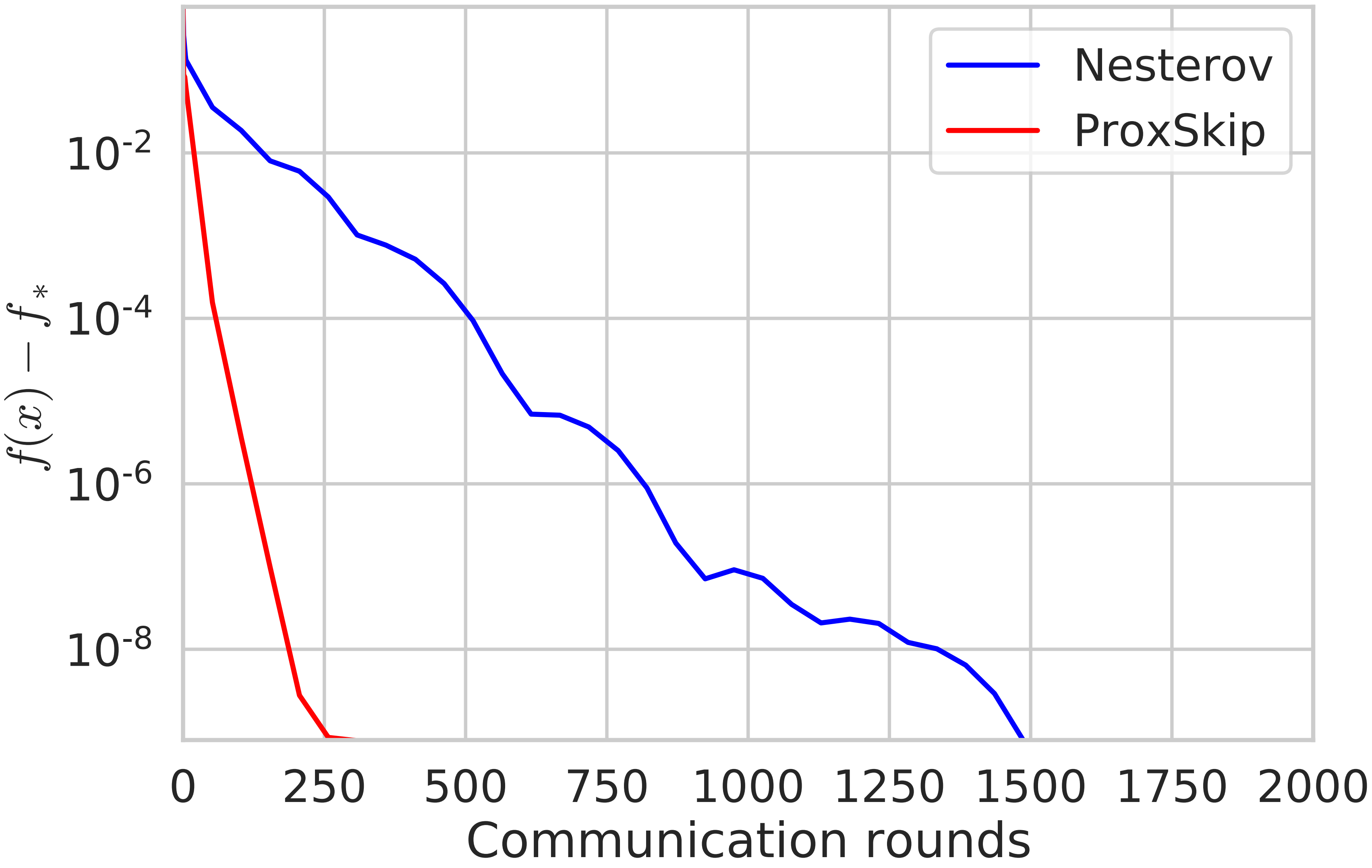
Federated Learning: ProxSkip vs Baselines

Table 1. The performance of federated learning methods employing multiple local gradient steps in the strongly convex regime.

method	# local steps per round	# floats sent per round	stepsize on client i	linear rate?	# rounds	rate better than GD?
GD (Nesterov, 2004)	1	d	$\frac{1}{L}$	✓	$\tilde{\mathcal{O}}(\kappa)$ [c]	✗
LocalGD (Khaled et al., 2019; 2020)	τ	d	$\frac{1}{\tau L}$	✗	$\mathcal{O}\left(\frac{G^2}{\mu n \tau \varepsilon}\right)$ [d]	✗
Scaffold (Karimireddy et al., 2020)	τ	$2d$	$\frac{1}{\tau L}$ [e]	✓	$\tilde{\mathcal{O}}(\kappa)$ [c]	✗
S-Local-GD [a] (Gorbunov et al., 2021)	τ	$d < \# < 2d$ [f]	$\frac{1}{\tau L}$	✓	$\tilde{\mathcal{O}}(\kappa)$	✗
FedLin [b] (Mitra et al., 2021)	τ_i	$2d$	$\frac{1}{\tau_i L}$	✓	$\tilde{\mathcal{O}}(\kappa)$ [c]	✗
Scaffnew [g] (this work) for any $p \in (0,1]$	$\frac{1}{p}$ [h]	d	$\frac{1}{L}$	✓	$\tilde{\mathcal{O}}\left(p\kappa + \frac{1}{p}\right)$ [c]	✓ (for $p > \frac{1}{\kappa}$)
Scaffnew [g] (this work) for optimal $p = \frac{1}{\sqrt{\kappa}}$	$\sqrt{\kappa}$ [h]	d	$\frac{1}{L}$	✓	$\tilde{\mathcal{O}}(\sqrt{\kappa})$ [c]	✓

[a] This is a special case of S-Local-SVRG, which is a more general method presented in (Gorbunov et al., 2021). S-Local-GD arises as a special case when full gradient is computed on each client.
[b] FedLin is a variant with a fixed but different number of local steps for each client. Earlier method S-Local-GD has the same update but random loop length.
[c] The $\tilde{\mathcal{O}}$ notation hides logarithmic factors.
[d] G is the level of dissimilarity from the assumption $\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x)\|^2 \leq G^2 + 2LB^2\left(f(x) - f_\star\right), \forall x$.
[e] We use Scaffold's cumulative local-global stepsize $\eta_l \eta_g$ for a fair comparison.
[f] The number of sent vectors depends on hyper-parameters, and it is randomized.
[g] Scaffnew (Algorithm 2) = ProxSkip (Algorithm 1) applied to the consensus formulation (6) + (7) of the finite-sum problem (5).
[h] ProxSkip (resp. Scaffnew) takes a *random* number of gradient (resp. local) steps before prox (resp. communication) is computed (resp. performed). What is shown in the table is the *expected* number of gradient (resp. local) steps.

Federated Learning: ProxSkip vs Nesterov



Part 2
Consensus Reformulation

Consensus Reformulation

Original problem:
optimization in $\mathbb{R}^d$

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

$$\Updownarrow$$

Consensus reformulation:
optimization in $\mathbb{R}^{nd}$

$$\min_{x_1, \ldots, x_n \in \mathbb{R}^d} \left\{ \frac{1}{n} \sum_{i=1}^{n} f_i(x_i) + \psi(x_1, \ldots, x_n) \right\}$$

Bad: Non-differentiable function

Good: Indicator function of a nonempty closed convex set

$$\psi(x_1, \ldots, x_n) \stackrel{\text{def}}{=} \begin{cases} 0, & \text{if } x_1 = \cdots = x_n, \\ +\infty, & \text{otherwise.} \end{cases}$$

Generalization 1: Constrained Optimization

Consensus reformulation:
optimization in $\mathbb{R}^{nd}$

$$\min_{x_1,\ldots,x_n \in \mathbb{R}^d} \left\{ \frac{1}{n} \sum_{i=1}^{n} f_i(x_i) + \psi(x_1,\ldots,x_n) \right\}$$

$$\psi(x_1,\ldots,x_n) \overset{\text{def}}{=} \begin{cases} 0, & \text{if } x_1 = \cdots = x_n, \\ +\infty, & \text{otherwise.} \end{cases}$$

Generalization 1 $\Downarrow$

$$\psi(x_1,\ldots,x_n) \overset{\text{def}}{=} \begin{cases} 0, & \text{if } (x_1,\cdots,x_n) \in C, \\ +\infty, & \text{otherwise.} \end{cases}$$

Arbitrary closed convex set (constraint)

Generalization 2: Composite Optimization

Consensus reformulation:
optimization in $\mathbb{R}^{nd}$

$$\min_{x_1,\ldots,x_n \in \mathbb{R}^d} \left\{ \frac{1}{n}\sum_{i=1}^{n} f_i\left(x_i\right) + \psi\left(x_1,\ldots,x_n\right) \right\}$$

$$\psi\left(x_1,\ldots,x_n\right) \stackrel{\text{def}}{=} \begin{cases} 0, & \text{if } x_1 = \cdots = x_n, \\ +\infty, & \text{otherwise.} \end{cases}$$

Generalization 2 $\Downarrow$

$$\psi\left(x_1,\ldots,x_n\right) : \mathbb{R}^{nd} \to \mathbb{R} \cup \{+\infty\}$$

is a proper closed convex function

The epigraph of ψ is a closed and convex set

$$\text{epi}(\psi) \stackrel{\text{def}}{=} \{(x,t) \mid \psi(x) \leq t)$$

Conceptual Simplification: from *nd* to *d'*

Composite optimization:

optimization in $\mathbb{R}^{nd}$

$$\min_{x_1,\ldots,x_n \in \mathbb{R}^d} \left\{ \frac{1}{n} \sum_{i=1}^{n} f_i(x_i) + \psi(x_1, \ldots, x_n) \right\}$$

Composite optimization:

optimization in $\mathbb{R}^{d'}$

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

$$\left\{ \begin{aligned} d' &= nd \\ x &= (x_1, \ldots, x_n) \\ f(x) &= \frac{1}{n} \sum_{i=1}^{n} f_i(x_i) \\ \psi(x) &= \psi(x_1, \ldots, x_n) \end{aligned} \right.$$

Part 3
Proximal Gradient Descent

Three Assumptions

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

The epigraph of ψ is a closed and convex set

$$\text{epi}(\psi) \overset{\text{def}}{=} \{(x,t) \in \mathbb{R}^d \times \mathbb{R} \mid \psi(x) \le t\}$$

A1 f is μ-convex and L-smooth:

$$\frac{\mu}{2}\|x-y\|^2 \le D_f(x,y) \le \frac{L}{2}\|x-y\|^2$$

A2 $\psi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is proper, closed, and convex

A3 ψ is proximable

Bregman divergence of f:

$$D_f(x,y) \overset{\text{def}}{=} f(x) - f(y) - \langle \nabla f(y), x - y \rangle$$

The proximal operator $\text{prox}_\psi : \mathbb{R}^d \to \mathbb{R}^d$ defined by

$$\text{prox}_\psi(x) \overset{\text{def}}{=} \arg\min_{u\in\mathbb{R}^d} \left(\psi(u) + \frac{1}{2}\|u-x\|^2 \right)$$

can be evaluated exactly (e.g., in closed form)

Key Method: Proximal Gradient Descent

proximal operator:

$$\mathrm{prox}_{\psi}(x) \stackrel{\text{def}}{=} \arg\min_{u \in \mathbb{R}^d} \left(\psi(u) + \frac{1}{2}\|u - x\|^2 \right)$$

stepsize

$$x_t - \gamma \nabla f(x_t)$$

gradient operator

$$x \mapsto x - \gamma \nabla f(x)$$

Proximal Gradient Descent: Theory

f is μ-convex and L-smooth:

$$\frac{\mu}{2}\|x-y\|^2 \le D_f(x,y) \le \frac{L}{2}\|x-y\|^2$$

$\frac{L}{\mu}$ is the condition number of f

Theorem:

$$t \ge \frac{L}{\mu}\log\frac{1}{\varepsilon} \implies \|x_t - x_\star\|^2 \le \varepsilon\|x_0 - x_\star\|^2$$

(for stepsize $\gamma = \frac{1}{L}$)

iterations

Error tolerance

$$x_\star \overset{\text{def}}{=} \arg\min_{x\in\mathbb{R}^d} f(x) + \psi(x)$$

Part 4
The ProxSkip Algorithm

What to do When the Prox is Expensive?

Can we somehow get away with fewer evaluations of the proximity operator in the Proximal GD method?

Approach 1

- We'll skipp ALL prox evaluations!
- The method is NOT implementable!
- Serves as an inspiration for Approach 2

Approach 2 (ProxSkip)

- We'll skip MANY prox evaluations!
- The method is implementable!

Approach 1:
Simple, Extreme but
Practically Useless Variant

Removing ψ via a Reformulation

$$h_\star \overset{\text{def}}{=} \nabla f(x_\star)$$

$$x_\star \overset{\text{def}}{=} \arg\min_{x \in \mathbb{R}^d} f(x) + \psi(x)$$

$$\min_{x \in \mathbb{R}^d} f(x) - \langle h_\star, x \rangle$$

$x_\star$ is a solution of the above problem!

By the 1st order optimality conditions, the solution satisfies $\nabla f(x) - \nabla f(x_\star) = 0$

We do not know $h_\star = \nabla f(x_\star)$!

Apply Gradient Descent to the Reformulation

$$h_\star \stackrel{\text{def}}{=} \nabla f(x_\star)$$
$$x_\star \stackrel{\text{def}}{=} \arg\min_{x \in \mathbb{R}^d} f(x) + \psi(x)$$

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

✓ We do not need to evaluate the prox of ψ at all!

✗ We do not know $h_\star$ and hence can't implement the method!

Idea: Try to "Learn" the Optimal Gradient Shift

Desire: $h_t \to h_\star$

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

Perhaps we can learn $h_\star$ with only occasional access to ψ?

Approach 2:
The ProxSkip Method

ProxSkip: The Algorithm (Bird's Eye View)

1

$$\hat{x}_{t+1} = x_t - \gamma \left(\nabla f(x_t) - h_t \right)$$

2a

with probability $1 - p$ do $\quad x_{t+1} = \hat{x}_{t+1} \quad h_{t+1} = h_t$

$1 - p \approx 1$

2b

with probability p do

$p \approx 0$

evaluate $\mathrm{prox}_{\frac{\gamma}{p}\psi}(?)$

$x_{t+1} = \, ? \quad h_{t+1} = \, ?$

ProxSkip: The Algorithm (Detailed View)

Algorithm 1 ProxSkip

stepsize $\gamma > 0$, probability $p > 0$, initial iterate $x_0 \in \mathbb{R}^d$, initial control variate $h_0 \in \mathbb{R}^d$, number of iterations $T \geq 1$

for $t = 0, 1, \ldots, T-1$ **do**

  $\hat{x}_{t+1} = x_t - \gamma(\nabla f(x_t) - h_t)$ ◇ Take a gradient-type step adjusted via the control variate h_t

  Flip a coin $\theta_t \in \{0, 1\}$ where $\mathrm{Prob}(\theta_t = 1) = p$ ◇ Flip a coin that decides whether to skip the prox or not

  if $\theta_t = 1$ **then**

    $x_{t+1} = \mathrm{prox}_{\frac{\gamma}{p}\psi}\left(\hat{x}_{t+1} - \frac{\gamma}{p} h_t\right)$ ◇ Apply prox, but only very rarely! (with small probability p)

  else

    $x_{t+1} = \hat{x}_{t+1}$ ◇ Skip the prox!

  end if

  $h_{t+1} = h_t + \frac{p}{\gamma}(x_{t+1} - \hat{x}_{t+1})$ ◇ Update the control variate h_t

end for

Part 5
ProxSkip Theory

ProxSkip: Bounding the # of Iterations

Theorem:

f is μ-convex and L-smooth:

$$\frac{\mu}{2}\|x-y\|^2 \leq D_f(x,y) \leq \frac{L}{2}\|x-y\|^2$$

$\frac{L}{\mu}$ is the condition number of f

$$t \geq \max\left\{\frac{L}{\mu}, \frac{1}{p^2}\right\}\log\frac{1}{\varepsilon} \quad \Longrightarrow \quad \mathbb{E}\left[\Psi_t\right] \leq \varepsilon\Psi_0$$

iterations

p = probability of evaluating the prox

Lyapunov function:

$$\Psi_t \stackrel{\text{def}}{=} \left\|x_t - x_\star\right\|^2 + \frac{1}{L^2p^2}\left\|h_t - h_\star\right\|^2$$

ProxSkip: Optimal Prox-Evaluation Probability

Since in each iteration we evaluate the prox with probability p, the expected number of prox evaluations after t iterations is:

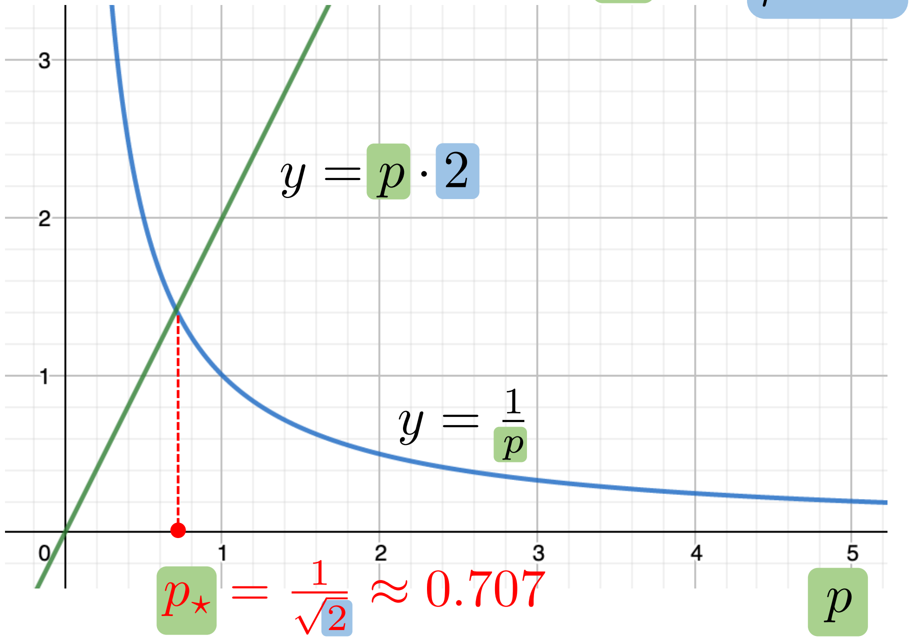
$\frac{L}{\mu}$ is the condition number of f

$$p \cdot t = p \cdot \max\left\{\frac{L}{\mu}, \frac{1}{p^2}\right\} \cdot \log \frac{1}{\varepsilon} = \max\left\{p \cdot \frac{L}{\mu}, \frac{1}{p}\right\} \cdot \log \frac{1}{\varepsilon}$$

Minimized for p satisfying $p \cdot \frac{L}{\mu} = \frac{1}{p}$

$$\Rightarrow \quad p_\star = \frac{1}{\sqrt{L/\mu}}$$

Computation of optimal $p_\star$ for $\frac{L}{\mu} = 2$



ProxSkip: # of Gradient and Prox Evaluations

$$p_\star = \frac{1}{\sqrt{L/\mu}} \quad \Rightarrow$$

# of iterations	$\max\left\{\frac{L}{\mu}, \frac{1}{p^2}\right\} \cdot \log\frac{1}{\varepsilon}$	$\frac{L}{\mu} \cdot \log\frac{1}{\varepsilon}$
# of gradient evaluations	$\max\left\{\frac{L}{\mu}, \frac{1}{p^2}\right\} \cdot \log\frac{1}{\varepsilon}$	$\frac{L}{\mu} \cdot \log\frac{1}{\varepsilon}$
Expected # of prox evaluations	$\max\left\{p \cdot \frac{L}{\mu}, \frac{1}{p}\right\} \cdot \log\frac{1}{\varepsilon}$	$\sqrt{\frac{L}{\mu}} \cdot \log\frac{1}{\varepsilon}$
Expected # of gradient evaluations between 2 prox evaluations	$\frac{1}{p}$	$\sqrt{\frac{L}{\mu}}$

Federated Learning: ProxSkip vs Baselines

Table 1. The performance of federated learning methods employing multiple local gradient steps in the strongly convex regime.

method	# local steps per round	# floats sent per round	stepsize on client i	linear rate?	# rounds	rate better than GD?
GD (Nesterov, 2004)	1	d	$\frac{1}{L}$	✓	$\tilde{\mathcal{O}}(\kappa)$ [c]	✗
LocalGD (Khaled et al., 2019; 2020)	τ	d	$\frac{1}{\tau L}$	✗	$\mathcal{O}\left(\frac{G^2}{\mu n \tau \varepsilon}\right)$ [d]	✗
Scaffold (Karimireddy et al., 2020)	τ	$2d$	$\frac{1}{\tau L}$ [e]	✓	$\tilde{\mathcal{O}}(\kappa)$ [c]	✗
S-Local-GD [a] (Gorbunov et al., 2021)	τ	$d < \# < 2d$ [f]	$\frac{1}{\tau L}$	✓	$\tilde{\mathcal{O}}(\kappa)$	✗
FedLin [b] (Mitra et al., 2021)	τ_i	$2d$	$\frac{1}{\tau_i L}$	✓	$\tilde{\mathcal{O}}(\kappa)$ [c]	✗
Scaffnew [g] (this work) for any $p \in (0, 1]$	$\frac{1}{p}$ [h]	d	$\frac{1}{L}$	✓	$\tilde{\mathcal{O}}\left(p\kappa + \frac{1}{p}\right)$ [c]	✓ (for $p > \frac{1}{\kappa}$)
Scaffnew [g] (this work) for optimal $p = \frac{1}{\sqrt{\kappa}}$	$\sqrt{\kappa}$ [h]	d	$\frac{1}{L}$	✓	$\tilde{\mathcal{O}}(\sqrt{\kappa})$ [c]	✓

[a] This is a special case of S-Local-SVRG, which is a more general method presented in (Gorbunov et al., 2021). S-Local-GD arises as a special case when full gradient is computed on each client.

[b] FedLin is a variant with a fixed but different number of local steps for each client. Earlier method S-Local-GD has the same update but random loop length.

[c] The $\tilde{\mathcal{O}}$ notation hides logarithmic factors.

[d] G is the level of dissimilarity from the assumption $\frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(x)\|^2 \leq G^2 + 2LB^2 (f(x) - f_\star), \forall x$.

[e] We use Scaffold's cumulative local-global stepsize $\eta_l \eta_g$ for a fair comparison.

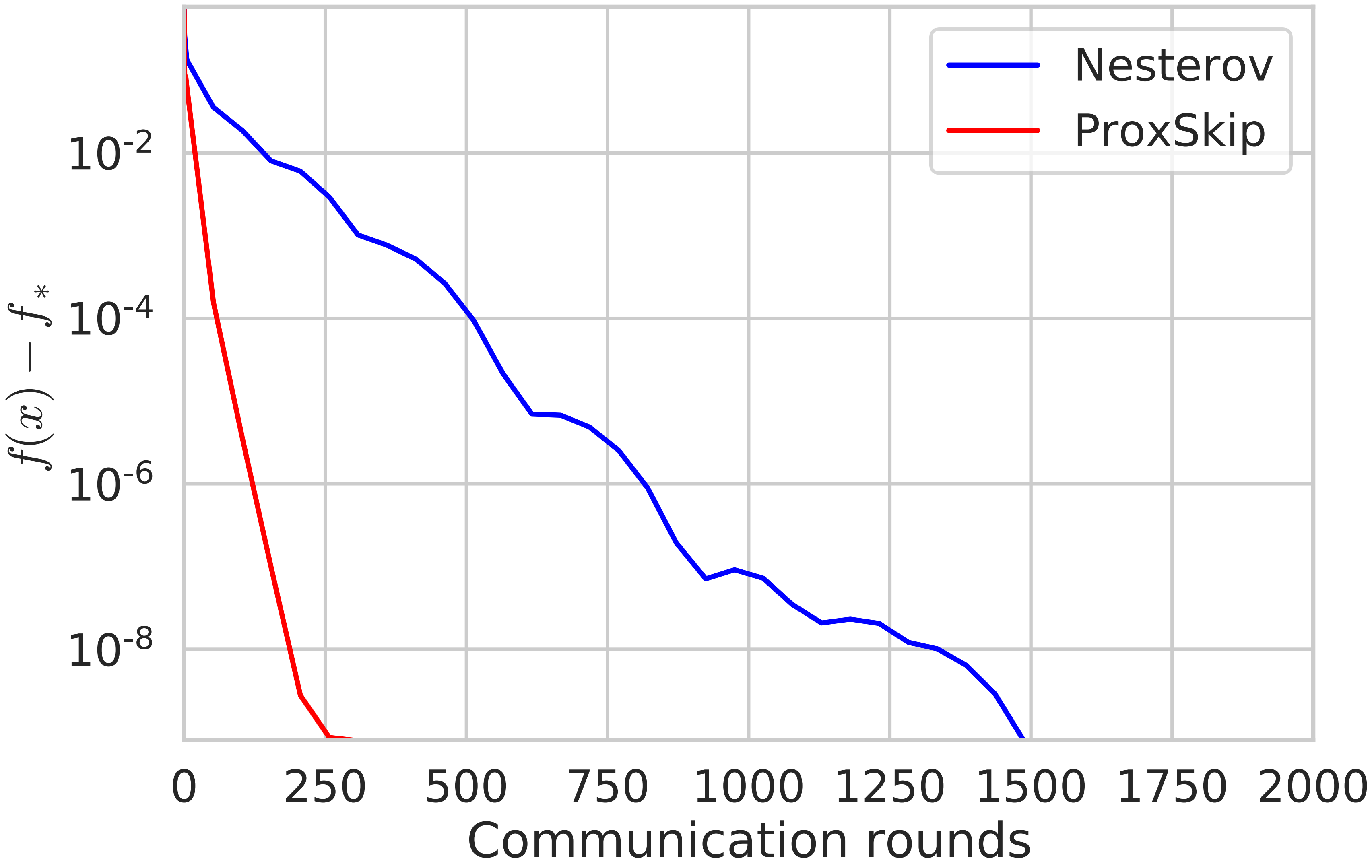
[f] The number of sent vectors depends on hyper-parameters, and it is randomized.

[g] Scaffnew (Algorithm 2) = ProxSkip (Algorithm 1) applied to the consensus formulation (6) + (7) of the finite-sum problem (5).

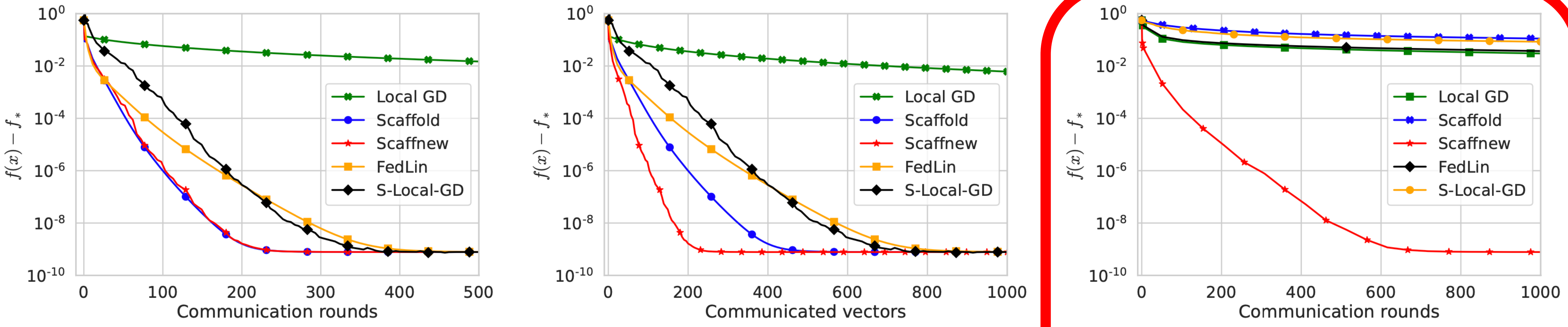
[h] ProxSkip (resp. Scaffnew) takes a *random* number of gradient (resp. local) steps before prox (resp. communication) is computed (resp. performed). What is shown in the table is the *expected* number of gradient (resp. local) steps.

Part 6
Experiments

Scaffnew (=ProxSkip applied to FL) vs Nesterov



Scaffnew (=ProxSkip applied to FL) vs Baselines



(a) tuned hyper-parameters (b) tuned hyper-parameters (c) theoretical hyper-parameters

Figure 1. **Deterministic Problem**. Comparison of Scaffnew to other local update methods that tackle data-heterogeneity and to LocalGD. In (a) we compare communication rounds with optimally tuned hyper-parameters. In (b) we compare communicated vectors (Scaffold, FedLin and S-Local-GD require transmission of additional variables). In (c), we compare communication rounds with the algorithm parameters set to the best theoretical stepsizes used in the convergence proofs.

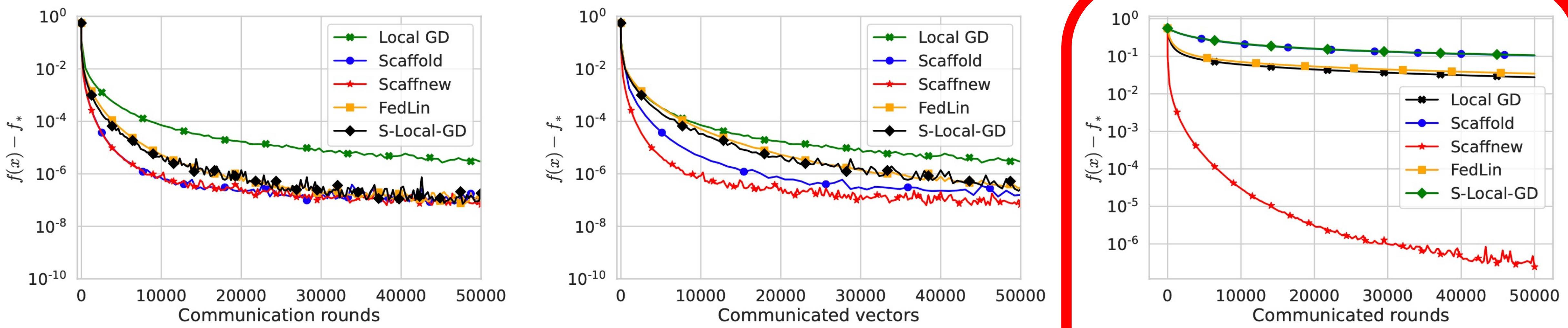
L2-regularized logistic regression:

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

$$a_i \in \mathbb{R}^d,\ b_i \in \{-1, +1\},\ \lambda = L/10^4$$

w8a dataset from LIBSVM library (Chang & Lin, 2011)

Scaffnew (=ProxSkip applied to FL) vs Baselines



(a) tuned hyper-parameters (b) tuned hyper-parameters (c) theoretical hyper-parameters

Figure 2. **Stochastic Problem**. Comparison of Scaffnew to other local update methods that tackle data-heterogeneity and to LocalSGD. In (a) we compare commnication rounds with optimally tuned hyper-parameters. In (b) we compare communicated vectors and in (c), we compare communication rounds with the algorithm parameters set to the best theoretical stepsizes used in the convergence proofs.

L2-regularized logistic regression:

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

$$a_i \in \mathbb{R}^d,\ b_i \in \{-1, +1\},\ \lambda = L/10^4$$

w8a dataset from LIBSVM library (Chang & Lin, 2011)

Part 7
Extensions

Extension 1: From Gradients to Stochastic Gradients

- As described, in ProxSkip each worker computes the **full gradient** of its local function
- It's often better to consider a **cheap stochastic approximation of the gradient** instead
  - We consider this extension in the paper
  - We provide theoretical convergence rates

$$\nabla f_i(x_t) \quad \Rightarrow \quad g_i(x_t)$$

Full gradient — Stochastic gradient

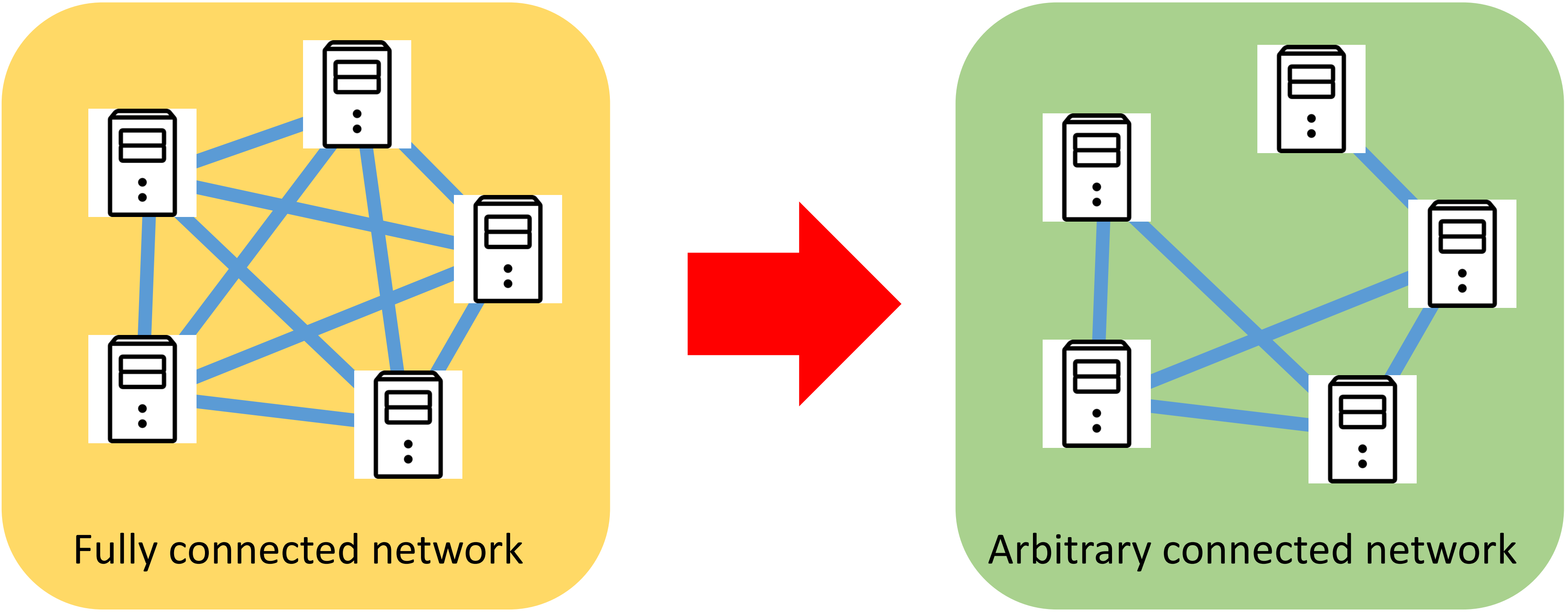
Assumptions:

(unbiasedness) $\mathbb{E}\left[g_{i,t}(x_t) \mid x_t\right] = \nabla f_i(x_t)$

(expected smoothness) (Gower et al, 2019) $\mathrm{E}\left[\left\|g_{i,t}\left(x_t\right) - \nabla f\left(x_\star\right)\right\|^2 \mid x_t\right] \leq 2AD_f\left(x_t, x_\star\right) + C$

Extension 2: From Fully Connected Network to Arbitrary Connected Network

- In each communication round of ProxSkip, **each worker sends messages to all oher workers** (e.g., through a server).
  - We can think of ProxSkip workers as the nodes of a **fully-connected network.**
  - In each communication round, all **workers communicate with their neighbors.**
- In the paper we provide extension to **arbitrary connected networks.**



Three Follow-up Papers

Extension 3: Compressing the Prox

Laurent Condat and Peter Richtárik
RandProx: Primal-dual optimization algorithms with randomized proximal updates
arXiv:2207.12891, 2022

Extension 4: Variance Reduction

Grigory Malinovsky, Kai Yi and Peter Richtárik
Variance reduced ProxSkip: Algorithm, theory and application to federated learning
arXiv:2207.04338, 2022

Extension 5: Less Local Training

Abdurakhmon Sadiev, Dmitry Kovalev and Peter Richtárik
Communication acceleration of local gradient methods via an accelerated primal-dual algorithm with inexact prox
arXiv:2207.03957, 2022

The End