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

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

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 the evaluation of the gradient, which is the case in many applications. ProxSkip allows for the expensive prox operator to be skipped in most iterations: while its iteration complexity is $\mathcal{O}(\kappa \log 1/\varepsilon)$, where κ is the condition number of f, the number of prox evaluations is $\mathcal{O}(\sqrt{\kappa} \log 1/\varepsilon)$ only. Our main motivation comes from federated learning, where evaluation of the gradient operator corresponds to taking a local GD step independently on all devices, and evaluation of prox corresponds to (expensive) communication in the form of gradient averaging. In this context, ProxSkip offers an effective *acceleration* of communication complexity. Unlike other local gradient-type methods, such as FedAvg, SCAFFOLD, S-Local-GD and FedLin, whose theoretical communication complexity is worse than, or at best matching, that of vanilla GD in the heterogeneous data regime, we obtain a provable and large improvement without any heterogeneity-bounding assumptions.

1. Introduction

We study optimization problems of the form

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

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).

1.1. Proximal gradient descent

One of the most canonical methods for solving (1), often used as the basis for further extensions and improvements, is proximal gradient descent (ProxGD), also known as the forward-backward algorithm (Combettes & Pesquet, 2009; Nesterov, 2013). This method solves (1) via the iterative process defined by

$$x_{t+1} = \mathrm{prox}_{\gamma_t\psi}(x_t - \gamma_t \nabla f(x_t)), \tag{2}$$

where $\gamma_t > 0$ is a suitably chosen stepsize at time t, and $\mathrm{prox}_{\gamma\psi}(\cdot)\colon \mathbb{R}^d \to \mathbb{R}^d$ is the proximity operator of ψ, defined via

$$\mathrm{prox}_{\gamma\psi}(x) := \arg\min_{y\in\mathbb{R}^d} \left[\frac{1}{2}\|y-x\|^2 + \gamma\psi(y)\right]. \tag{3}$$

It is typically assumed that the proximity operator (3) can be evaluated in closed form, which means that the iteration (2) defining ProxGD can be performed exactly. ProxGD is most suited to situations when the proximity operator is relatively cheap to evaluate, so that the bottleneck of (2) is in the forward step (i.e., computation of the gradient ∇f) rather than in the backward step (i.e., computation of $\mathrm{prox}_{\gamma\psi}$). This is the case for many regularizers, including 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

[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.*

"complicated-enough" nonempty constraint set $\mathcal{C} \subset \mathbb{R}^d$ via

$$\psi(x) = \begin{cases} 0 & x \in \mathcal{C} \\ +\infty & x \notin \mathcal{C} \end{cases}. \tag{4}$$

The evaluation of the proximity operator of ψ given by (4) reduces to Euclidean projection onto $\mathcal{C}$,

$$\mathrm{prox}_{\gamma\psi}(x) = \arg\min_{y \in \mathcal{C}} \|y - x\|,$$

which can be a difficult optimization problem on its own. For instance, this is the case when $\mathcal{C}$ is a polyhedral or a spectral set (Parikh & Boyd, 2014).[1]

1.3. Distributed machine learning and consensus constraints

An important example of expensive proximity operators associated with indicator functions (4) arise in the *consensus* formulation of distributed optimization problems. In particular, consider the problem of minimizing the average of n functions using a cluster of n compute nodes/clients,

$$\min_{x \in \mathbb{R}^d} \left\{ f(x) := \frac{1}{n} \sum_{i=1}^{n} f_i(x) \right\}, \tag{5}$$

where function $f_i \colon \mathbb{R}^d \to \mathbb{R}$, and the data describing it, is owned by and stored on client $i \in [n] := \{1, 2, \dots, n\}$. This problem is of key importance in machine learning as it is an abstraction of the *empirical risk minimization* (Shalev-Shwartz & Ben-David, 2014), which is currently the dominant paradigm for training supervised machine learning models.

By cloning the model $x \in \mathbb{R}^d$ into n independent copies $x_1, \dots, x_n \in \mathbb{R}^d$, problem (5) can be reformulated into the *consensus form* (see e.g. Parikh & Boyd, 2014)

$$\min_{x_1, \dots, x_n \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} f_i(x_i) + \psi(x_1, \dots, x_n), \tag{6}$$

where the regularizer $\psi \colon \mathbb{R}^{nd} \to \mathbb{R}$ given by

$$\psi(x_1, \dots, x_n) := \begin{cases} 0, & \text{if } x_1 = \dots = x_n, \\ +\infty, & \text{otherwise,} \end{cases} \tag{7}$$

encodes the consensus constraint

$$\mathcal{C} := \{(x_1, \dots, x_n) \in \mathbb{R}^{nd} \;:\; x_1 = \dots = x_n\}.$$

Evaluating the proximity operator of (7) is not computationally expensive as it simply amounts to taking the average of the variables (Parikh & Boyd, 2014):

$$\mathrm{prox}_{\gamma\psi}(x_1, \dots, x_n) = (\bar{x}, \dots, \bar{x}) \in \mathbb{R}^{nd}, \tag{8}$$

where

$$\bar{x} := \frac{1}{n} \sum_{i=1}^{n} x_i. \tag{9}$$

However, it often involves *high communication cost* since the vectors $x_1, \dots, x_n$ are stored on different compute nodes. Indeed, even simple averaging can be very time consuming if the communication links connecting the clients (e.g., through an orchestrating server) are slow and the dimension d of the aggregated vectors/models high, which is the case in *federated learning (FL)* (Konečný et al., 2016; Kairouz et al., 2021).

1.4. Federated learning

For the above reasons, practical FL algorithms generally use various communication-reduction mechanisms to achieve a useful computation-to-communication ratio, such as delayed communication. That is, the methods perform multiple local steps independently, based on their local objective (Mangasarian & Solodov, 1994; McDonald et al., 2010; Zhang et al., 2016; McMahan et al., 2016; Stich, 2019; Lin et al., 2018).

However, when all the local functions f_i are *different* (i.e., when each individual machine has data drawn from a different distribution), local steps introduce a *drift* in the updates of each client, which results in convergence issues. Indeed, even in the case of the simplest local gradient-type method, LocalGD, a theoretical understanding that would not require any data similarity/homogeneity assumptions eluded the community for a long time. A resolution was found only recently (Khaled et al., 2019; 2020; Koloskova et al., 2020). However, the rates obtained in these works paint a pessimistic picture for LocalGD; for example, due to client drift, they are sublinear even for smooth and strongly convex problems.

The next task for the FL community was to propose algorithmic adjustments that could provably mitigate the client drift issue. A handful of recent methods, including Scaffold (Karimireddy et al., 2020), S-Local-GD (Gorbunov et al., 2021) and FedLin (Mitra et al., 2021), managed to do that. For instance, under the assumption that f is L-smooth and μ-strongly convex, with condition number $\kappa = {}^{L}/_{\mu}$, Scaffold, S-Local-GD and FedLin obtain a $\mathcal{O}(\kappa \log {}^{1}/_{\varepsilon})$ communication complexity, which matches the communication complexity of GD (that computes a single gradient on every client per round of communication). However, and despite the empirical superiority of these methods over vanilla GD, their theoretical communication complexity does *not* improve upon GD. This reveals a fundamental gap in our understanding of local methods.

Due to the enormous effort that was exerted over the last several years by the FL community in this direction without it

[1] Other examples of expensive proximity operators include Schatten-p norms of matrices (e.g., the nuclear norm), and certain variants of quadratic support functions (Friedlander & Goh, 2016).

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 $p \in \left(\frac{1}{\kappa}, 1\right)$
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.

bearing the desired fruit (Kairouz et al., 2021), it seems very challenging to establish theoretically that performing independent local updates improves upon the communication complexity of GD. In contrast, accelerated gradient descent (without local steps) can reach the optimal $\mathcal{O}(\sqrt{\kappa}\log{}^1\!/\!_\varepsilon)$ communication complexity (Lan, 2012; Woodworth et al., 2020b; 2021).

This raises the question of whether this is a fundamental limitation of local methods. Is it possible to prove a better communication complexity than $\mathcal{O}(\kappa \log{}^1\!/\!_\varepsilon)$ for *simple* local gradient-type methods, without resorting to any explicit acceleration mechanisms?

2. Contributions

We now summarize the main contributions of this work.

2.1. ProxSkip: a general prox skipping algorithm

We develop a new ProxGD-like algorithm for solving the general regularized problem (1). Our method, which we call ProxSkip (see Algorithm 1), is designed to handle expensive proximal operators.

A key ingredient in its design is a *randomized prox-skipping procedure*: in each iteration of ProxSkip, we evaluate the proximity operator with probability $p \in (0, 1]$. If $p = 1$, several steps in our method are vacuous, and we recover ProxGD as a special case (and the associated standard theory). Of course, the interesting choice is $0 < p < 1$. In expectation, the proximity operator is evaluated every ${}^1\!/\!_p$ iterations, which can be very rare if p is small.

Control variates stabilizing prox skipping. We had to introduce several new algorithmic design adjustments for such a method to provably work. In particular, ProxSkip uses a control variate h_t on line 3 to shift the gradient $\nabla f(x_t)$ when the forward step is performed.

Note that h_t stays constant in between two consecutive prox calls. Indeed, this is because in that case we have $x_{t+1} = \hat{x}_{t+1}$ from line 8, and line 10 therefore simplifies to $h_{t+1} = h_t$. So, when operating in between two prox calls, our method performs iterations of the form

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

where $\gamma > 0$ is a stepsize parameter. When a prox step is executed, both the iterate x_t and the control variate h_t are adjusted, and the process is repeated.

This control mechanism is necessary to allow for prox-skipping to work. To illustrate this, consider an optimal point $x_\star = \operatorname{argmin}_x f(x) + \psi(x)$. In general, it does not hold $\nabla f(x_\star) = 0$, so skipping the prox (without control variate adjustment) would imply a *drift away* from $x_\star$. We show below that the control variate converges to

$$h_t \to \nabla f(x_\star),$$

which means that $x_\star$ is a fixed point. This allows skipping the prox for a significant amount of steps without impacting the convergence.

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 $\mathrm{Prob}(\theta_t = 1) = p$    ◇ 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
```

Theory. If f is L-smooth and μ-strongly convex, we prove that ProxSkip converges at a linear rate. In particular, we show that after T iterations,

$$\mathbb{E}\left[\Psi_T\right] \leq (1 - \min\{\gamma\mu, p^2\})^T \Psi_0,$$

where Ψ_t is a certain Lyapunov function (see (11)) involving both x_t and h_t. If we choose $\gamma = {}^1/_L$ and $p = {}^1/_{\sqrt{\kappa}}$, where $\kappa = {}^L/_\mu$ is the condition number, then the iteration complexity of ProxSkip is $\mathcal{O}(\kappa \log {}^1/_\varepsilon)$, whereas the number of prox evaluations (in expectation) is $\mathcal{O}(\sqrt{\kappa} \log {}^1/_\varepsilon)$ only! For more details related to theory, see Section 3.

2.2. Scaffnew: ProxSkip applied to federated learning

When applied to the consensus reformulation (6)–(7) of problem (5), ProxSkip can be interpreted as a new distributed gradient-type method performing local steps, adding to the existing rich literature on local methods. In this context, we decided to call our method Scaffnew (Algorithm 2).[2] Since prox evaluation now means communication via averaging across the nodes (see (8) and (9)), and since Scaffnew inherits the strong theoretical prox-skipping properties of its parent method ProxSkip:

> *We resolve one of the most important open problems in the FL literature: breaking the $\mathcal{O}(\kappa \log {}^1/_\varepsilon)$ communication complexity barrier with a simple local method. In particular,* Scaffnew *reaches an $\mathcal{O}(\sqrt{\kappa} \log {}^1/_\varepsilon)$ communication complexity without imposing any additional assumptions (e.g., data similarity or stronger smoothness assumptions).*

Note that since the iteration complexity of Scaffnew is $\mathcal{O}(\kappa \log {}^1/_\varepsilon)$, the number of local steps per communication round is (on average) $\mathcal{O}(\sqrt{\kappa})$. According to Arjevani & Shamir (2015), the communication lower bound for first order distributed algorithms is $\mathcal{O}(\sqrt{\kappa} \log {}^1/_\varepsilon)$. This means that Scaffnew is optimal in terms of communication rounds.

Please refer to Table 1 in which we compare our results with the results obtained by existing state-of-the-art methods.

2.3. Extensions

We develop two extensions of the vanilla ProxSkip method; see Section 5. We are not attempting to be exhaustive: these extensions are meant to illustrate that our method and proof technique combine well with other tricks and techniques often used in the literature.

From deterministic to stochastic gradients. First, in Section 5.1 we perform an extension enabling us to use a *stochastic gradient* $g_t(x_t) \approx \nabla f(x_t)$ in ProxSkip instead of the true gradient $\nabla f(x_t)$. This is of importance in many applications, and is of particular importance for our method since now that the cost of the prox step was reduced, the cost of the gradient steps becomes more important. We operate under the modern *expected smoothness* assumption introduced by Gower et al. (2019; 2021), which is less restrictive than the standard bounded variance assumption.

From a central server to fully decentralized training. Second, in Section 5.2 we present and analyze ProxSkip in a fully *decentralized* optimization setting, where the communication between nodes is restricted to a communication graph. Our decentralized algorithm inherits the property that it is not affected by data-heterogeneity. The covariate technique in ProxSkip resembles, to some extent, some of the existing *gradient tracking* mechanisms (Lorenzo & Scutari, 2016; Nedić et al., 2016). However, while gradient tracking provably addresses data-heterogeneity, its communication complexity scales proportional to the iteration complexity, $\mathcal{O}(\kappa)$ (Yuan & Alghunaim, 2021; Koloskova et al., 2021). The same holds for almost all other schemes that have been designed to address data-heterogeneity in decentralized optimization (Tang et al., 2018; Vogels et al., 2021). Notable exceptions include the optimal methods developed by Ko-

[2] This is a homage to the influential Scaffold method of Karimireddy et al. (2020), which in our experiments performs very similarly to Scaffnew if the former method is used with fine-tuned stepsizes.

valev et al. (2020; 2021b;a); see also the references therein. However, these methods are based on classical acceleration schemes, and do not perform multiple local steps.

3. Theory

We are now ready to describe our key theoretical development: the convergence analysis of ProxSkip.

3.1. Assumptions

We rely on several standard assumptions to establish our results. First, we need f to be smooth and strongly convex (see Appendix A for complementary details).

Assumption 3.1. f is L-smooth and μ-strongly convex.

We also need the following standard assumption[3] on the regularizer ψ.

Assumption 3.2. ψ is proper, closed and convex.

These assumption imply that problem (1) has a unique minimizer, which we denote $x_\star := \operatorname{argmin} f(x) + \psi(x)$.

3.2. Firm nonexpansiveness

In one step of our analysis we will rely on firm nonexpansiveness of the proximity operator (see, e.g., Bauschke et al., 2021):

Lemma 3.3. *Let Assumption 3.2 be satisfied. Let $P(x) := \operatorname{prox}_{\frac{\gamma}{p}\psi}(x)$ and $Q(x) := x - P(x)$. Then*

$$\|P(x) - P(y)\|^2 + \|Q(x) - Q(y)\|^2 \leq \|x - y\|^2, \quad (10)$$

for all $x, y \in \mathbb{R}^d$ and any $\gamma, p > 0$.

3.3. Two technical lemmas

The strength of our method comes from the role the control variates h_t play in stabilizing the effect of skipping prox evaluations. Our analysis captures this effect. In particular, a by-product of our analysis is a proof that the control variates converge to $h_\star := \nabla f(x_\star)$, where $x_\star$. In order to show this, we work with the following natural candidate for a Lyapunov function:

$$\Psi_t := \|x_t - x_\star\|^2 + \frac{\gamma^2}{p^2}\|h_t - h_\star\|^2 . \quad (11)$$

We further define

$$w_t := x_t - \gamma\nabla f(x_t), \quad \text{and} \quad w_\star := x_\star - \gamma\nabla f(x_\star). \quad (12)$$

Note that if our method works, i.e., if $x_t \to x_\star$, then gradient smoothness implies that $w_t \to w_\star$. In our first technical lemma, we show that after one step of ProxSkip, the Lyapunov function can be bounded in terms of the distance $\|w_t - w_\star\|^2$ and the control variate error $\|h_t - h_\star\|^2$. It is this lemma in the proof of which we rely on firm nonexpansiveness. We do not use it anywhere else.

Lemma 3.4. *If Assumptions 3.1 and 3.2 hold, $\gamma > 0$ and $0 < p \leq 1$, then*

$$\mathbb{E}\left[\Psi_{t+1}\right] \leq \|w_t - w_\star\|^2 + (1 - p^2)\frac{\gamma^2}{p^2}\|h_t - h_\star\|^2 , \quad (13)$$

where the expectation is taken over the θ_t in Algorithm 1.

Our next lemma bounds the first term in the right-hand side of (36) by a multiple of $\|x_t - x_\star\|^2$.

Lemma 3.5. *Let Assumption 3.1 hold with any $\mu \geq 0$. If $0 < \gamma \leq \frac{1}{L}$, then*

$$\|w_t - w_\star\|^2 \leq (1 - \gamma\mu)\|x_t - x_\star\|^2. \quad (14)$$

3.4. Main theorem

As we shall now see, our main theorem follows simply by combining the last two lemmas.

Theorem 3.6. *Let Assumption 3.1 and Assumption 3.2 hold, and let $0 < \gamma \leq \frac{1}{L}$ and $0 < p \leq 1$. Then, the iterates of ProxSkip (Algorithm 1) satisfy*

$$\mathbb{E}\left[\Psi_T\right] \leq (1 - \zeta)^T \Psi_0, \quad (15)$$

where $\zeta := \min\{\gamma\mu, p^2\}$.

Proof. By combining Lemmas 3.4 and 3.5, we get

$$\begin{aligned}\mathbb{E}\left[\Psi_{t+1}\right] &\leq (1 - \gamma\mu)\|x_t - x_\star\|^2 + (1 - p^2)\frac{\gamma^2}{p^2}\|h_t - h_\star\|^2 \\ &\leq (1 - \zeta)\left(\|x_t - x_\star\|^2 + \frac{\gamma^2}{p^2}\|h_t - h_\star\|^2\right) \\ &= (1 - \zeta)\Psi_t.\end{aligned}$$

We get the theorem's claim by unrolling the recurrence. □

3.5. How often should one skip the prox?

Note that by choosing $p = 1$ (no prox skipping) and $\gamma = 1/L$, we get $\zeta = 1/\kappa$, which leads to the rate $\mathcal{O}(\kappa \log 1/\varepsilon)$ of ProxGD. This is not a surprise since when $p = 1$, ProxSkip *is* identical to ProxGD.

More importantly, note that for any fixed stepsize $\gamma > 0$, the reduction factor $\zeta := \min\{\gamma\mu, p^2\}$ in (15) remains unchanged as we decrease p from 1 down to $p = 1/\sqrt{\gamma\mu}$. This is the reason why we can often *skip the prox*, and get away with it *for free*, i.e., without any deterioration of the convergence rate!

[3] Note that this assumption is automatically satisfied for ψ defined in (7).

Algorithm 2 Scaffnew: Application of ProxSkip to Federated Learning (i.e., to problem (6)–(7))

```
stepsize $\gamma > 0$, probability $p > 0$, initial iterate $x_{1,0} = \cdots = x_{n,0} \in \mathbb{R}^d$, initial control variates $h_{1,0}, \ldots, h_{n,0} \in \mathbb{R}^d$ on each client such that $\sum_{i=1}^n h_{i,0} = 0$, number of iterations $T \geq 1$
server: flip a coin, $\theta_t \in \{0,1\}$, $T$ times, where $\mathrm{Prob}(\theta_t = 1) = p$    ◇ Decide when to skip communication
send the sequence $\theta_0, \ldots, \theta_{T-1}$ to all workers
for $t = 0, 1, \ldots, T-1$ do
    in parallel on all workers $i \in [n]$ do
        $\hat{x}_{i,t+1} = x_{i,t} - \gamma(g_{i,t}(x_{i,t}) - h_{i,t})$    ◇ Local gradient-type step adjusted via the local control variate $h_{i,t}$
        if $\theta_t = 1$ then
            $x_{i,t+1} = \frac{1}{n}\sum_{i=1}^{n} \hat{x}_{i,t+1}$    ◇ Average the iterates, but only very rarely! (with small probability $p$)
        else
            $x_{i,t+1} = \hat{x}_{t+1}$    ◇ Skip communication!
        end if
        $h_{i,t+1} = h_{i,t} + \frac{p}{\gamma}(x_{i,t+1} - \hat{x}_{i,t+1})$    ◇ Update the local control variate $h_{i,t}$
    end local updates
end for
```

By inspecting (15) it is easy to see that

$$T \geq \max\left\{\frac{1}{\gamma\mu}, \frac{1}{p^2}\right\} \log\frac{1}{\varepsilon} \implies \mathbb{E}\left[\Psi_T\right] \leq \varepsilon \Psi_0. \quad (16)$$

Since in each iteration we evaluate the prox with probability p, the *expected number of prox evaluations* is

$$pT \overset{(16)}{\approx} \max\left\{\frac{p}{\gamma\mu}, \frac{1}{p}\right\} \log\frac{1}{\varepsilon}. \quad (17)$$

Clearly, the best result is obtained if we use the largest stepsize allowed by Theorem 3.6:

$$\gamma = \frac{1}{L}. \quad (18)$$

Next, the value of p that minimizes expression (17) satisfies $\frac{pL}{\mu} = \frac{1}{p}$, which gives the *optimal probability*

$$p = \sqrt{\frac{\mu}{L}} = \frac{1}{\sqrt{\kappa}}, \quad (19)$$

where $\kappa := L/\mu$ is the condition number. With these optimal choices of the parameters γ and p, the number of iterations of ProxSkip is

$$T \overset{(16)}{\approx} \max\left\{\frac{1}{\gamma\mu}, \frac{1}{p^2}\right\} \log\frac{1}{\varepsilon} \overset{(18)+(19)}{=} \kappa \log\frac{1}{\varepsilon},$$

and the expected number of prox evaluations performed in the process is

$$pT \overset{(17)}{\approx} \max\left\{\frac{p}{\gamma\mu}, \frac{1}{p}\right\} \log\frac{1}{\varepsilon} \overset{(18)+(19)}{=} \sqrt{\kappa} \log\frac{1}{\varepsilon}.$$

Let us summarize the above findings.

Corollary 3.7. *If we choose $\gamma = 1/L$ and $p = 1/\sqrt{\kappa}$, then the iteration complexity of ProxSkip (Algorithm 1) is $\mathcal{O}(\kappa \log 1/\varepsilon)$ and its prox calculation complexity is $\mathcal{O}(\sqrt{\kappa} \log 1/\varepsilon)$.*

4. Application to Federated Learning

Let us now consider the problem of minimizing the average of n functions stored on n devices, as formulated in (5). This is the canonical problem in federated learning (McMahan et al., 2016; Kairouz et al., 2021).[4] In this setting the functions $f_i \colon \mathbb{R}^d \to \mathbb{R}$ denote the local loss function of client i defined over its own private data. For simplicity, we assume in this section that every client can compute the gradient $\nabla f_i(x)$ exactly (i.e., a full pass over the local data), see Section 5.1 for the discussion of the stochastic setting. When applied to the consensus reformulation (6)–(7) of problem (5), ProxSkip reduces to Scaffnew (Algorithm 2).

Method description. Algorithm 2 has three main steps: local updates to the client model $x_{i,t} \in \mathbb{R}^d$, local updates to the client control variate $h_{i,t} \in \mathbb{R}^d$, and averaging the client models with probability p in every iteration.

When $g_{i,t}(x_{i,t}) = \nabla f_i(x_{i,t})$, then each local update on client i takes the form

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

We will show below that $h_{i,t} \overset{t\to\infty}{\to} \nabla f_i(x_\star)$, so that it becomes evident that the optimal solution $x_\star$ is a fixed point of the algorithm (this is a key differentiation from, e.g., LocalGD (Khaled et al., 2020; Koloskova et al., 2020; Malinovskiy et al., 2020)). The local covariates $h_{i,t}$ are updated after each communication round, i.e., when $\theta_t = 1$, as

$$h_{i,t+1} = h_{i,t} + \frac{p}{\gamma}\underbrace{\left(\frac{1}{n}\sum_{j=1}^{n} \hat{x}_{j,t+1} - \hat{x}_{i,t+1}\right)}_{\text{accumulated `client drift'}}.$$

[4] As our focus is on a new communication-efficient scheme, we disregard here other important aspects such as client sampling.

The local drift (i.e., deviation from the client mean) is divided by the stepsize and the expected length (i.e., $1/p$) of the local phase during which the drift has been accumulated. This drift correction shares similarities with option II in Scaffold (Karimireddy et al., 2020) and QG-DSGD (Lin et al., 2021), yet differs from option I in Scaffold and FedLin (Mitra et al., 2021), that both propose to compute an additional gradient at the client average.

4.1. Convergence

We will need an assumption on the individual functions f_i:

Assumption 4.1. Each f_i is L-smooth and μ-strongly convex.

Note though that we do not need to make any assumption on the similarity of the functions f_i. Convergence of Scaffnew (Algorithm 2) in the deterministic case follows as a corollary of Theorem 3.6.

Corollary 4.2 (Federated Learning)**.** *Let Assumption 4.1 hold and let $\gamma = 1/L$, $p = 1/\sqrt{\kappa}$ and $g_{i,t}(x_{i,t}) = \nabla f_i(x_{i,t})$. Then the iteration complexity of Algorithm 2 is $\mathcal{O}(\kappa \log 1/\varepsilon)$ and its communication complexity is $\mathcal{O}(\sqrt{\kappa} \log 1/\varepsilon)$.*

Usefulness of local steps. Our result shows for the first time a real advantage of local update methods *without imposing any similarity assumptions*. For instance, Woodworth et al. (2020a) assume quadratic functions, Karimireddy et al. (2020; 2021) bounded Hessian dissimilarity, and Yuan & Ma (2020) bounded Hessian. Without any such assumption, we show here that local methods can converge in significantly fewer update rounds than large-batch methods without local steps (Dekel et al., 2012). The method matches the communication-complexity lower bound derived in (Arjevani & Shamir, 2015) and is optimal in this regard. Moreover, and unlike the approach adopted by Hanzely & Richtárik (2020); Hanzely et al. (2020), our improvements do not rely on interpreting local methods as methods for solving *personalized formulations of FL*.

5. Extensions

5.1. Stochastic gradients

In machine learning, calculating full gradients may be extremely expensive and in some cases not possible. In this section, we are going to make an extension of the basic ProxSkip (Algorithm 1) to allow stochastic updates:

$$\hat{x}_{t+1} = x_t - \gamma(g_t(x_t) - h_t). \tag{20}$$

In a generic SGD method, we work with unbiased estimators of gradients only.

Assumption 5.1 (Unbiasedness)**.** For all $t \geq 0$, $g_t(x_t)$ is an unbiased estimator of the gradient $\nabla f(x_t)$. That is,

$$\mathbb{E}\left[g_t(x_t) \mid x_t\right] = \nabla f(x_t). \tag{21}$$

In our analysis of ProxSkip in the stochastic case, we rely on the *expected smoothness* assumption introduced by Gower et al. (2021) in the context of variance reduction, and later adopted and simplified by Gower et al. (2019) in the context if SGD analysis.

Assumption 5.2 (Expected smoothness)**.** There exist constants $A \geq 0$ and $C \geq 0$ such that for all $t \geq 0$,

$$\mathrm{E}\left[\left\|g_t(x_t) - \nabla f\left(x_\star\right)\right\|^2 \mid x_t\right] \leq 2AD_f\left(x_t, x_\star\right) + C. \tag{22}$$

This assumption is satisfied in many practical settings, including when the randomness in g_t arises from subsampling (i.e., minibatching) and compression (Gorbunov et al., 2021). It is also satisfied in the popular but artificial setting when an additive zero mean and bounded variance noise is added to the gradient, formalized next.

Assumption 5.3 (Bounded variance)**.** For all $t \geq 0$, the stochastic estimator $g_t(x_t)$ has bounded variance:

$$\mathrm{Var}[g_t(x_t) \mid x_t] \leq \sigma^2. \tag{23}$$

The next lemma, due to Gower et al. (2019), shows that this is indeed the case.

Lemma 5.4. *Let Assumption 5.1 and Assumption 5.3 hold and let f be convex and L-smooth, then expected smoothness (i.e., Assumption 5.2) holds with $A = L$ and $C = \sigma^2$.*

The main result of this section is formulated next.

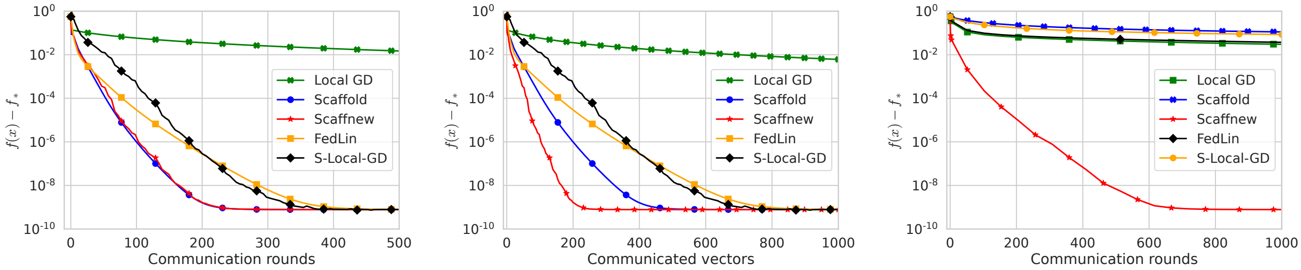
Theorem 5.5. *Let Assumptions 3.1, 3.2, 5.2 and 5.1 hold. Let $0 < \gamma \leq 1/A$ and $0 < p \leq 1$. Then, the iterates of SProxSkip (Algorithm 3) satisfy*

$$\mathbb{E}\left[\Psi_T\right] \leq (1 - \zeta)^T \Psi_0 + \frac{\gamma^2 C}{\zeta},$$

where $\zeta := \min\{\gamma\mu, p^2\}$.

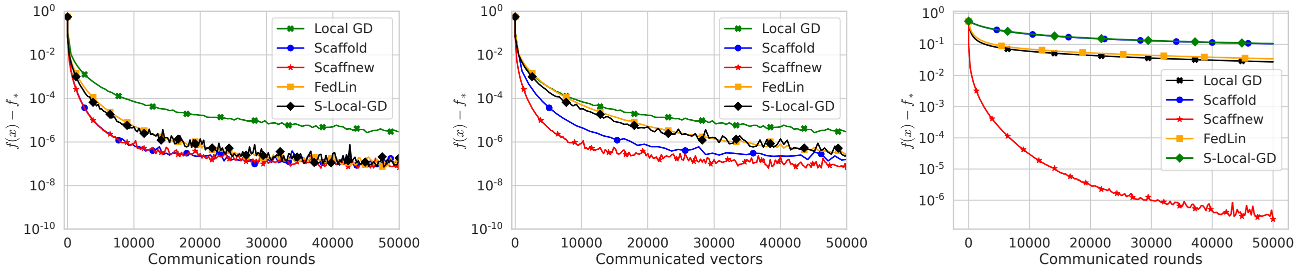
This result also gives us rates for Scaffnew (Algorithm 2).

Corollary 5.6. *Consider Scaffnew (Algorithm 2) or SProxSkip (Algorithm 3). Choose any $0 < \varepsilon < 1$. If we choose $\gamma = \min\left\{\frac{1}{A}, \frac{\varepsilon\mu}{2C}\right\}$ and $p = \sqrt{\gamma\mu}$, then in order to guarantee $\mathbb{E}\left[\Psi_T\right] \leq \varepsilon$, it suffices to take $T \geq \max\left\{\frac{A}{\mu}, \frac{2C}{\varepsilon\mu^2}\right\} \log\left(\frac{2\Psi_0}{\varepsilon}\right)$ iterations, which results in $\max\left\{\sqrt{\frac{A}{\mu}}, \sqrt{\frac{2C}{\varepsilon\mu^2}}\right\} \log\left(\frac{2\Psi_0}{\varepsilon}\right)$ communications (in case of Scaffnew) resp. prox evaluations (in case of SProxSkip) on average.*



(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.



(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.

Limitations. The main limitation of applying analysis of SProxSkip in the FL setting (Algorithm 2) is that we do not achieve linear speedup in terms of the number of clients. This issue comes from the analysis technique and it needs deeper investigation. The same problem appears in the analysis of FedLin, but it does not in the analysis of Scaffold.

5.2. Decentralized training

Let us now discuss the minimization problem with decentralized communication. Given a graph $G = (V, E)$ with nodes V and edges E, we assume that every communication node i receives a weighted average of its neighbors' vectors with weights $W_{i1}, \dots, W_{in} \in [0, 1]$. Besides, nodes i and j communicate if and only if $W_{ij} \neq 0$, which is also equivalent to $(i, j) \in E$. The weights W_{ij} define the *mixing matrix* $\mathbf{W}$ that we assume to be symmetric, doubly stochastic, and positive semi-definite. Then, the problem is equivalent to

$$\min_{x\in\mathbb{R}^d} f(x) \quad \text{subject to} \quad (\mathbf{I}-\mathbf{W})x = 0,$$

where $\mathbf{I}$ is the identity matrix. Let us set $\mathbf{L}$ to be the square-root of $\mathbf{I} - \mathbf{W}$ and define the indicator function $\psi(y)$ by setting $\psi(0) = 0$ and $\psi(y) = +\infty$ for any $y \neq 0$, which is similar to our previous definition in equation (7). Then, the constraint $(\mathbf{I} - \mathbf{W})x = 0$ is equivalent to $\mathbf{L}x = 0$, so the problem can be rewritten as

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

The reason we define $\mathbf{L}$ this way is that splitting algorithms require computation of $\mathbf{L}\mathbf{L}^\top u$ for some vector u, which in our case is exactly $(\mathbf{I} - \mathbf{W})u$. Algorithmically, computing this product corresponds to communicating over the graph. For convenience, we provide the algorithm formulation in the graph notation in the appendix; see Algorithm 5. The convergence of our decentralized algorithm is stated below.

Theorem 5.7. *Let f satisfy Assumption 4.1 and define the spectral gap of $\mathbf{W}$ as $\delta = 1 - \lambda_2(\mathbf{W}) \in (0, 1)$. If we set $p \in (0, 1]$, $\gamma \leq {}^1\!/\!_L$, $\tau \leq {}^p\!/\!_\gamma$, then the average iterate $\overline{x}_T$ satisfies*

$$\mathbb{E}\left[\|\overline{x}_T - x_\star\|^2\right] \leq (1 - \min(\gamma\mu, p\gamma\tau\delta))^T \Phi_0,$$

where $\Phi_0 \leq \|x_0 - x_\star\|^2 + \frac{\gamma}{p\tau\delta n}\sum_{i=1}^n \|\nabla f_i(x_*)\|^2$.

If we plug-in $\tau = {}^p\!/\!_\gamma$, the theorem implies that the new rate is $\tilde{\mathcal{O}}(\kappa + \frac{1}{p^2\delta})$. Thus, it is optimal to choose $p = \sqrt{{}^1\!/\!_{(\delta\kappa)}}$ whenever the network is sufficiently well-connected. If passing a message is challenging, which happens when

$\delta \leq 1/\kappa$, then it is optimal to communicate every iteration by setting $p = 1$.

6. Experiments

To test the performance of algorithms and illustrate theoretical results, we use classical logistic regression problem. The loss function for this model has the following form:

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

where $a_i \in \mathbb{R}^d$ and $b_i \in \{-1, +1\}$ are the data samples. We set the regularization parameter $\lambda = 10^{-4}L$, where L is the smoothness constant.

We implemented all algorithms in Python using the package RAY (Moritz et al., 2018) to utilize parallelization. All methods were evaluated on a workstation with an Intel(R) Xeon(R) Gold 6146 CPU at 3.20GHz with 24 cores. We use the 'w8a' dataset from LIBSVM library (Chang & Lin, 2011).

In our experiments, we have two settings: deterministic (Figure 1) and stochastic problems (Figure 2). First, we provide results with tuned hyper-parameters (subplot (a)). Local GD converges to the neighborhood of the solution due to data-heterogeneity. Scaffold and Scaffnew have the same convergence rate in terms of communication rounds and this rate is better than others. However, Scaffnew outperforms Scaffold in terms of communicated vectors since it does not transmit control variables (subplot (b)). Second, we test algorithms with theoretical hyper-parameters (subplot (c)). In this setting, Scaffnew outperforms other methods dramatically since our theory guarantees that we can use large stepsizes. The number of local steps is set to be $\sqrt{\hat{\kappa}}$, where $\hat{\kappa} = \frac{L}{\lambda}$ is an estimate of the condition number.

References

Arjevani, Y. and Shamir, O. Communication complexity of distributed convex learning and optimization. In *Advances in Neural Information Processing Systems*, 2015. (Cited on pages 4 and 7)

Bauschke, H. H., Moursi, W. M., and Wang, X. Generalized monotone operators and their averaged resolvents. *Mathematical Programming*, 189(1):55–74, 2021. (Cited on page 5)

Beck, A. *First order methods in optimization*. MOS-SIAM Series on Optimization, 2017. (Cited on page 1)

Chang, C.-C. and Lin, C.-J. LIBSVM: A library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011. (Cited on page 9)

Chen, P., Huang, J., and Zhang, X. A primal–dual fixed point algorithm for convex separable minimization with applications to image restoration. *Inverse Problems*, 29(2):025011, 2013. (Cited on page 17)

Combettes, P. L. and Pesquet, J.-C. Proximal splitting methods in signal processing. *arXiv preprint arXiv:0912.3522*, 2009. (Cited on page 1)

Combettes, P. L., Condat, L., Pesquet, J.-C., and Vũ, B. C. A forward-backward view of some primal-dual optimization methods in image recovery. In *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 4141–4145. IEEE, 2014. (Cited on page 17)

Condat, L., Kitahara, D., Contreras, A., and Hirabayashi, A. Proximal splitting algorithms for convex optimization: A tour of recent advances, with new twists. *arXiv preprint arXiv:1912.00137*, 2019. (Cited on page 17)

Condat, L., Malinovsky, G., and Richtárik, P. Distributed proximal splitting algorithms with rates and acceleration. *Frontiers in Signal Processing*, pp. 12, 2022. (Cited on page 17)

Dekel, O., Gilad-Bachrach, R., Shamir, O., and Xiao, L. Optimal distributed online prediction using mini-batches. *Journal of Machine Learning Research*, 13(1):165–202, January 2012. (Cited on page 7)

Drori, Y., Sabach, S., and Teboulle, M. A simple algorithm for a class of nonsmooth convex–concave saddle-point problems. *Operations Research Letters*, 43(2):209–214, 2015. (Cited on page 17)

Friedlander, M. and Goh, G. Efficient evaluation of scaled proximal operators. *Electronic Transactions on Numerical Analysis*, 46:1–23, 03 2016. (Cited on page 2)

Gorbunov, E., Hanzely, F., and Richtárik, P. Local SGD: Unified theory and new efficient methods. In *International Conference on Artificial Intelligence and Statistics*, pp. 3556–3564. PMLR, 2021. (Cited on pages 2, 3, and 7)

Gower, R. M., Loizou, N., Qian, X., Sailanbayev, A., Shulgin, E., and Richtárik, P. SGD: General analysis and improved rates. In *International Conference on Machine Learning*, pp. 5200–5209. PMLR, 2019. (Cited on pages 4 and 7)

Gower, R. M., Richtárik, P., and Bach, F. Stochastic quasi-gradient methods: Variance reduction via Jacobian sketching. *Mathematical Programming*, 188(1):135–192, 2021. (Cited on pages 4 and 7)

Hanzely, F. and Richtárik, P. Federated learning of a mixture of global and local models. *arXiv:2002.05516*, 2020. (Cited on page 7)

Hanzely, F., Hanzely, S., Horváth, S., and Richtárik, P. Lower bounds and optimal algorithms for personalized federated learning. In *NeurIPS*, 2020. (Cited on page 7)

Kairouz, P., McMahan, H. B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A. N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., D'Oliveira, R. G. L., Rouayheb, S. E., Evans, D., Gardner, J., Garrett, Z., Gascón, A., Ghazi, B., Gibbons, P. B., Gruteser, M., Harchaoui, Z., He, C., He, L., Huo, Z., Hutchinson, B., Hsu, J., Jaggi, M., Javidi, T., Joshi, G., Khodak, M., Konečný, J., Korolova, A., Koushanfar, F., Koyejo, S., Lepoint, T., Liu, Y., Mittal, P., Mohri, M., Nock, R., Özgür, A., Pagh, R., Raykova, M., Qi, H., Ramage, D., Raskar, R., Song, D., Song, W., Stich, S. U., Sun, Z., Suresh, A. T., Tramèr, F., Vepakomma, P., Wang, J., Xiong, L., Xu, Z., Yang, Q., Yu, F. X., Yu, H., and

Zhao, S. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021. (Cited on pages 2, 3, and 6)

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S. U., and Suresh, A. T. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pp. 5132–5143. PMLR, 2020. (Cited on pages 2, 3, 4, and 7)

Karimireddy, S. P., Jaggi, M., Kale, S., Mohri, M., Reddi, S. J., Stich, S. U., and Suresh, A. T. Breaking the centralized barrier for cross-device federated learning. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*. Curran Associates, Inc., 2021. (Cited on page 7)

Khaled, A., Mishchenko, K., and Richtárik, P. First analysis of local GD on heterogeneous data. In *NeurIPS Workshop on Federated Learning for Data Privacy and Confidentiality*, pp. 1–11, 2019. (Cited on pages 2 and 3)

Khaled, A., Mishchenko, K., and Richtárik, P. Tighter theory for local SGD on identical and heterogeneous data. In *The 23rd International Conference on Artificial Intelligence and Statistics (AISTATS 2020)*, 2020. (Cited on pages 2, 3, and 6)

Koloskova, A., Loizou, N., Boreiri, S., Jaggi, M., and Stich, S. U. A unified theory of decentralized SGD with changing topology and local updates. In *37th International Conference on Machine Learning (ICML)*. PMLR, 2020. (Cited on pages 2 and 6)

Koloskova, A., Lin, T., and Stich, S. U. An improved analysis of gradient tracking for decentralized machine learning. In *Advances in Neural Information Processing Systems 34 (NeurIPS)*, volume 34. Curran Associates, Inc., 2021. (Cited on page 4)

Konečný, J., McMahan, H. B., Yu, F., Richtárik, P., Suresh, A. T., and Bacon, D. Federated learning: strategies for improving communication efficiency. In *NIPS Private Multi-Party Machine Learning Workshop*, 2016. (Cited on page 2)

Kovalev, D., Salim, A., and Richtárik, P. Optimal and practical algorithms for smooth and strongly convex decentralized optimization. *Neural Information Processing Systems (NeurIPS)*, 2020. (Cited on page 4)

Kovalev, D., Gasanov, E., Richtárik, P., and Gasnikov, A. Lower bounds and optimal algorithms for smooth and strongly convex decentralized optimization over time-varying networks. In *Advances in Neural Information Processing Systems 34*, 2021a. (Cited on page 5)

Kovalev, D., Shulgin, E., Richtárik, P., Rogozin, A., and Gasnikov, A. ADOM: Accelerated decentralized optimization method for time-varying networks. In *International Conference on Machine Learning*, pp. 5784–5793. PMLR, 2021b. (Cited on page 5)

Lan, G. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133:365–397, 2012. (Cited on page 3)

Lin, T., Stich, S. U., and Jaggi, M. Don't use large mini-batches, use local SGD. In *International Conference on Learning Representations (ICLR)*, 2018. (Cited on page 2)

Lin, T., Karimireddy, S. P., Stich, S., and Jaggi, M. Quasi-global momentum: Accelerating decentralized deep learning on heterogeneous data. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139, pp. 6654–6665. PMLR, 2021. (Cited on page 7)

Lorenzo, P. D. and Scutari, G. NEXT: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, 2016. (Cited on page 4)

Loris, I. and Verhoeven, C. On a generalization of the iterative soft-thresholding algorithm for the case of non-separable penalty. *Inverse Problems*, 27(12), 2011. (Cited on page 17)

Luke, D. R. *Proximal Methods for Image Processing*, pp. 165–202. Springer International Publishing, Cham, 2020. ISBN 978-3-030-34413-9. (Cited on page 1)

Malinovskiy, G., Kovalev, D., Gasanov, E., Condat, L., and Richtarik, P. From local SGD to local fixed-point methods for federated learning. In *International Conference on Machine Learning*, pp. 6692–6701. PMLR, 2020. (Cited on page 6)

Mangasarian, O. L. and Solodov, M. V. Backpropagation convergence via deterministic nonmonotone perturbed minimization. In Cowan, J., Tesauro, G., and Alspector, J. (eds.), *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann, 1994. (Cited on page 2)

McDonald, R., Hall, K., and Mann, G. Distributed training strategies for the structured perceptron. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pp. 456–464. Association for Computational Linguistics, 2010. (Cited on page 2)

McMahan, H. B., Moore, E., Ramage, D., and y Arcas, B. A. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2016. (Cited on pages 2 and 6)

Mitra, A., Jaafar, R., Pappas, G., and Hassani, H. Linear convergence in federated learning: Tackling client heterogeneity and sparse gradients. *Advances in Neural Information Processing Systems*, 34, 2021. (Cited on pages 2, 3, and 7)

Moritz, P., Nishihara, R., Wang, S., Tumanov, A., Liaw, R., Liang, E., Elibol, M., Yang, Z., Paul, W., Jordan, M. I., and Stoica, I. Ray: A distributed framework for emerging AI applications. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*, pp. 561–577, 2018. (Cited on page 9)

Nedić, A., Olshevsky, A., and Shi, W. Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27, 07 2016. (Cited on page 4)

Nesterov, Y. *Introductory lectures on convex optimization: a basic course (Applied Optimization)*. Kluwer Academic Publishers, 2004. (Cited on page 3)

Nesterov, Y. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013. (Cited on page 1)

Parikh, N. and Boyd, S. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, jan 2014. (Cited on pages 1 and 2)

Salim, A., Condat, L., Mishchenko, K., and Richtárik, P. Dualize, split, randomize: Fast nonsmooth optimization algorithms. *arXiv preprint arXiv:2004.02635*, 2020. (Cited on page 17)

Shalev-Shwartz, S. and Ben-David, S. *Understanding machine learning: from theory to algorithms*. Cambridge University Press, 2014. (Cited on pages 1 and 2)

Stich, S. U. Local SGD converges fast and communicates little. In *International Conference on Learning Representations (ICLR)*, 2019. (Cited on page 2)

Tang, H., Lian, X., Yan, M., Zhang, C., and Liu, J. D^2: Decentralized training over decentralized data. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80, pp. 4848–4856. PMLR, 2018. (Cited on page 4)

Vogels, T., He, L., Koloskova, A., Lin, T., Karimireddy, S. P., Stich, S. U., and Jaggi, M. Relaysum for decentralized deep learning on heterogeneous data. In *Advances in Neural Information Processing Systems (NeurIPS)*. Curran Associates, Inc., 2021. (Cited on page 4)

Woodworth, B., Patel, K. K., Stich, S. U., Dai, Z., Bullins, B., McMahan, H. B., Shamir, O., and Srebro, N. Is local SGD better than minibatch SGD? In *37th International Conference on Machine Learning (ICML)*. PMLR, 2020a. (Cited on page 7)

Woodworth, B. E., Patel, K. K., and Srebro, N. Minibatch vs local sgd for heterogeneous distributed learning. In *NeurIPS*, 2020b. (Cited on page 3)

Woodworth, B. E., Bullins, B., Shamir, O., and Srebro, N. The min-max complexity of distributed stochastic convex optimization with intermittent communication. In *Proceedings of Thirty Fourth Conference on Learning Theory*, volume 134 of *Proceedings of Machine Learning Research*, pp. 4386–4437. PMLR, 15–19 Aug 2021. (Cited on page 3)

Yuan, H. and Ma, T. Federated accelerated stochastic gradient descent. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5332–5344. Curran Associates, Inc., 2020. (Cited on page 7)

Yuan, K. and Alghunaim, S. A. Removing data heterogeneity influence enhances network topology dependence of decentralized SGD. *arXiv preprint arXiv:2105.08023*, 2021. (Cited on page 4)

Zhang, J., De Sa, C., Mitliagkas, I., and Ré, C. Parallel SGD: When does averaging help? *arXiv preprint arXiv:1606.07365*, 2016. (Cited on page 2)

Zhou, H. and Hastie, T. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B*, 67: 301–320, 2005. (Cited on page 1)

Appendix

A. Basic Facts

The Bregman divergence of a differentiable function $f\colon \mathbb{R}^d \to \mathbb{R}$ is defined by

$$D_f(x,y) \coloneqq f(x) - f(y) - \langle \nabla f(y), x-y\rangle.$$

It is easy to see that

$$\langle \nabla f(x) - \nabla f(y), x-y\rangle = D_f(x,y) + D_f(y,x), \quad \forall x,y \in \mathbb{R}^d \tag{24}$$

For an L-smooth and μ-strongly convex function $f\colon \mathbb{R}^d \to \mathbb{R}$, we have

$$\frac{\mu}{2}\|x-y\|^2 \le D_f(x,y) \le \frac{L}{2}\|x-y\|^2, \quad \forall x,y \in \mathbb{R}^d \tag{25}$$

and

$$\frac{1}{2L}\|\nabla f(x) - \nabla f(y)\|^2 \le D_f(x,y) \le \frac{1}{2\mu}\|\nabla f(x) - \nabla f(y)\|^2, \quad \forall x,y \in \mathbb{R}^d. \tag{26}$$

Given $\psi\colon \mathbb{R}^d \to \mathbb{R}$, we define $\psi^*(y) \coloneqq \sup_{x\in\mathbb{R}^d}\{\langle x,y\rangle - \psi(x)\}$ to be its Fenchel conjugate. The proximity operator of ψ^* satisfies for any $\tau > 0$

$$\text{if} \quad u = \operatorname{prox}_{\tau\psi^*}(y), \quad \text{then} \quad u \in y - \tau\partial\psi^*(u). \tag{27}$$

B. Analysis of ProxSkip (Algorithm 1)

B.1. Proof of Lemma 3.4

Proof. In order to simplify notation, let $P(\cdot) \coloneqq \operatorname{prox}_{\frac{\gamma}{p}\psi}(\cdot)$, and

$$x \coloneqq \hat{x}_{t+1} - \frac{\gamma}{p}h_t, \qquad y \coloneqq x_\star - \frac{\gamma}{p}h_\star. \tag{28}$$

STEP 1 (Optimality conditions). Using the first-order optimality conditions for $f+\psi$ and using $h_\star \coloneqq \nabla f(x_\star)$, we obtain the following fixed-point identity for $x_\star$:

$$x_\star = \operatorname{prox}_{\frac{\gamma}{p}\psi}\left(x_\star - \frac{\gamma}{p}h_\star\right) \overset{(28)}{=} P(y). \tag{29}$$

STEP 2 (Recalling the steps of the method). Recall that the vectors x_t and h_t are in Algorithm 1 updated as follows:

$$x_{t+1} = \begin{cases} P(x) & \text{with probability} \quad p \\ \hat{x}_{t+1} & \text{with probability} \quad 1-p \end{cases}, \tag{30}$$

and

$$h_{t+1} = h_t + \frac{p}{\gamma}(x_{t+1} - \hat{x}_{t+1}) = \begin{cases} h_t + \frac{p}{\gamma}(P(x) - \hat{x}_{t+1}) & \text{with probability} \quad p \\ h_t & \text{with probability} \quad 1-p \end{cases}. \tag{31}$$

STEP 3 (One-step expectation of the Lyapunov function).

The expected value of the Lyapunov function

$$\Psi_t \coloneqq \|x_t - x_\star\|^2 + \frac{\gamma^2}{p^2}\|h_t - h_\star\|^2 \tag{32}$$

at time $t+1$, with respect to the coin toss at iteration t, is

$$\begin{aligned}
\mathbb{E}\left[\Psi_{t+1}\right] &\overset{(30)+(31)+(32)}{=} p\left(\|P(x)-x_\star\|^2+\frac{\gamma^2}{p^2}\left\|h_t+\frac{p}{\gamma}(P(x)-\hat{x}_{t+1})-h_\star\right\|^2\right)+(1-p)\left(\|\hat{x}_{t+1}-x_\star\|^2+\frac{\gamma^2}{p^2}\|h_t-h_\star\|^2\right)\\
&\overset{(29)}{=} p\left(\|P(x)-P(y)\|^2+\left\|\frac{\gamma}{p}h_t+P(x)-\hat{x}_{t+1}-\frac{\gamma}{p}h_\star\right\|^2\right)+(1-p)\left(\|\hat{x}_{t+1}-x_\star\|^2+\frac{\gamma^2}{p^2}\|h_t-h_\star\|^2\right)\\
&\overset{(28)+(29)}{=} p\left(\|P(x)-P(y)\|^2+\underbrace{\|P(x)-x+y-P(y)\|^2}_{\|Q(x)-Q(y)\|^2}\right)+(1-p)\left(\|\hat{x}_{t+1}-x_\star\|^2+\frac{\gamma^2}{p^2}\|h_t-h_\star\|^2\right).
\end{aligned}$$

STEP 4 (Applying firm nonexpansiveness). Applying firm nonexpansiveness of P (Lemma 3.3), this leads to the inequality

$$\begin{aligned}
\mathbb{E}\left[\Psi_{t+1}\right] &\overset{(10)}{\leq} p\|x-y\|^2+(1-p)\left(\|\hat{x}_{t+1}-x_\star\|^2+\frac{\gamma^2}{p^2}\|h_t-h_\star\|^2\right)\\
&\overset{(28)}{=} p\left\|\hat{x}_{t+1}-\frac{\gamma}{p}h_t-\left(x_\star-\frac{\gamma}{p}h_\star\right)\right\|^2+(1-p)\left(\|\hat{x}_{t+1}-x_\star\|^2+\frac{\gamma^2}{p^2}\|h_t-h_\star\|^2\right).
\end{aligned}$$

STEP 5 (Simple algebra). Next, we expand the squared norm and collect the terms, obtaining

$$\begin{aligned}
\mathbb{E}\left[\Psi_{t+1}\right] &\leq p\|\hat{x}_{t+1}-x_\star\|^2+p\frac{\gamma^2}{p^2}\|h_t-h_\star\|^2-2\gamma\langle\hat{x}_{t+1}-x_\star,h_t-h_\star\rangle+(1-p)\left(\|\hat{x}_{t+1}-x_\star\|^2+\frac{\gamma^2}{p^2}\|h_t-h_\star\|^2\right)\\
&= \|\hat{x}_{t+1}-x_\star\|^2-2\gamma\langle\hat{x}_{t+1}-x_\star,h_t-h_\star\rangle+\frac{\gamma^2}{p^2}\|h_t-h_\star\|^2. \qquad (33)
\end{aligned}$$

Finally, note that by our definition of w_t, we have the identity $\hat{x}_{t+1}=w_t+\gamma h_t$. Therefore, the first two terms above can be rewritten as

$$\begin{aligned}
\|\hat{x}_{t+1}-x_\star\|^2-2\gamma\langle\hat{x}_{t+1}-x_\star,h_t-h_\star\rangle &= \|w_t-w_\star+\gamma(h_t-h_\star)\|^2-2\gamma\langle w_t-w_\star+\gamma(h_t-h_\star),h_t-h_\star\rangle\\
&= \|w_t-w_\star\|^2+2\gamma\langle w_t-w_\star,h_t-h_\star\rangle+\gamma^2\|h_t-h_\star\|^2\\
&\quad -2\gamma\langle w_t-w_\star,h_t-h_\star\rangle-2\gamma^2\|h_t-h_\star\|^2\\
&= \|w_t-w_\star\|^2-\gamma^2\|h_t-h_\star\|^2. \qquad (34)
\end{aligned}$$

It remains to plug (34) into (33). □

B.2. Proof of Lemma 3.5

Proof. Recall the definition of w_t and $w_\star$ in (12). Plugging these expressions into $\|w_t-w_\star\|^2$, expanding the square, and applying properties of f as a μ-strong convex and L-smooth function, we get

$$\begin{aligned}
\|w_t-w_\star\|^2 &\overset{(12)}{=} \|x_t-x_\star-\gamma(\nabla f(x_t)-\nabla f(x_\star))\|^2\\
&= \|x_t-x_\star\|^2+\gamma^2\|\nabla f(x_t)-\nabla f(x_\star)\|^2-2\gamma\langle\nabla f(x_t)-\nabla f(x_\star),x_t-x_\star\rangle\\
&\overset{(25)}{\leq} (1-\gamma\mu)\|x_t-x_\star\|^2-2\gamma D_f(x_t,x_\star)+\gamma^2\|\nabla f(x_t)-\nabla f(x_\star)\|^2\\
&= (1-\gamma\mu)\|x_t-x_\star\|^2-2\gamma\left(D_f(x_t,x_\star)-\frac{\gamma}{2}\|\nabla f(x_t)-\nabla f(x_\star)\|^2\right)\\
&\overset{(26)}{\leq} (1-\gamma\mu)\|x_t-x_\star\|^2,
\end{aligned}$$

where the last inequality holds if $0\leq\gamma\leq\frac{1}{L}$. □

C. Analysis of SProxSkip (Algorithm 3)

C.1. The algorithm

We consider a variant of ProxSkip which uses a *stochastic* gradient $g_t(x_t)$ instead of $\nabla f(x_t)$; see Algorithm 3.

Algorithm 3 SProxSkip (Stochastic gradient version of 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
    $\hat{x}_{t+1} = x_t - \gamma(g_t(x_t) - h_t)$    ◇ Take a stochastic 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
```

C.2. Two lemmas

Lemma C.1 is an extension of Lemma 3.4 to the stochastic case. In this result, we work with

$$w_t' = x_t - \gamma g_t(x_t) \tag{35}$$

instead of $w_t = x_t - \gamma \nabla f(x_t)$.

Lemma C.1. *If Assumptions 3.1 and 3.2 hold, $\gamma > 0$ and $0 < p \leq 1$, then*

$$\mathbb{E}\left[\Psi_{t+1}\right] \leq \|w_t' - w_\star\|^2 + (1 - p^2)\frac{\gamma^2}{p^2}\|h_t - h_\star\|^2, \tag{36}$$

where the expectation is taken over the θ_t in Algorithm 3.

Proof. The proof is identical to the proof of Lemma 3.4. □

Likewise, Lemma C.2 is an extension of Lemma 3.5 to the stochastic case.

Lemma C.2. *Let Assumption 3.1 hold with any $\mu \geq 0$. If $0 < \gamma \leq \frac{1}{A}$, then*

$$\mathbb{E}\left[\|w_t' - w_\star\|^2\right] \leq (1 - \gamma\mu)\|x_t - x_\star\|^2 + \gamma^2 C, \tag{37}$$

where the expectation is w.r.t. the randomness in the stochastic gradient $g_t(\cdot)$.

Proof. Recall the definition of w_t' in (35) and $w_\star$ in (12). Plugging these expressions into $\|w_t' - w_\star\|^2$, expanding the square, we get

$$\begin{aligned} \|w_t' - w_\star\|^2 &\overset{(12)+(35)}{=} \|x_t - x_\star - \gamma(g_t(x_t) - \nabla f(x_\star))\|^2 \\ &= \|x_t - x_\star\|^2 + \gamma^2\|g_t(x_t) - \nabla f(x_\star)\|^2 - 2\gamma\langle g_t(x_t) - \nabla f(x_\star), x_t - x_\star\rangle. \end{aligned} \tag{38}$$

Taking expectation w.r.t. the randomness of the stochastic gradient $g_t(x_t)$, and using unbiasedness (Assumption 5.2) and expected smoothness (Assumption 5.1), we get

$$\begin{aligned} \mathbb{E}\left[\|w_t' - w_\star\|^2\right] &\overset{(38)}{=} \|x_t - x_\star\|^2 + \gamma^2\mathbb{E}\left[\|g_t(x_t) - \nabla f(x_\star)\|^2\right] - 2\gamma\langle \mathbb{E}\left[g_t(x_t)\right] - \nabla f(x_\star), x_t - x_\star\rangle \\ &= \|x_t - x_\star\|^2 - 2\gamma\langle \nabla f(x_t) - \nabla f(x_\star), x_t - x_\star\rangle + \gamma^2\mathbb{E}\left[\|g_t(x_t) - \nabla f(x_\star)\|^2\right]. \end{aligned}$$

The second term can be decomposed using the identity $\langle \nabla f(x_t) - \nabla f(x_\star), x_t - x_\star \rangle = D_f(x_t, x_\star) + D_f(x_\star, x_t)$ (see (24)), and the third term can be bounded via $\mathbb{E}\left[\|g_t(x_t) - \nabla f(x_\star)\|^2\right] \leq 2AD_f(x_t, x_\star) + C$ (see expected smoothness; Assumption 5.2), which leads to

$$\mathbb{E}\left[\|w_t' - w_\star\|^2\right] \leq \|x_t - x_\star\|^2 - 2\gamma(D_f(x_t, x_\star) + D_f(x_\star, x_t)) + \gamma^2\left(2AD_f(x_t, x_\star) + C\right). \tag{39}$$

By plugging the inequality $\mu\|x_t - x_\star\|^2 \leq 2D_f(x_\star, x_t)$ (see (25)) into (39), we get

$$\begin{aligned} \mathbb{E}\left[\|w_t' - w_\star\|^2\right] &\leq (1-\gamma\mu)\|x_t - x_\star\|^2 - 2\gamma D_f(x_t, x_\star) + \gamma^2\left(2AD_f(x_t, x_\star) + C\right) \\ &\leq (1-\gamma\mu)\|x_t - x_\star\|^2 - 2\gamma(1-\gamma A)D_f(x_t, x_\star) + \gamma^2 C. \end{aligned}$$

Finally, the stepsize restriction $\gamma \leq \frac{1}{A}$ allows us to produce the estimate

$$\mathbb{E}\left[\|w_t' - w_\star\|^2\right] \leq (1-\gamma\mu)\|x_t - x_\star\|^2 + \gamma^2 C, \tag{40}$$

which is what we wanted to show. □

C.3. Proof of Theorem 5.5

Proof. Combining Lemma C.1 and Lemma C.2, we get

$$\begin{aligned} \mathbb{E}\left[\Psi_{t+1}\right] &\overset{(36)+(40)}{\leq} (1-\gamma\mu)\|x_t - x_\star\|^2 + (1-p^2)\frac{\gamma^2}{p^2}\|h_t - h_\star\|^2 + \gamma^2 C \\ &\leq \max\{1-\gamma\mu, 1-p^2\}\Psi_t + \gamma^2 C \\ &= (1-\zeta)\Psi_t + \gamma^2 C\,, \end{aligned}$$

where $\zeta \coloneqq \min\{\gamma\mu, p^2\}$. Taking full expectation, we get $\mathbb{E}\left[\Psi_{t+1}\right] \leq (1-\zeta)\mathbb{E}\left[\Psi_t\right] + \gamma^2 C$, and unrolling the recurrence, we finally obtain

$$\mathbb{E}\left[\Psi_T\right] \leq (1-\zeta)^T \Psi_0 + \frac{\gamma^2 C}{\zeta}\,. \tag{41}$$

□

C.4. Proof of Corollary 5.6

Recall that Theorem 5.5 requires the stepsize γ to satisfy

$$0 < \gamma \leq \frac{1}{A}. \tag{42}$$

Pick $0 < \varepsilon < 1$. We will now choose γ and T such that $\mathbb{E}\left[\Psi_T\right] \leq \varepsilon$. We shall do so by bounding both terms on the right-hand side of (41) by $\frac{\varepsilon}{2}$.

- In order to minimize the number of prox evaluations, whatever the choice of γ will be, we choose the smallest probability p which does not lead to any degradation of the rate $\zeta \coloneqq \min\{\gamma\mu, p^2\}$. That is, we choose

  $$p = \sqrt{\gamma\mu}, \tag{43}$$

  in which case $\zeta = \gamma\mu$.

- The first term on the right-hand side of (41) can be bounded as follows:

  $$T \geq \frac{1}{\gamma\mu}\log\left(\frac{2\Psi_0}{\varepsilon}\right) \quad \Longrightarrow \quad (1-\zeta)^T\Psi_0 \leq \frac{\varepsilon}{2}. \tag{44}$$

- The second term on the right-hand side of (41) can be bounded as follows:

  $$\gamma \leq \frac{\varepsilon\mu}{2C} \quad \Longrightarrow \quad \frac{\gamma^2 C}{\zeta} = \frac{\gamma C}{\mu} \leq \frac{\varepsilon}{2}. \tag{45}$$

Since the number of iterations (44) depends inversely on the stepsize γ, we choose the largest stepsize consistent with the bounds (42) and (45):

$$\gamma = \min\left\{\frac{1}{A}, \frac{\varepsilon\mu}{2C}\right\}. \tag{46}$$

By plugging this into (44), we get the iteration complexity bound

$$T \geq \max\left\{\frac{A}{\mu}, \frac{2C}{\varepsilon\mu^2}\right\}\log\left(\frac{2\Psi_0}{\varepsilon}\right) \quad \Longrightarrow \quad \mathbb{E}\left[\Psi_T\right] \leq \varepsilon\,.$$

Since in each iteration we evaluate the prox with probability p given by (43), and since there are T iterations, the expected number of prox evaluations is given by

$$pT \overset{(44)}{\geq} p\frac{1}{\gamma\mu}\log\left(\frac{2\Psi_0}{\varepsilon}\right) \overset{(43)}{=} \sqrt{\frac{1}{\gamma\mu}}\log\left(\frac{2\Psi_0}{\varepsilon}\right) \overset{(46)}{=} \max\left\{\sqrt{\frac{A}{\mu}}, \sqrt{\frac{2C}{\varepsilon\mu^2}}\right\}\log\left(\frac{2\Psi_0}{\varepsilon}\right).$$

D. Decentralized Analysis

Let us now analyze the convergence of Algorithm 4 and Algorithm 5.

Algorithm 4 SplitSkip

```
stepsizes $\gamma > 0$ and $\tau > 0$, matrix $\mathbf{L} \in \mathbb{R}^{d\times m}$, probability $p > 0$, initial iterate $x_0 \in \mathbb{R}^d$, initial control variate $y_0 = 0 \in \mathbb{R}^m$, number of iterations $T \geq 1$
for $t = 0, 1, \ldots, T-1$ do
    $\hat{x}_{t+1} = x_t - \gamma(\nabla f(x_t) + \mathbf{L}^\top y_t)$        ◇ Take a gradient-type step adjusted via the control variate $y_t$
    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
        $y_{t+1} = \mathrm{prox}_{\tau\psi^*}\left(y_t + \tau\mathbf{L}\hat{x}_{t+1}\right)$        ◇ Apply prox, but only very rarely! (with small probability $p$)
        $x_{t+1} = \hat{x}_{t+1} - \frac{\gamma}{p}\mathbf{L}^\top(y_{t+1} - y_t)$
    else
        $x_{t+1} = \hat{x}_{t+1}$
        $y_{t+1} = y_t$        ◇ Skip the prox!
    end if
end for
```

Algorithm 5 Decentralized Scaffnew

```
stepsizes $\gamma > 0$ and $\tau > 0$, initial iterates $x_{1,0} = \ldots = x_{n,0} = x_0 \in \mathbb{R}^d$, initial control variables $h_{1,0} = \ldots = h_{n,0} = 0 \in \mathbb{R}^d$, weights for averaging $\mathbf{W} = (W_{ij})_{i,j=1}^n$
for $t = 0, 1, \ldots, T-1$ do
    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
    for $i = 1, \ldots, n$ do
        $\hat{x}_{i,t+1} = x_{i,t} - \gamma(\nabla f_i(x_{i,t}) - h_{i,t})$        ◇ Take a gradient-type step adjusted via the control variate $h_{i,t}$
        if $\theta_t = 1$ then
            $x_{i,t+1} = \left(1 - \frac{\gamma\tau}{p}\right)\hat{x}_{i,t+1} + \frac{\gamma\tau}{p}\sum_{j=1}^n W_{ij}\hat{x}_{j,t+1}$        ◇ Communicate, but only very rarely! (with small prob. $p$)
            $h_{i,t+1} = h_{i,t} + \frac{p}{\gamma}(x_{i,t+1} - \hat{x}_{i,t+1})$        ◇ Update the control variate $h_{i,t}$
        else
            $x_{i,t+1} = \hat{x}_{i,t+1}$        ◇ Skip communication!
            $h_{i,t+1} = h_{i,t}$
        end if
    end for
end for
```

Notice Algorithm 5 is a special case of Algorithm 4 with $\mathbf{L} = (\mathbf{I} - \mathbf{W})^{1/2}$ and ψ being indicator function of 0. Indeed, the conjugate of ψ is simply 0 everywhere, so $\mathrm{prox}_{\tau\psi^*}(y) = y$ for any y. Thus, if we define $h_t := -\mathbf{L}^\top y_t$, we obtain Algorithm 5, which we explain in more detail in Section D.1. For this reason, we will do the analysis for the more general Algorithm 4.

Let us also add a few connections of this method to the existing literature on primal–dual algorithms. When $p = 1$, Algorithm 4 reverts to a primal–dual algorithm first proposed by Loris and Verhoven for least squares problems (Loris & Verhoeven, 2011), and rediscovered later under the names PDFP2O (Chen et al., 2013) and PAPC (Drori et al., 2015). The convergence of this algorithm has been analyzed by Combettes et al. (2014); Condat et al. (2019; 2022) and generalized to the case of stochastic gradients by Salim et al. (2020), who also studied its linear convergence under similar assumptions, albeit without skipping the prox step.

Our analysis is based on the following Lyapunov function:

$$\Phi_t := \|x_t - x_\star\|^2 + \frac{\gamma}{p\tau}\|y_t - y_\star\|^2, \tag{47}$$

where y_t is the dual variable from Algorithm 4.

Theorem D.1. *Let Assumption 3.1 and Assumption 3.2 hold, and assume that for any y, we have $\partial\psi^*(y) \subseteq \mathrm{range}(\mathbf{L})$. If we choose $p \in (0,1]$, $\gamma \leq \frac{1}{L}$, $\gamma\tau \leq \frac{p}{\|\mathbf{L}\mathbf{L}^\top\|}$, then*

$$\mathbb{E}\left[\|x_T - x_\star\|^2\right] \leq (1-\zeta)^T \Phi_0,$$

where $\zeta = \min\{\gamma\mu, p\gamma\tau\lambda_{\min}^+(\mathbf{L}\mathbf{L}^\top)\}$.

Proof. Let us define $\hat{y}_{t+1} \coloneqq \mathrm{prox}_{\tau\psi^*}(y_t + \tau\mathbf{L}\hat{x}_{t+1})$. As stated in equation (27), this definition implies the following implicit representation of $\hat{y}_{t+1}$:

$$\hat{y}_{t+1} = y_t + \tau\mathbf{L}\hat{x}_{t+1} - \tau(\psi^*)'(y_{t+1}),$$

where $(\psi^*)'(y_{t+1}) \in \partial\psi^*(y_{t+1})$ is a subgradient of ψ^* at y_{t+1}. With the help of $\hat{y}_{t+1}$, we can expand the expected distance to the solution,

$$\begin{aligned}
\mathbb{E}\left[\|x_{t+1} - x_\star\|^2\right] &= p\|\hat{x}_{t+1} - x_\star - \frac{\gamma}{p}\mathbf{L}^\top(\hat{y}_{t+1} - y_t)\|^2 + (1-p)\|\hat{x}_{t+1} - x_\star\|^2 \\
&= p\left[\|\hat{x}_{t+1} - x_\star\|^2 - 2\frac{\gamma}{p}\langle\hat{x}_{t+1} - x_\star, \mathbf{L}^\top(\hat{y}_{t+1} - y_t)\rangle + \frac{\gamma^2}{p^2}\|\mathbf{L}^\top(\hat{y}_{t+1} - y_t)\|^2\right] + (1-p)\|\hat{x}_{t+1} - x_\star\|^2 \\
&= \|\hat{x}_{t+1} - x_\star\|^2 - 2\gamma\langle\hat{x}_{t+1} - x_\star, \mathbf{L}^\top(\hat{y}_{t+1} - y_t)\rangle + \frac{\gamma^2}{p}\|\mathbf{L}^\top(\hat{y}_{t+1} - y_t)\|^2.
\end{aligned}$$

Next, let us recur the first term to $\|w_t - w_\star\|$ by using the expansion $\|a+b\|^2 = \|a\|^2 + 2\langle a+b, b\rangle - \|b\|^2$,

$$\begin{aligned}
\|\hat{x}_{t+1} - x_\star\|^2 &= \|w_t - w_\star - \gamma\mathbf{L}^\top(y_t - y_\star)\|^2 \\
&= \|w_t - w_\star\|^2 - 2\gamma\langle\hat{x}_{t+1} - x_\star, \mathbf{L}^\top(y_t - y_\star)\rangle - \gamma^2\|\mathbf{L}^\top(y_t - y_\star)\|^2.
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathbb{E}\left[\|x_{t+1} - x_\star\|^2\right] &= \|w_t - w_\star\|^2 - 2\gamma\langle\hat{x}_{t+1} - x_\star, \mathbf{L}^\top(y_t - y_\star)\rangle - 2\gamma\langle\hat{x}_{t+1} - x_\star, \mathbf{L}^\top(\hat{y}_{t+1} - y_t)\rangle \\
&\quad - \gamma^2\|\mathbf{L}^\top(y_t - y_\star)\|^2 + \frac{\gamma^2}{p}\|\mathbf{L}^\top(\hat{y}_{t+1} - y_t)\|^2 \\
&= \|w_t - w_\star\|^2 - 2\gamma\langle\hat{x}_{t+1} - x_\star, \mathbf{L}^\top(\hat{y}_{t+1} - y_\star)\rangle - \gamma^2\|\mathbf{L}^\top(y_t - y_\star)\|^2 + \frac{\gamma^2}{p}\|\mathbf{L}^\top(\hat{y}_{t+1} - y_t)\|^2.
\end{aligned}$$

Now, we can turn our attention to the convergence of the dual variable y_t. Since y_{t+1} is updated with probability p, it holds

$$\begin{aligned}
\mathbb{E}\left[\|y_{t+1} - y_\star\|^2\right] &= p\|y_t - y_\star + (\hat{y}_{t+1} - y_t)\|^2 + (1-p)\|y_t - y_\star\|^2 \\
&= p\|y_t - y_\star\|^2 + 2p\langle\hat{y}_{t+1} - y_\star, \hat{y}_{t+1} - y_t\rangle - p\|\hat{y}_{t+1} - y_t\|^2 + (1-p)\|y_t - y_\star\|^2 \\
&= \|y_t - y_\star\|^2 + 2p\langle\hat{y}_{t+1} - y_\star, \hat{y}_{t+1} - y_t\rangle - p\|\hat{y}_{t+1} - y_t\|^2 \\
&= \|y_t - y_\star\|^2 + 2p\tau\langle\hat{y}_{t+1} - y_\star, \mathbf{L}\hat{x}_{t+1} - (\psi^*)'(\hat{y}_{t+1})\rangle - p\|\hat{y}_{t+1} - y_t\|^2.
\end{aligned}$$

By the first-order optimality conditions, we have $\mathbf{L}x_\star = (\psi^*)'(x_\star)$, where $(\psi^*)'(x_\star)$ is some subgradient of ψ^* at $x_\star$. Therefore, convexity of ψ^* gives

$$\begin{aligned}
\langle\hat{y}_{t+1} - y_\star, \mathbf{L}\hat{x}_{t+1} - (\psi^*)'(\hat{y}_{t+1})\rangle &= \langle\hat{y}_{t+1} - y_\star, \mathbf{L}(\hat{x}_{t+1} - x_\star) - (\psi^*)'(\hat{y}_{t+1}) + (\psi^*)'(y_\star)\rangle \\
&\leq \langle\hat{y}_{t+1} - y_\star, \mathbf{L}(\hat{x}_{t+1} - x_\star)\rangle.
\end{aligned}$$

Combining the recursions for the iterates x_{t+1} and y_{t+1}, we obtain

$$\begin{aligned}
\mathbb{E}\left[\Phi_{t+1}\right] &= \mathbb{E}\left[\|x_{t+1} - x_\star\|^2 + \frac{\gamma}{p\tau}\|y_{t+1} - y_\star\|^2\right] \\
&\leq \|w_t - w_\star\|^2 - 2\gamma\langle\hat{x}_{t+1} - x_\star, \mathbf{L}^\top(\hat{y}_{t+1} - y_\star)\rangle - \gamma^2\|\mathbf{L}^\top(y_t - y_\star)\|^2 + \frac{\gamma^2}{p}\|\mathbf{L}^\top(\hat{y}_{t+1} - y_t)\|^2 \\
&\quad + \frac{\gamma}{p\tau}\|y_t - y_\star\|^2 + 2\gamma\langle\hat{y}_{t+1} - y_\star, \mathbf{L}(\hat{x}_{t+1} - x_\star)\rangle - \frac{\gamma}{\tau}\|\hat{y}_{t+1} - y_t\|^2 \\
&= \|w_t - w_\star\|^2 + \frac{\gamma}{p\tau}\|y_t - y_\star\|^2 - \gamma^2\|\mathbf{L}^\top(y_t - y_\star)\|^2 + \frac{\gamma^2}{p}\|\mathbf{L}^\top(\hat{y}_{t+1} - y_t)\|^2 - \frac{\gamma}{\tau}\|\hat{y}_{t+1} - y_t\|^2.
\end{aligned}$$

Using the assumption that $\tau \le \frac{p}{\gamma\|\mathbf{L}\mathbf{L}^\top\|}$, we get

$$\frac{\gamma^2}{p}\|\mathbf{L}^\top(\hat{y}_{t+1} - y_t)\|^2 \le \frac{\gamma^2}{p}\|\mathbf{L}\mathbf{L}^\top\|\|\hat{y}_{t+1} - y_t\|^2 \le \frac{\gamma}{\tau}\|\hat{y}_{t+1} - y_t\|^2.$$

Plugging this back, we derive

$$\mathbb{E}\left[\Phi_{t+1}\right] \le \|w_t - w_\star\|^2 + \frac{\gamma}{p\tau}\|y_t - y_\star\|^2 - \gamma^2\|\mathbf{L}^\top(y_t - y_\star)\|^2 \overset{(14)}{\le} (1-\gamma\mu)\|x_t - x_*\|^2 + \frac{\gamma}{p\tau}\|y_t - y_\star\|^2 - \gamma^2\|\mathbf{L}^\top(y_t - y_\star)\|^2.$$

Since we assume that $\partial\psi^*(y) \subseteq \mathrm{range}(\mathbf{L})$ and $y_0 = 0 \in \mathbb{R}^d$, we have that $y_t - y_\star \in \mathrm{range}(\mathbf{L})$. Therefore, $\|\mathbf{L}^\top(y_t - y_\star)\|^2 \ge \lambda_{\min}^+(\mathbf{L}\mathbf{L}^\top)\|y_t - y_\star\|^2$, where $\lambda_{\min}^+$ is the smallest positive eigenvalue. Combining these results, we get

$$\begin{aligned}\mathbb{E}\left[\|x_t - x_\star\|^2\right] &\le \mathbb{E}\left[\Phi_t\right] \le (1-\gamma\mu)\|x_{t-1} - x_\star\|^2 + \frac{\gamma}{p\tau}\left(1 - p\gamma\tau\lambda_{\min}^+(\mathbf{L}\mathbf{L}^\top)\right)\|y_{t-1} - y_\star\|^2 \\ &\le (1 - \min(\gamma\mu, p\gamma\tau\lambda_{\min}^+(\mathbf{L}\mathbf{L}^\top))^t\Phi_0. \end{aligned}$$ □

D.1. Proof of Theorem 5.7

Proof. To obtain the communication step as a special case of the proximity operator, we set ψ to be the indicator function of the singleton $\{0\} \subseteq \mathbb{R}^d$,

$$\psi(x) = \begin{cases} 0 & x = 0 \\ +\infty & x \ne 0 \end{cases}.$$

Its Fenchel conjugate equals, by definition, $\psi^*(y) = \sup_{x\in\mathbb{R}^d}\{\langle x, y\rangle - \psi(x)\} = \langle 0, y\rangle = 0$. Therefore, for any y, $\mathrm{prox}_{\tau\psi^*}(y) = y$ and $\partial\psi^*(y) = \{0\} \subseteq \mathrm{range}(\mathbf{L})$, and the conditions of Theorem D.1 hold. Next, let us establish that Algorithm 5 is a special case of Algorithm 4. If we consider the iterates of Algorithm 4 and define $h_t \coloneqq -\mathbf{L}^\top y_t$, then its first step can be rewritten as

$$\hat{x}_{t+1} = x_t - \gamma(\nabla f(x_t) + \mathbf{L}^\top y_t) = x_t - \gamma(\nabla f(x_t) - h_t),$$

which is exactly the first step of Algorithm 5. The second step of Algorithm 4 is either to do nothing or to update y_{t+1}. Using the fact that $\mathrm{prox}_{\tau\psi^*}(y_t + \tau\mathbf{L}\hat{x}_{t+1}) = y_t + \tau\mathbf{L}\hat{x}_{t+1}$, it is easy to see that

$$h_{t+1} \coloneqq -\mathbf{L}^\top y_{t+1} = -\mathbf{L}^\top(y_t + \tau\mathbf{L}\hat{x}_{t+1}) = h_t - \tau\mathbf{L}^\top\mathbf{L}\hat{x}_{t+1}.$$

By setting $\mathbf{L} = (\mathbf{I} - \mathbf{W})^{1/2}$, we get $\mathbf{L}^\top\mathbf{L} = \mathbf{I} - \mathbf{W}$, and we recover the second step of Algorithm 5 in an equivalent form:

$$\begin{cases} h_{i,t+1} &= h_{i,t} + \tau\left(\hat{x}_{i,t+1} - \sum_{j=1}^n W_{ij}\hat{x}_{j,t+1}\right), \\ x_{i,t+1} &= \hat{x}_{i,t+1} + \frac{\gamma}{p}(h_{i,t+1} - h_{i,t}). \end{cases} \iff \begin{cases} x_{i,t+1} &= \left(1 - \frac{\gamma\tau}{p}\right)\hat{x}_{i,t+1} + \frac{\gamma\tau}{p}\sum_{j=1}^n W_{ij}\hat{x}_{j,t+1}, \\ h_{i,t+1} &= h_{i,t} + \frac{p}{\gamma}(x_{i,t+1} - \hat{x}_{i,t+1}). \end{cases}$$

Finally, notice that $\lambda_{\min}^+(\mathbf{L}^\top\mathbf{L}) = \lambda_{\min}^+(\mathbf{I} - \mathbf{W}) = 1 - \lambda_2(\mathbf{W}) = \delta$, $\|\mathbf{L}^\top\mathbf{L}\| = \|\mathbf{I} - \mathbf{W}\| < 1$, and $y_{i,0} = 0$, so applying Theorem D.1 yields

$$\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n\|x_{i,T} - x_\star\|^2\right] \le (1-\zeta)^T\Phi_0 = (1-\zeta)^T\left(\|x_0 - x_\star\|^2 + \frac{\gamma}{p\tau}\frac{1}{n}\sum_{i=1}^n\|y_{i,*}\|^2\right).$$

By Jensen's inequality, the average iterate $\overline{x}_T \coloneqq \frac{1}{n}\sum_{i=1}^n x_{i,T}$ satisfies

$$\mathbb{E}\left[\|\overline{x}_T - x_\star\|^2\right] \le \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n\|x_{i,T} - x_\star\|^2\right] \le (1-\zeta)^T\left(\|x_0 - x_\star\|^2 + \frac{\gamma}{p\tau}\frac{1}{n}\sum_{i=1}^n\|y_{i,*}\|^2\right).$$

Finally, notice that $\|y_{i,*}\|^2 = \|\mathbf{L}^\dagger\mathbf{L}y_{i,*}\|^2 = \|\mathbf{L}^\dagger\nabla f_i(x_*)\|^2 \le \frac{1}{\lambda_{\min}^+(\mathbf{L}^\top\mathbf{L})}\|\nabla f_i(x_*)\|^2 = \frac{1}{\delta}\|\nabla f_i(x_*)\|^2$. □

E. Additional Experiments

In this experiment we compare Scaffnew methods with different parameters of probability p. Note that the expected value of the number of local steps in between two communication rounds is equal to $\frac{1}{p}$.

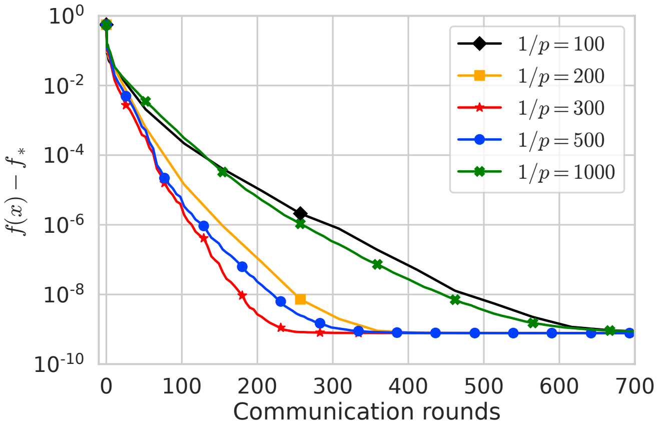


Figure 3. The communication complexity of Scaffnew with several different choices of parameter p in the deterministic case (i.e., when full gradients are computed rather than stochastic gradients).

As we can see in Figure 3, if our method communicates either too often ($^1/_p = 100$) or too rarely ($^1/_p = 1000$), convergence suffers. The optimal number of local steps in this experiment is $^1/_p = 300$. This value is close to $\sqrt{\hat{\kappa}} \approx 100$, where $\hat{\kappa}$ is an estimate of the true condition number κ, which can be smaller than $\hat{\kappa}$. Our theory predicted that the choice $p = \frac{1}{\sqrt{\kappa}}$ is optimal up to constant factors, which is close the experiment's results. Thus, whenever an estimate of the conditioning number is available, our estimate provides a practical recipe for the number of local steps that provides decent empirical performance without any tuning.