On 5th Generation of Local Training Methods in Federated Learning

Peter Richtárik

KAUST
King Abdullah University of Science and Technology

2022 Workshop on Federated Learning and Analytics

Virtual via Google Meet

November 9-10, 2022

Konstantin Mishchenko

Arto Maranjyan

Dmitry Kovalev

Michal Grudzien

Laurent Condat

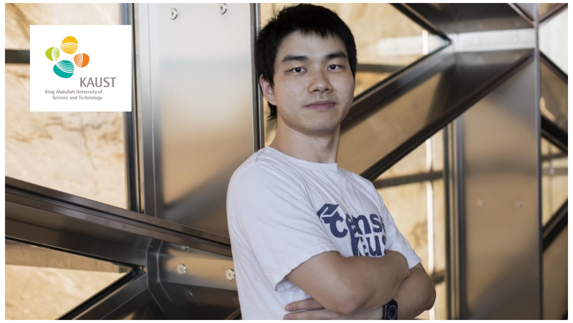
Sebastian Stich

Ivan Agarský

Abdurakhmon Sadiev

Mher Safaryan

Grigory Malinovsky



Kai Yi

Coauthors

Outline of the Talk

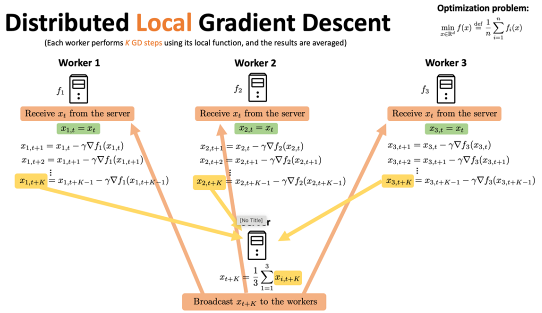
Local Training

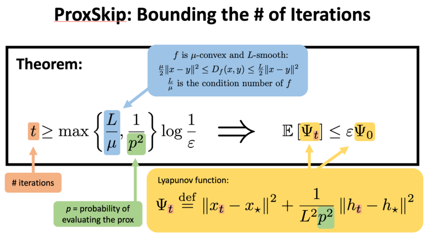
Brief History of Local Training

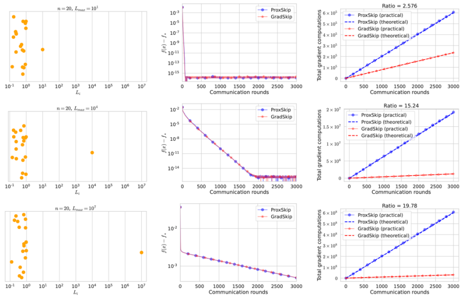
5th Generation of Local Training Methods

ProxSkip

GradSkip

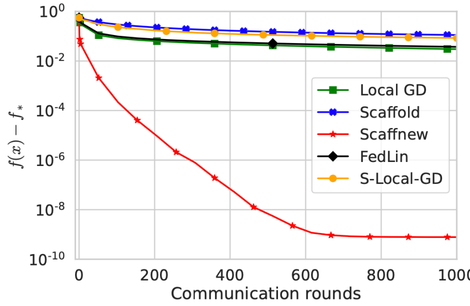






Algorithm 1 ProxSkip

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 $t = 0, 1, \dots, T-1$ **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 $\text{Prob}(\theta_t = 1) = p$ ◇ Flip a coin that decides whether to skip the prox or not
if $\theta_t = 1$ **then**
$x_{t+1} = \text{prox}_{\frac{\gamma}{p}\psi}\left(\hat{x}_{t+1} - \frac{\gamma}{p} h_t\right)$ ◇ 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



(c) theoretical hyper-parameters

Part I
Local Training

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

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

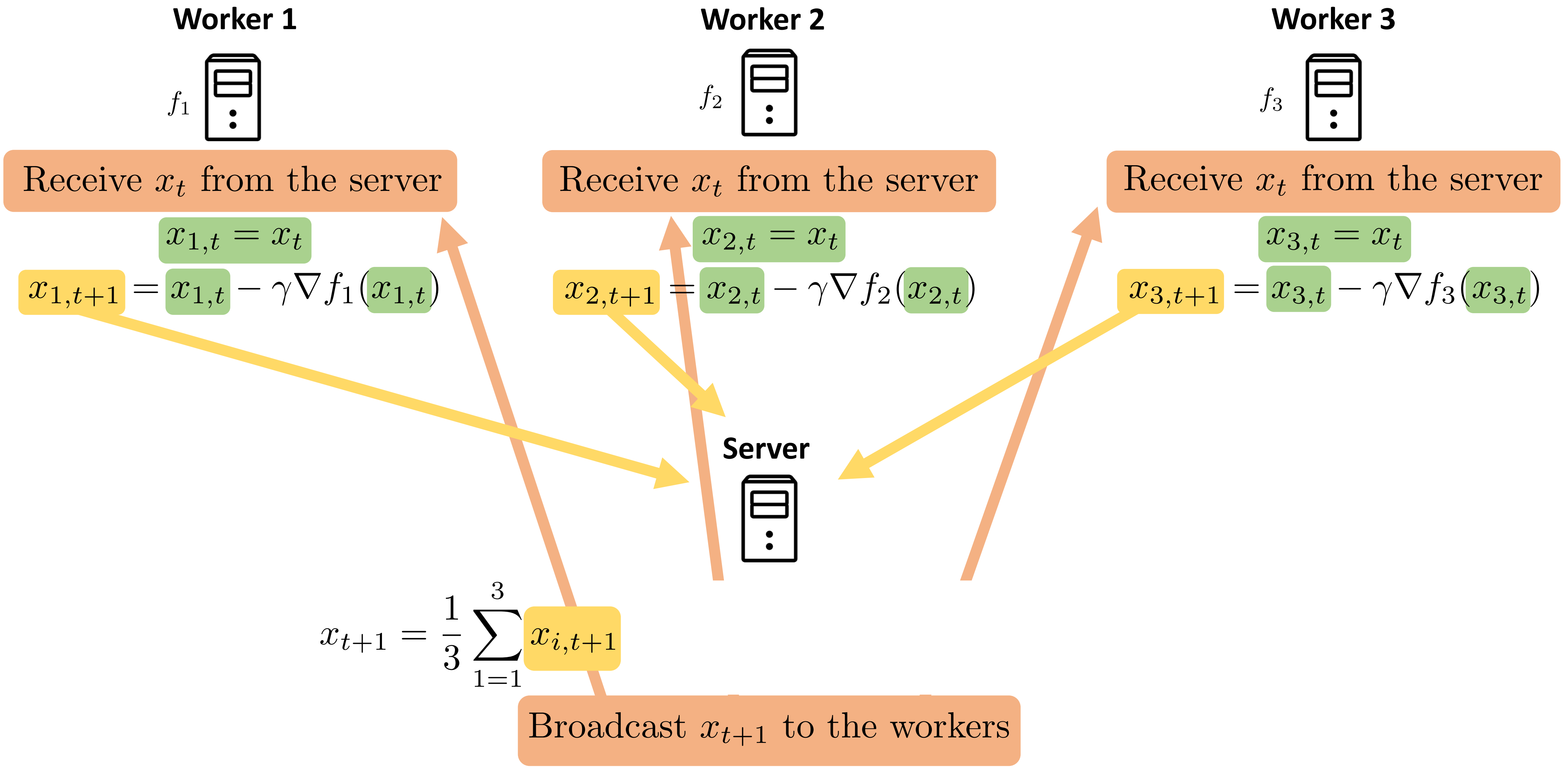
The datasets $\mathcal{D}_1, \ldots, \mathcal{D}_n$ can be arbitrarily heterogeneous

Distributed Gradient Descent

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

Optimization problem:

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

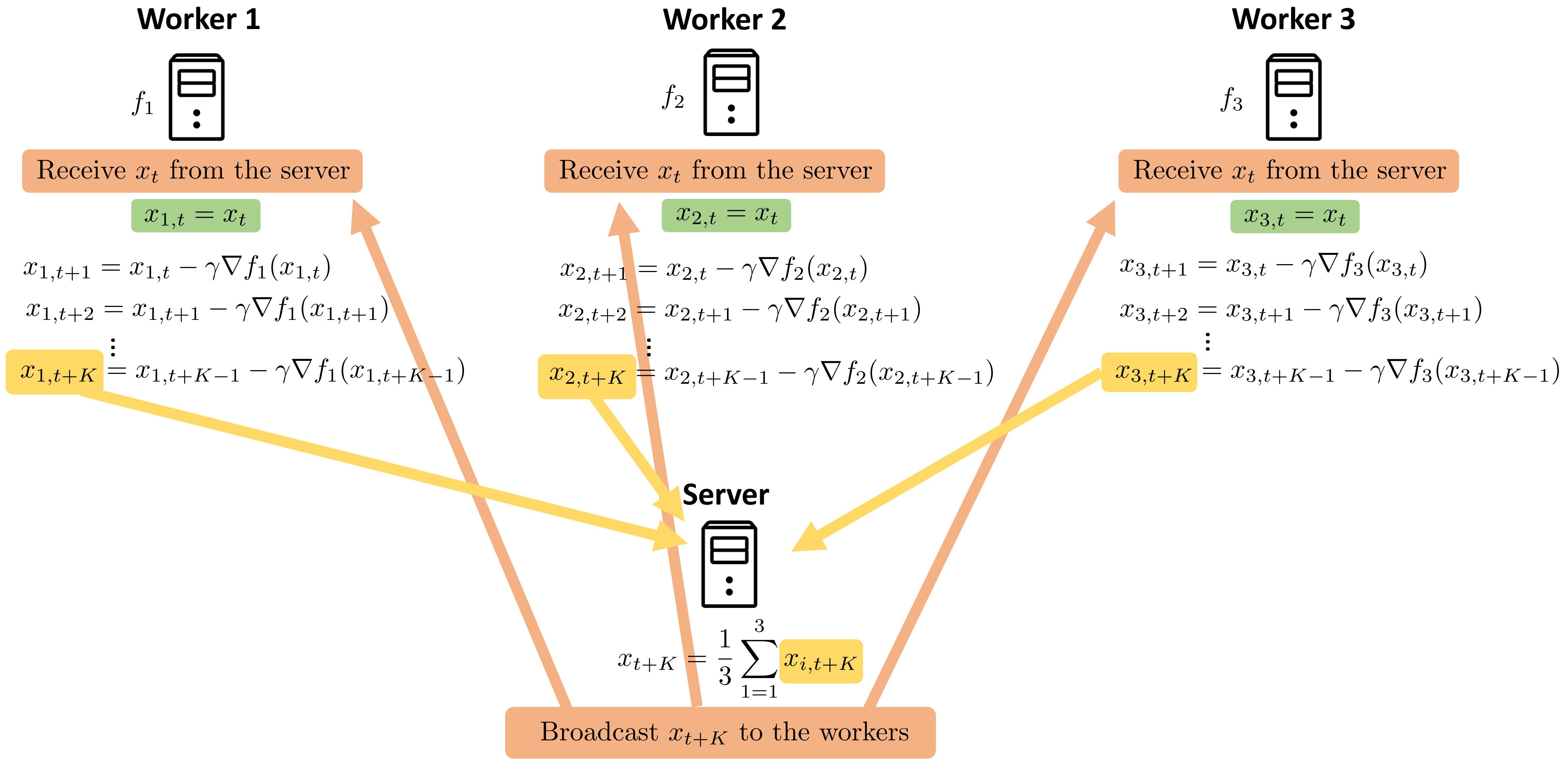


Distributed Local Gradient Descent

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

Optimization problem:

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



Part II
Brief History of Local Training

Grigory Malinovsky, Kai Yi and P.R.
Variance reduced ProxSkip: algorithm, theory and application to federated learning
NeurIPS 2022

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[(a)]	Theory	Assumptions	Comm. Complexity[(b)]	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[(c)]	linear but worse than GD	LFGD [Haddadpour and Mahdavi, 2019]
3. Sublinear	✓	standard[(d)]	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 **[THIS WORK]**

[(a)] 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.

[(b)] 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.

[(c)] *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.

[(d)] 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 and P.R.

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

NeurIPS 2022

Brief History of Local Training Methods
Generation 1: Heuristic

"No theory"

10/2014 PDF
Daniel Povey, Xiaohui Zhang, and Sanjeev Khudanpur
Parallel Training of DNNs with Natural Gradient and Parameter Averaging
ICLR Workshops 2015

11/2015 PDF
Philipp Moritz, Robert Nishihara, Ion Stoica, Michael I. Jordan
SparkNet: Training Deep Networks in Spark
ICLR 2015

02/2016 PDF
H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, Blaise Agüera y Arcas
Communication-Efficient Learning of Deep Networks from Decentralized Data
AISTATS 2017

Brief History of Local Training Methods
Generation 2: Homogeneous

"Theory requires data to be similar/homogeneous across the clients"

07/2019

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang and Zhihua Zhang
On the Convergence of FedAvg on Non-IID Data
ICLR 2020

Bounded gradients:

$$\|\nabla f_i(x)\| \leq B \quad \forall x \in \mathbb{R}^d \quad \forall i \in \{1, 2, \ldots, n\}$$

10/2019

Farzin Haddadpour and Mehrdad Mahdavi
On the Convergence of Local Descent Methods in Federated Learning
arXiv:1910.14425, 2019

Bounded gradient diversity (aka strong growth):

$$\frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(x)\|^2 \leq C \|\nabla f(x)\|^2 \quad \forall x \in \mathbb{R}^d$$

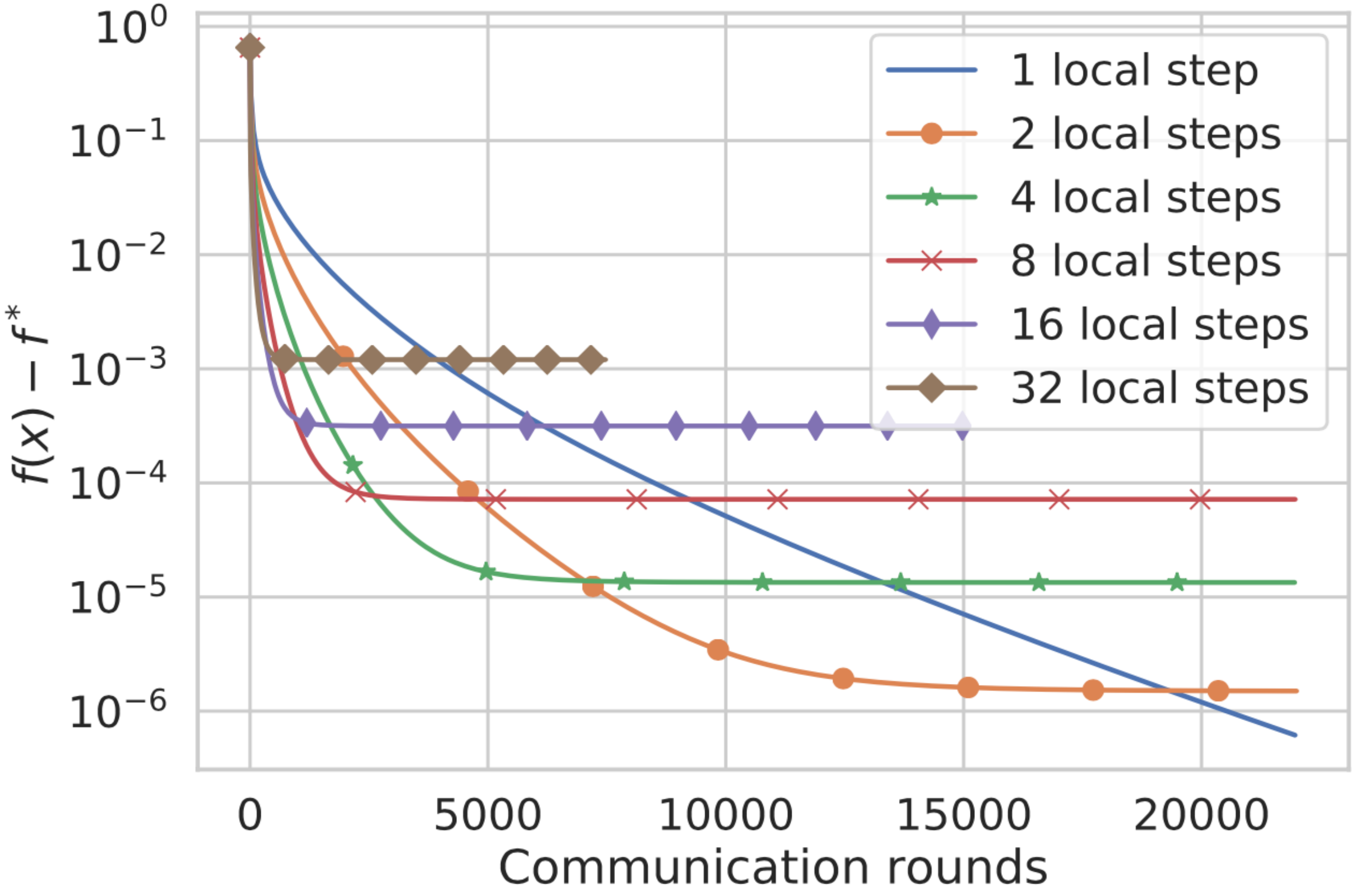
Brief History of Local Training Methods
Generation 3: Sublinear

"Heterogeneous data is allowed, but the rate is worse than GD"

10/2019 PDF
Ahmed Khaled, Konstantin Mishchenko and P.R.
First Analysis of Local GD on Heterogeneous Data
NeurIPS 2019 Workshop on Federated Learning for Data Privacy and Confidentiality, 2019

10/2019 PDF
Ahmed Khaled, Konstantin Mishchenko and P.R.
Tighter Theory for Local SGD on Identical and Heterogeneous Data
AISTATS 2020

Brief History of Local Training Methods
Generation 3: Sublinear



L2-regularized logistic regression
LibSVM `mushrooms` dataset

Brief History of Local Training Methods
Generation 4: Linear

“Heterogeneous data is allowed, but the rate ay best matches that of GD”

10/2019
Scaffold

Sai P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, A. T. Suresh
SCAFFOLD: Stochastic Controlled Averaging for Federated Learning
ICML 2020

11/2020
S-Local-GD, Local-GD*
S-Local-SVRG

Eduard Gorbunov, Filip Hanzely and P.R.
Local SGD: Unified Theory and New Efficient Methods
AISTATS 2021

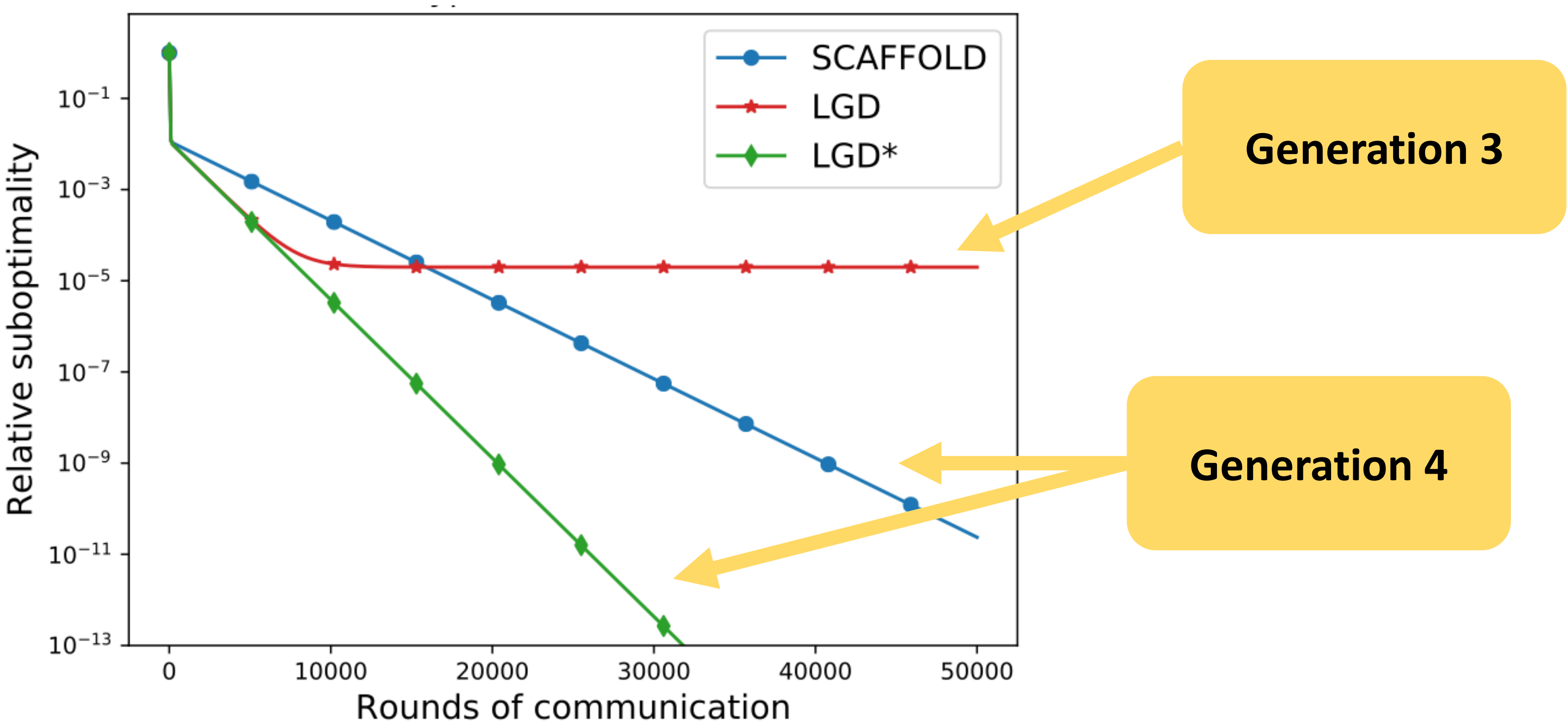
Method	a_i^k, b_i^k, l_i^k	Complexity	Setting	Sec
Local-SGD, Alg. 1 (Woodworth et al., 2020a)	$f_{\xi_i}(x_i^k), 0, -$	$\frac{L}{\mu} + \frac{\sigma^2}{n\mu\varepsilon} + \sqrt{\frac{L\tau(\sigma^2+\tau\zeta^2)}{\mu^2\varepsilon}}$	UBV, ζ-Het	G.1.1
Local-SGD, Alg. 1 (Koloskova et al., 2020)	$f_{\xi_i}(x_i^k), 0, -$	$\frac{\tau L}{\mu} + \frac{\sigma^2}{n\mu\varepsilon} + \sqrt{\frac{L(\tau-1)(\sigma^2+(\tau-1)\zeta_*^2)}{\mu^2\varepsilon}}$	UBV, Het	G.1.1
Local-SGD, Alg. 1 (Khaled et al., 2020)♣	$f_{\xi_i}(x_i^k), 0, -$	$\frac{L+\mathcal{L}/n+\sqrt{(\tau-1)L\mathcal{L}}}{\mu} + \frac{\sigma_*^2}{n\mu\varepsilon} + \frac{L\zeta^2(\tau-1)}{\mu^2\varepsilon} + \sqrt{\frac{L(\tau-1)(\sigma_*^2+\zeta_*^2)}{\mu^2\varepsilon}}$	ES, ζ-Het	G.1.2
Local-SGD, Alg. 1 (Khaled et al., 2020)♣	$f_{\xi_i}(x_i^k), 0, -$	$\frac{L\tau+\mathcal{L}/n+\sqrt{(\tau-1)L\mathcal{L}}}{\mu} + \frac{\sigma_*^2}{n\mu\varepsilon} + \sqrt{\frac{L(\tau-1)(\sigma_*^2+(\tau-1)\zeta_*^2)}{\mu^2\varepsilon}}$	ES, Het	G.1.2
Local-SVRG, Alg. 2 (NEW)	$\nabla f_{i,j_i}(x_i^k) - \nabla f_{i,j_i}(y_i^k) + \nabla f_i(y_i^k)$, $0, -$	$m + \frac{L+\max L_{ij}/n+\sqrt{(\tau-1)L\max L_{ij}}}{\mu} + \frac{L\zeta^2(\tau-1)}{\mu^2\varepsilon} + \sqrt{\frac{L(\tau-1)\zeta_*^2}{\mu^2\varepsilon}}$	simple, ζ-Het	G.2
Local-SVRG, Alg. 2 (NEW)	$\nabla f_{i,j_i}(x_i^k) - \nabla f_{i,j_i}(y_i^k) + \nabla f_i(y_i^k)$, $0, -$	$m + \frac{L\tau+\max L_{ij}/n+\sqrt{(\tau-1)L\max L_{ij}}}{\mu} + \sqrt{\frac{L(\tau-1)^2\zeta_*^2}{\mu^2\varepsilon}}$	simple, Het	G.2
S*-Local-SGD, Alg. 3 (NEW)	$f_{\xi_i}(x_i^k), \nabla f_i(x^*), -$	$\frac{\tau L}{\mu} + \frac{\sigma^2}{n\mu\varepsilon} + \sqrt{\frac{L(\tau-1)\sigma^2}{\mu^2\varepsilon}}$	UBV, Het	G.3
SS-Local-SGD, Alg. 4 (Karimireddy et al., 2019a)	$f_{\xi_i}(x_i^k), h_i^k - \frac{1}{n}\sum_{i=1}^n h_i^k$, $\nabla f_{\tilde{\xi}_i^k}(y_i^k)$	$\frac{L}{p\mu} + \frac{\sigma^2}{n\mu\varepsilon} + \sqrt{\frac{L(1-p)\sigma^2}{p\mu^2\varepsilon}}$	UBV, Het	G.4.1
SS-Local-SGD, Alg. 4 (NEW)	$f_{\xi_i}(x_i^k), h_i^k - \frac{1}{n}\sum_{i=1}^n h_i^k$, $\nabla f_{\tilde{\xi}_i^k}(y_i^k)$	$\frac{L}{p\mu} + \frac{\mathcal{L}}{n\mu} + \frac{\sqrt{L\mathcal{L}(1-p)}}{p\mu} + \frac{\sigma_*^2}{n\mu\varepsilon} + \sqrt{\frac{L(1-p)\sigma_*^2}{p\mu^2\varepsilon}}$	ES, Het	G.4.2
S*-Local-SGD*, Alg. 5 (NEW)	$\nabla f_{i,j_i}(x_i^k) - \nabla f_{i,j_i}(x^*) + \nabla f_i(x^*), \nabla f_i(x^*), -$	$\left(\frac{\tau L}{\mu} + \frac{\max L_{ij}}{n\mu} + \frac{\sqrt{(\tau-1)L\max L_{ij}}}{\mu}\right)\log\frac{1}{\varepsilon}$	simple, Het	G.5
S-Local-SVRG, Alg. 6 (NEW)	$\nabla f_{i,j_i}(x_i^k) - \nabla f_{i,j_i}(y^k) + \nabla f_i(y^k)$, $h_i^k - \frac{1}{n}\sum_{i=1}^n h_i^k, \nabla f_i(y^k)$	$\left(m + \frac{L}{p\mu} + \frac{\max L_{ij}}{n\mu} + \frac{\sqrt{L\max L_{ij}(1-p)}}{p\mu}\right)\log\frac{1}{\varepsilon}$	simple, Het	G.6

02/2021
FedLin

Aritra Mitra, Rayana Jaafar, George J. Pappas, Hamed Hassani
Linear Convergence in Federated Learning: Tackling Client Heterogeneity & Sparse Gradients
NeurIPS 2021

Brief History of Local Training Methods
Generation 4: Linear

“Heterogeneous data is allowed, but the rate ay best matches that of GD”



Part III
5th Generation of Local Training Methods

Brief History of Local Training Methods
Generation 5: Accelerated

"Communication complexity is better than GD for heterogeneous data"

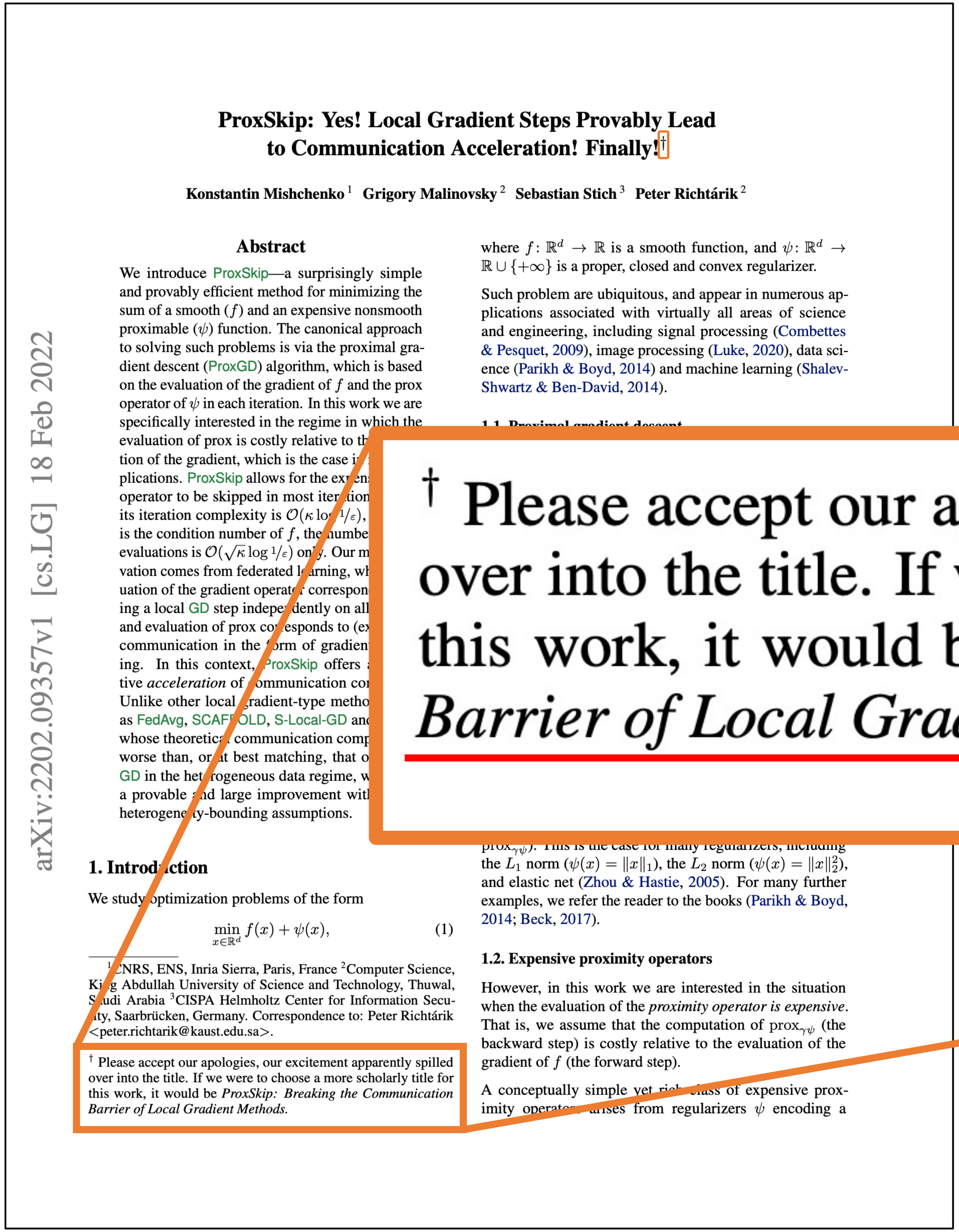
?

In practice, local training significantly improves communication efficiency.

However, there is no theoretical result explaining this!

Is the situation hopeless, or can we show/prove that local training helps?

The Beginning


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… tion of the gradient, which is the case i… plications. ProxSkip allows for the expens… operator to be skipped in most iteration… its iteration complexity is $\mathcal{O}(\kappa \log 1/\varepsilon)$, … is the condition number of f, the number … evaluations is $\mathcal{O}(\sqrt{\kappa} \log 1/\varepsilon)$ only. Our m… vation comes from federated learning, wh… uation of the gradient operator correspon… ing a local GD step independently on all … and evaluation of prox corresponds to (ex… communication in the form of gradient … ing. In this context, ProxSkip offers … tive *acceleration* of communication co… Unlike other local gradient-type metho… as FedAvg, SCAFFOLD, S-Local-GD and … whose theoretical communication comp… worse than, or at best matching, that o… GD in the heterogeneous data regime, w… a provable and large improvement with… heterogeneity-bounding assumptions.

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.*

where $f : \mathbb{R}^d \to \mathbb{R}$ is a smooth function, and $\psi : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ 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).

…

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 $\text{prox}_{\gamma\psi}$ (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

† 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.*

ICML
International Conference
On Machine Learning

PDF

Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich and P.R.
ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication Acceleration! Finally!
ICML 2022

Brief History of Local Training Methods
Generation 5: Accelerated

"Communication complexity is better than GD for heterogeneous data"

02/2022
ProxSkip

PDF

Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich and P.R.
ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication Acceleration! Finally!
ICML 2022

07/2022
APDA; APDA-Inexact

PDF

Abdurakhmon Sadiev, Dmitry Kovalev and P.R.
Communication Acceleration of Local Gradient Methods via an Accelerated Primal-Dual Algorithm with Inexact Prox
NeurIPS 2022

07/2022
ProxSkip-LSVRG

PDF

Grigory Malinovsky, Kai Yi and P.R.
Variance Reduced ProxSkip: Algorithm, Theory and Application to Federated Learning
NeurIPS 2022

07/2022
RandProx

PDF

Laurent Condat and P.R.
RandProx: Primal-Dual Optimization Algorithms with Randomized Proximal Updates
arXiv:2207.12891, 2022

Brief History of Local Training Methods
Generation 5: Accelerated

"Communication complexity is better than GD for heterogeneous data"

10/2022
GradSkip

Artavazd Maranjyan, Mher Safaryan and P.R.
GradSkip: Communication-Accelerated Local Gradient Methods with Better Computational Complexity
arXiv:2210.16402, 2022

10/2022
Compressed-Scaffnew

Laurent Condat, Ivan Agarský and P.R.
Provably Doubly Accelerated Federated Learning: The First Theoretically Successful Combination of Local Training and Compressed Communication
arXiv:2210.13277, 2022

10/2022
5GCS

Michal Grudzien, Grigory Malinovsky and P.R.
Can 5th Generation Local Training Methods Support Client Sampling? Yes!
preprint, 2022

Brief History of Local Training Methods

Generation 5: Accelerated

	Comm. Acceleration	Local Optimizer	# Local Training Steps	Total Complexity (Comm. + Compute)	Client Sampling?	Comm. Compression?	Supports Decentralized Setup?	Key Insight
ProxSkip 2/22, ICML 22	✔ $\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$	GD	$\sqrt{\frac{L}{\mu}}$	=	✘	✘	✔	First 5th generation local training method
APDA-Inexact 7/22, NeurIPS 22	✔ $\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$	any	better	better	✘	✘	✔	Can use more powerful local solvers which take fewer local GD-type steps
VR-ProxSkip 7/22, NeurIPS 22	✔ worse	VR-SGD	worse	better	✘	✘	✘	Running variance reduced SGD locally can lead to better total complexity than ProxSkip
RandProx 7/22	✔ $\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$	GD	$\sqrt{\frac{L}{\mu}}$	=	✘	✘	✔	ProxSkip = VR mechanism for compressing the prox
GradSkip 10/22	✔ $\mathcal{O}\left(\sqrt{\frac{L}{\mu}}\log\frac{1}{\varepsilon}\right)$	GD	better	better	✘	✘	✘	Workers containing less imortant data can do fewer local training steps!
Compressed Scaffnew 10/22	✔ worse	GD	worse	better	✘	✔	✘	Can compress uplink, leads to better overal communication complexity than ProxSkip.
5GCS 10/22	✔ worse	any	$\sqrt{\frac{L}{\mu}}$	worse	✔	✘	✘	Can do client sampling

Part IV
ProxSkip: Local Training Provably Leads to Communication Acceleration

Konstantin Mishchenko, Grigory Malinovsky, Sebastian Stich and P.R.
ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication Acceleration! Finally!
ICML 2022

Consensus Reformulation

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

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

$$\Updownarrow$$

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

Good: Indicator function of a nonempty closed convex set

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

ProxSkip: Bird's Eye View

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

1

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

2a

with probability $1 - p$ do

$1 - p \approx 1$

$x_{t+1} = \hat{x}_{t+1}$

$h_{t+1} = h_t$

2b

with probability p do

$p \approx 0$

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

$x_{t+1} = ?$

$h_{t+1} = ?$

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 $\frac{1}{n}\sum_{i=1}^{n} \|\nabla f_i(x)\|^2 \leq G^2 + 2LB^2\left(f(x) - f_\star\right), \forall x$.

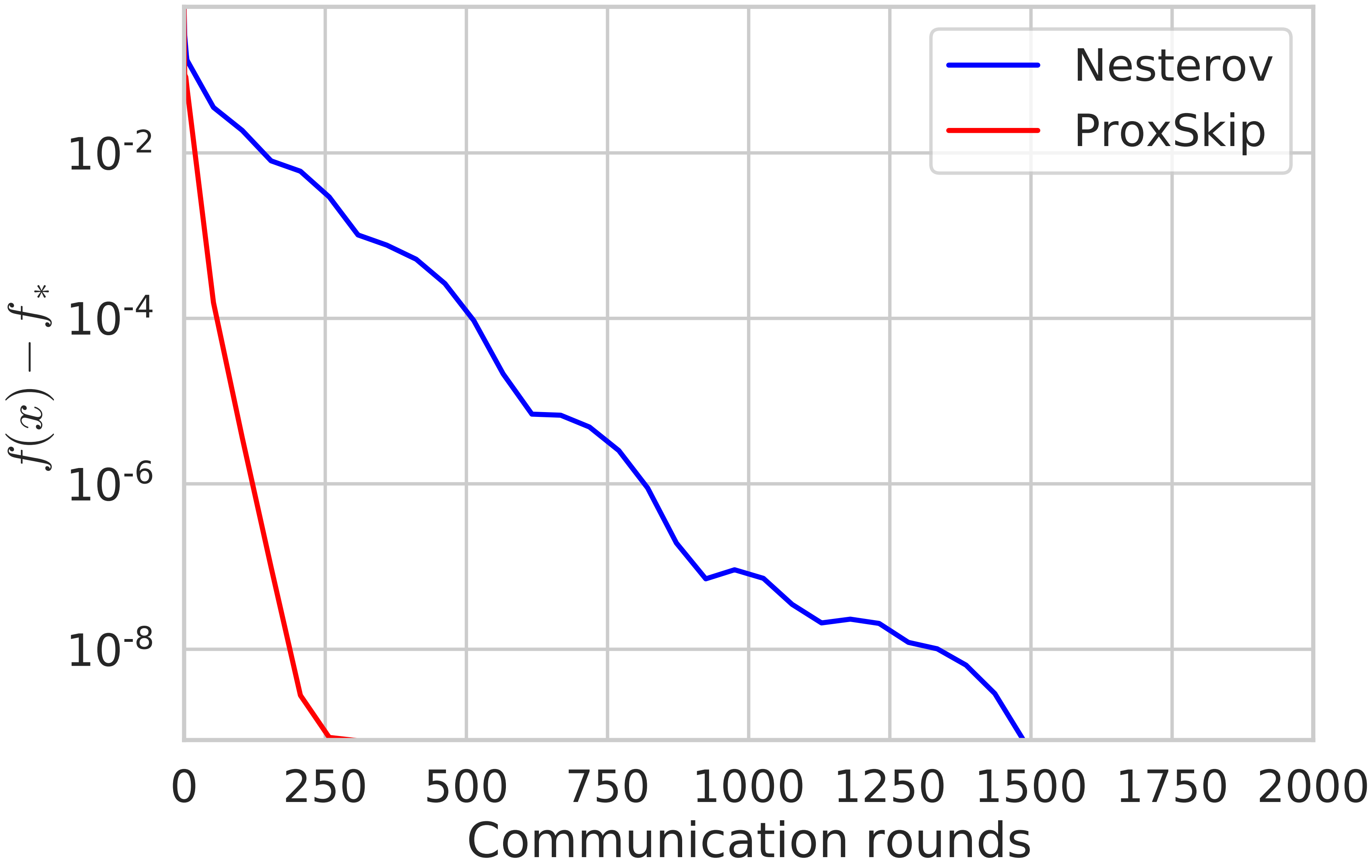
[e] We use Scaffold's cumulative local-global stepsize $\eta_l \eta_g$ for a fair comparison.

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

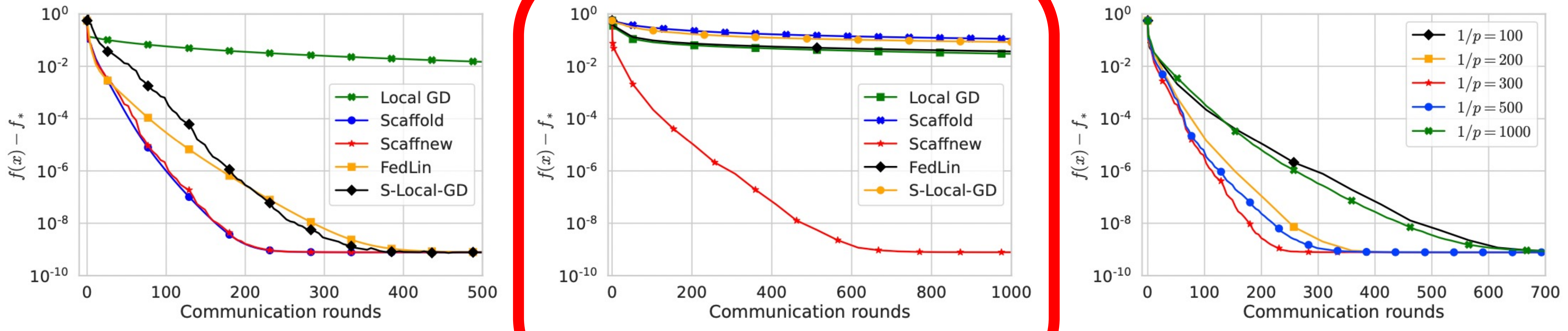
[g] Scaffnew (Algorithm 2) = ProxSkip (Algorithm 1) applied to the consensus formulation (6) + (7) of the finite-sum problem (5).

[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.

ProxSkip vs Nesterov



ProxSkip + Deterministic Gradients



(a) tuned hyper-parameters (b) theoretical hyper-parameters (c) different options of p

Figure 1. **Deterministic Case**. 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 communication rounds with the algorithm parameters set to the best theoretical stepsizes used in the convergence proofs. In (c), we compare communication rounds with the algorithm stepsize set to the best theoretical stepsize and different options of parameter p.

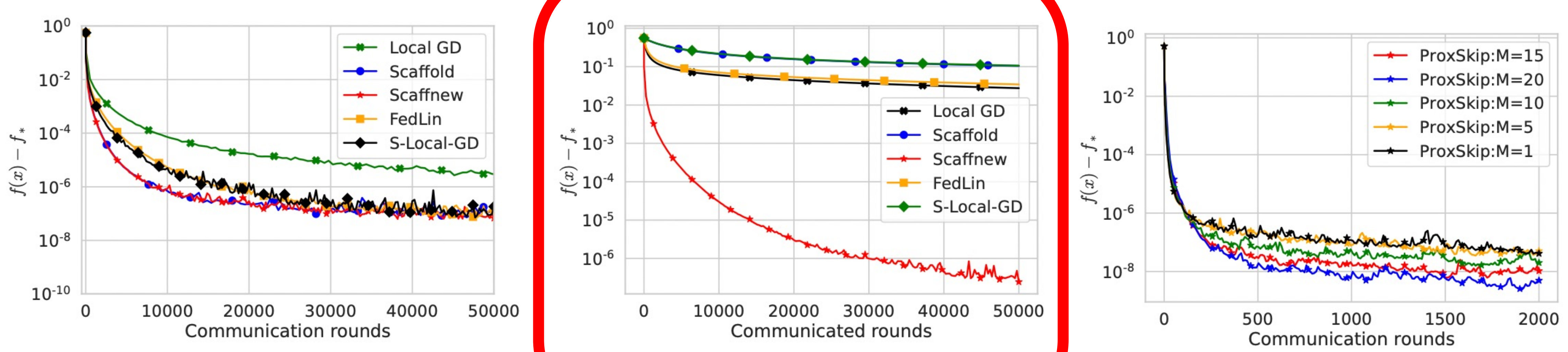
L2-regularized logistic regression:

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

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

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

ProxSkip + Stochastic Gradients



(a) tuned hyper-parameters (b) theoretical hyper-parameters (c) different number of clients

Figure 2. **Stochastic Case**. 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 communication rounds with the algorithm parameters set to the best theoretical stepsizes used in the convergence proofs. In (c), we compare communication rounds with the algorithm parameters set to the best theoretical stepsizes used in the convergence proofs and different number of clients.

L2-regularized logistic regression:

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

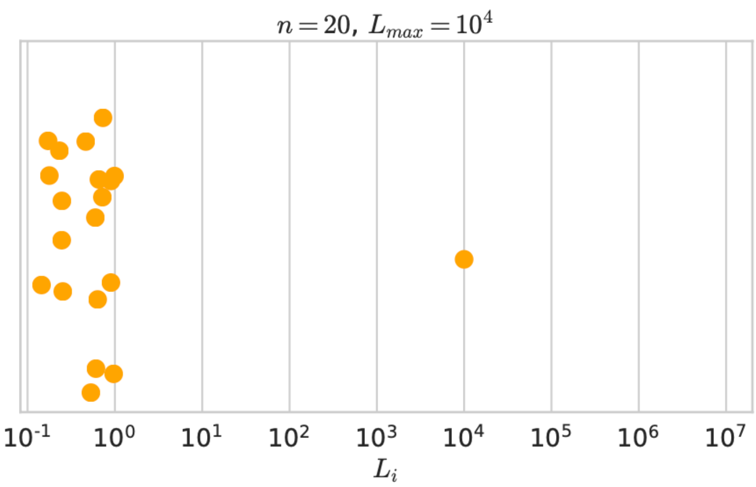
$$a_i \in \mathbb{R}^d,\ b_i \in \{-1,+1\},\ \lambda = L/10^4$$

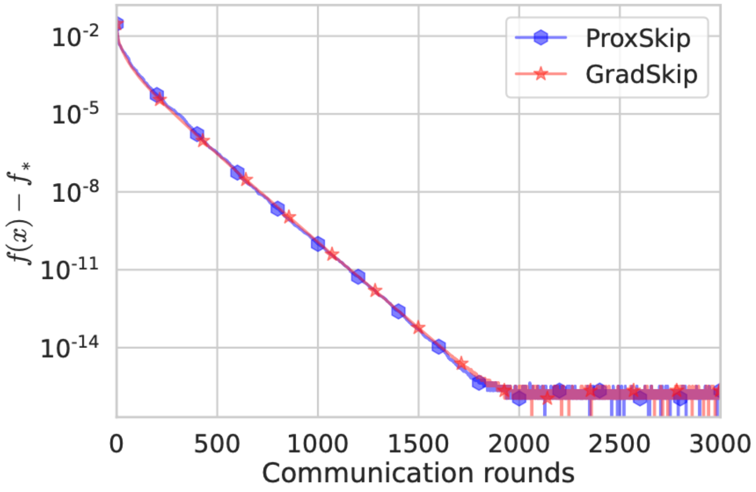
`w8a` dataset from LIBSVM library (Chang & Lin, 2011)

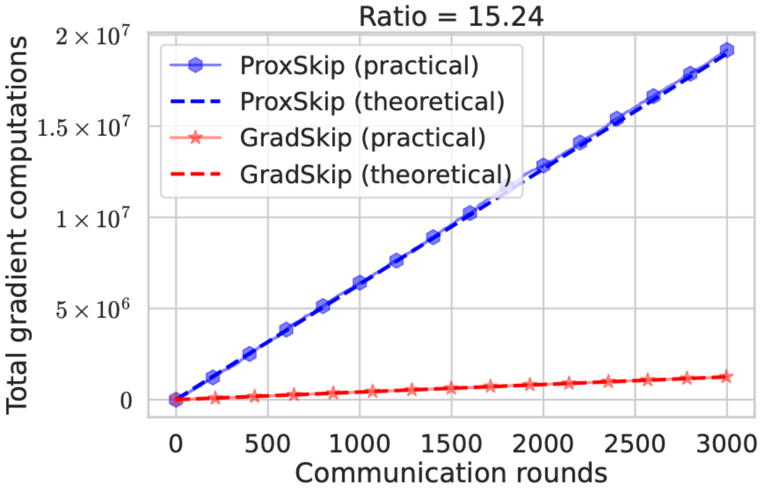
Part V
GradSkip: Clients with Less Important Data can do Less Local Training

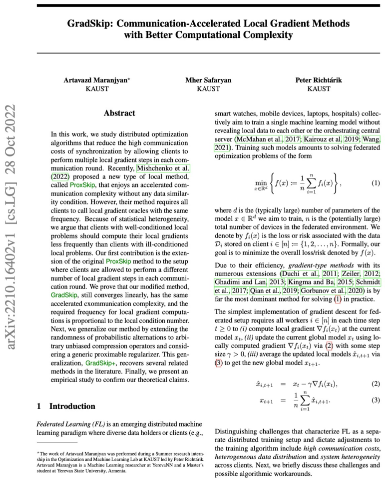
Artavazd Maranjyan, Mher Safaryan and P.R.
GradSkip: Communication-Accelerated Local Gradient Methods with Better Computational Complexity
arXiv:2210.16402, 2022

GradSkip









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

Artavazd Maranjyan, Mher Safaryan, Peter Richtárik — KAUST

arXiv:2210.16402v1 [cs.LG] 28 Oct 2022

Algorithm 2 GradSkip+

1: **Parameters:** stepsize $\gamma > 0$, compressors $\mathcal{C}_\omega \in \mathbb{B}^d(\omega)$ and $\mathcal{C}_{\mathbf{\Omega}} \in \mathbb{B}^d(\mathbf{\Omega})$.

2: **Input:** initial iterate $x_0 \in \mathbb{R}^d$, initial control variate $h_0 \in \mathbb{R}^d$, number of iterations $T \geq 1$.

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

4: $\hat{h}_{t+1} = \nabla f(x_t) - (\mathbf{I} + \mathbf{\Omega})^{-1} \mathcal{C}_{\mathbf{\Omega}} \left(\nabla f(x_t) - h_t\right)$ ◇ Update the shift $\hat{h}_{i,t}$ via shifted compression

5: $\hat{x}_{t+1} = x_t - \gamma(\nabla f(x_t) - \hat{h}_{t+1})$ ◇ Update the iterate $\hat{x}_{i,t}$ via shifted gradient step

6: $\hat{g}_t = \frac{1}{\gamma(1+\omega)} \mathcal{C}_\omega \left(\hat{x}_{t+1} - \mathrm{prox}_{\gamma(1+\omega)\psi}\left(\hat{x}_{t+1} - \gamma(1+\omega)\hat{h}_{t+1}\right)\right)$ ◇ Estimate the proximal gradient

7: $x_{t+1} = \hat{x}_{t+1} - \gamma \hat{g}_t$ ◇ Update the main iterate $x_{i,t}$

8: $h_{t+1} = \hat{h}_{t+1} + \frac{1}{\gamma(1+\omega)}(x_{t+1} - \hat{x}_{t+1})$ ◇ Update the main shift $h_{i,t}$

9: **end for**

The End

Appendix A
Consensus Reformulation of Federated Learning

Optimization Formulation of Federated Learning

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

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

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

Good: Indicator function of a nonempty closed convex set

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

Consensus Reformulation

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

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

$$\Updownarrow$$

Bad: non-differentiable

Good: proper closed convex

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

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

$\psi(x_1, \ldots, x_n) : \mathbb{R}^{nd} \to \mathbb{R} \cup \{+\infty\}$
is a proper closed convex function

$\text{epi}(\psi) \overset{\text{def}}{=} \{(x,t) \mid \psi(x) \le t\}$ The epigraph of ψ is a closed and convex set

Appendix B
Proximal Gradient Descent

Three Assumptions

The epigraph of ψ is a closed and convex set

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

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

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 $\mathrm{prox}_\psi : \mathbb{R}^d \to \mathbb{R}^d$ defined by

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

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

Key Method: Proximal Gradient Descent

proximal operator:

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

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:

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

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

Theorem:

$$t \geq \frac{L}{\mu}\log\frac{1}{\varepsilon} \quad \Rightarrow \quad \|x_t - x_\star\|^2 \leq \varepsilon\|x_0 - x_\star\|^2$$

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

iterations

Error tolerance

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

Appendix C
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

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

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

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

$x_\star$ 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

$1 - p \approx 1$

$x_{t+1} = \hat{x}_{t+1}$

$h_{t+1} = h_t$

2b

with probability p do

$p \approx 0$

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

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

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

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

Appendix D
ProxSkip Theory

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

Consensus Reformulation

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

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

$$\Updownarrow$$

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

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

Bad: non-differentiable

Good: Indicator function of a nonempty closed convex set

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

Three Assumptions

The epigraph of ψ is a closed and convex set

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

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

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 $\mathrm{prox}_\psi : \mathbb{R}^d \to \mathbb{R}^d$ defined by

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

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

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

$1 - p \approx 1$

$x_{t+1} = \hat{x}_{t+1}$

$h_{t+1} = h_t$

2b

with probability p do

$p \approx 0$

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

$x_{t+1} = ?$

$h_{t+1} = ?$

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:

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

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:

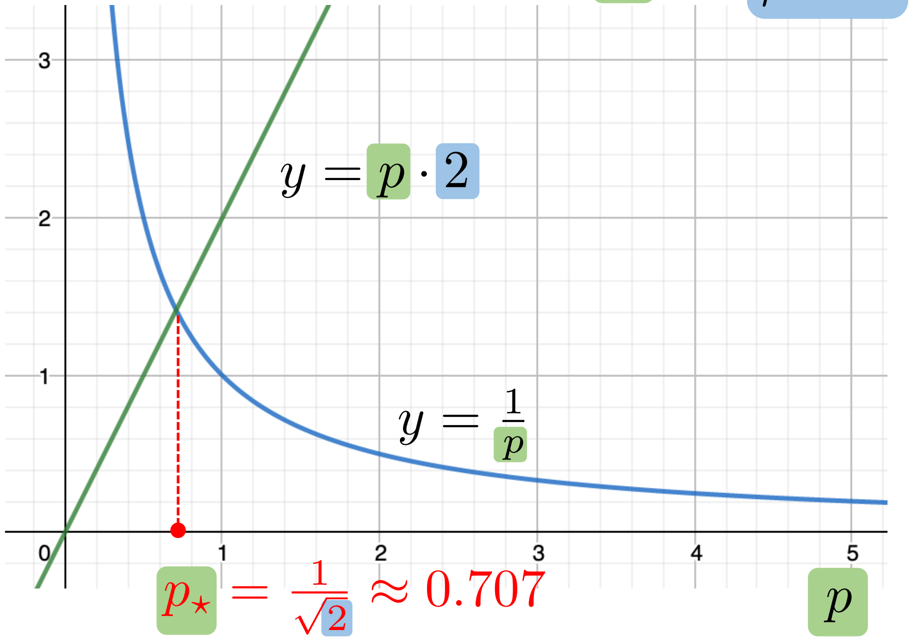
$\frac{L}{\mu}$ is the condition number of f

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

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 $\frac{1}{n}\sum_{i=1}^{n}\|\nabla f_i(x)\|^2 \leq G^2 + 2LB^2\left(f(x) - f_\star\right), \forall x$.

[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.

Appendix E
Extensions

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

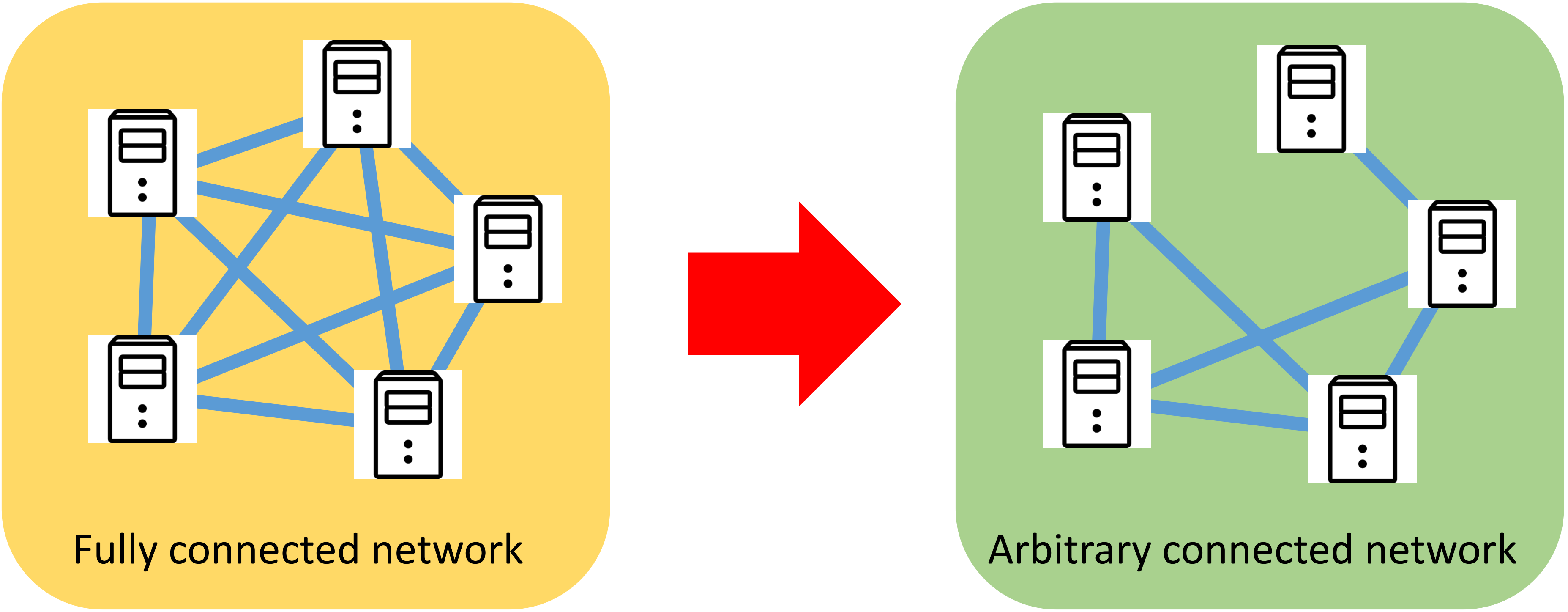
Assumptions:

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

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

From Fully Connected Networks to Arbitrary Connected Networks

- 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.**



The End
(for real)