KAUST
King Abdullah University of Science and Technology

Inria

CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

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

**Peter Richtárik**

ICML 2022, Baltimore, Maryland, USA

**Konstantin Mishchenko**
KAUST Inria

**Grigory Malinovsky**
KAUST

**Sebastian Stich**
CISPA

# Optimization Formulation of Federated Learning

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

# 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

# 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

# Optimization Formulation of Federated Learning

# devices / machines

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

# 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 Local Gradient Descent

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

**Optimization problem:**

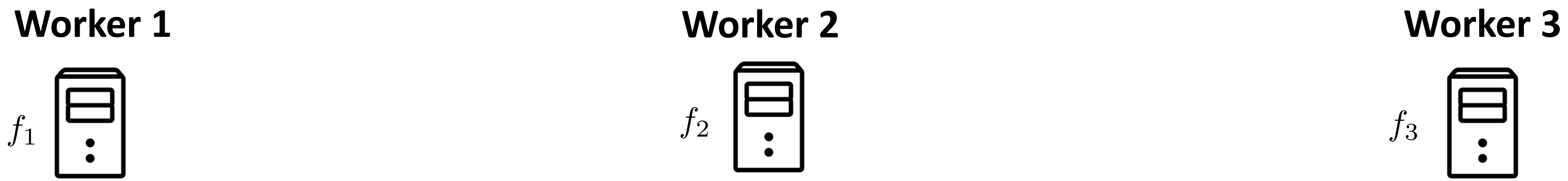
$$\min_{x \in \mathbb{R}^d} f(x) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} f_i(x)$$

# Distributed Local Gradient Descent

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

**Optimization problem:**

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

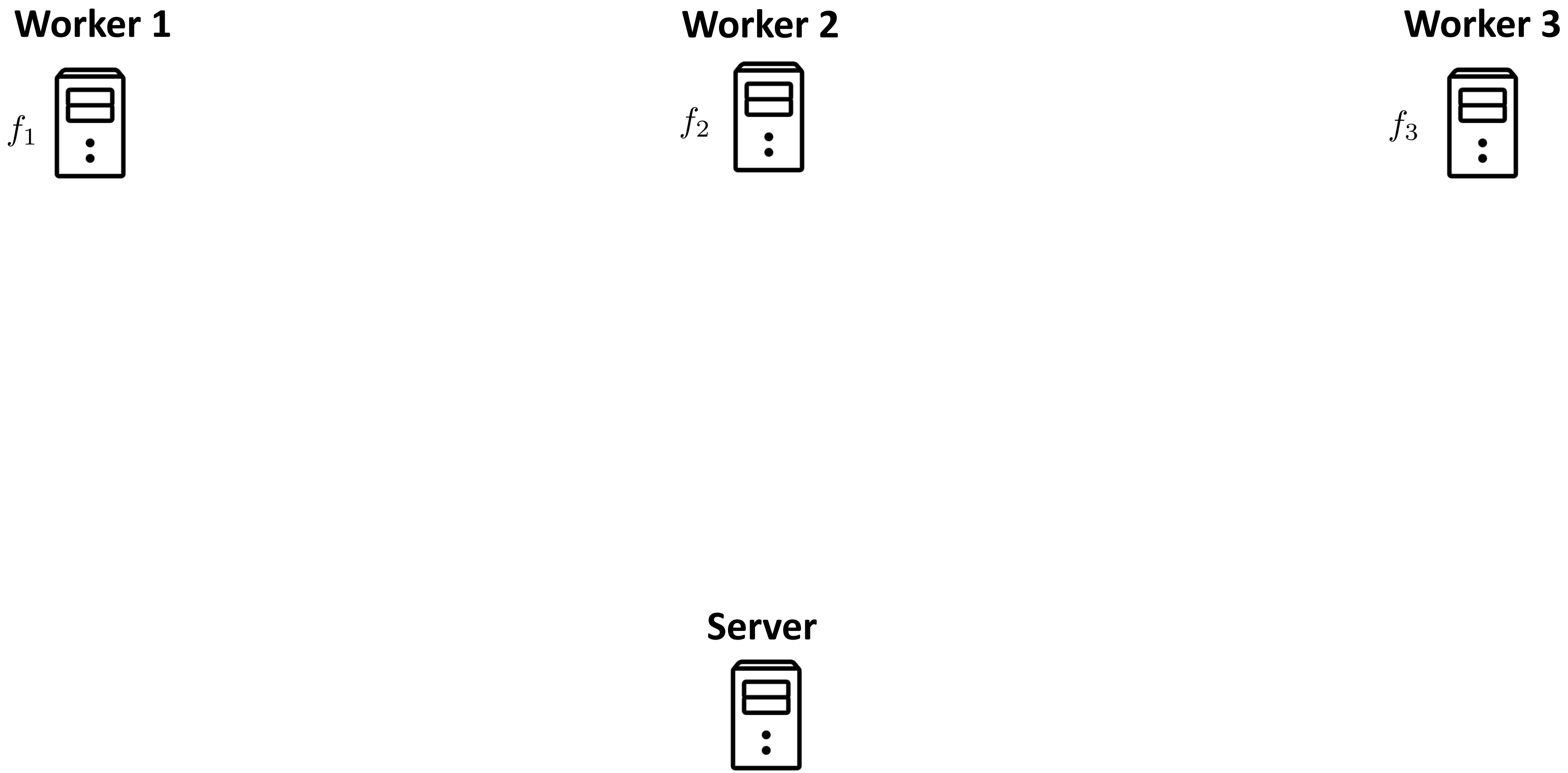


# Distributed Local Gradient Descent

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

**Optimization problem:**

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



# Distributed Local Gradient Descent

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

**Optimization problem:**

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

**Worker 1**

$f_1$

Receive $x_t$ from the server

**Worker 2**

$f_2$

Receive $x_t$ from the server

**Worker 3**

$f_3$

Receive $x_t$ from the server

**Server**

# Distributed Local Gradient Descent

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

**Optimization problem:**

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

**Worker 1**

$f_1$

Receive $x_t$ from the server

$x_{1,t} = x_t$

**Worker 2**

$f_2$

Receive $x_t$ from the server

$x_{2,t} = x_t$

**Worker 3**

$f_3$

Receive $x_t$ from the server

$x_{3,t} = x_t$

**Server**

# Distributed Local Gradient Descent

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

**Optimization problem:**

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

**Worker 1**

$f_1$

Receive $x_t$ from the server

$x_{1,t} = x_t$

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

**Worker 2**

$f_2$

Receive $x_t$ from the server

$x_{2,t} = x_t$

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

**Worker 3**

$f_3$

Receive $x_t$ from the server

$x_{3,t} = x_t$

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

**Server**

# Distributed Local Gradient Descent

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

**Optimization problem:**

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

**Worker 1**

$f_1$

$\text{Receive } x_t \text{ from the server}$

$x_{1,t} = x_t$

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

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

**Worker 2**

$f_2$

$\text{Receive } x_t \text{ from the server}$

$x_{2,t} = x_t$

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

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

**Worker 3**

$f_3$

$\text{Receive } x_t \text{ from the server}$

$x_{3,t} = x_t$

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

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

**Server**

# Distributed Local Gradient Descent

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

**Optimization problem:**

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

**Worker 1**

$f_1$

Receive $x_t$ from the server

$x_{1,t} = x_t$

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

**Worker 2**

$f_2$

Receive $x_t$ from the server

$x_{2,t} = x_t$

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

**Worker 3**

$f_3$

Receive $x_t$ from the server

$x_{3,t} = x_t$

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

**Server**

# Distributed Local Gradient Descent

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

**Optimization problem:**

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

**Worker 1**

$f_1$

Receive $x_t$ from the server

$x_{1,t} = x_t$

$$\begin{aligned} x_{1,t+1} &= x_{1,t} - \gamma\nabla f_1(x_{1,t}) \\ x_{1,t+2} &= x_{1,t+1} - \gamma\nabla f_1(x_{1,t+1}) \\ &\vdots \\ x_{1,t+K} &= x_{1,t+K-1} - \gamma\nabla f_1(x_{1,t+K-1}) \end{aligned}$$

**Worker 2**

$f_2$

Receive $x_t$ from the server

$x_{2,t} = x_t$

$$\begin{aligned} x_{2,t+1} &= x_{2,t} - \gamma\nabla f_2(x_{2,t}) \\ x_{2,t+2} &= x_{2,t+1} - \gamma\nabla f_2(x_{2,t+1}) \\ &\vdots \\ x_{2,t+K} &= x_{2,t+K-1} - \gamma\nabla f_2(x_{2,t+K-1}) \end{aligned}$$

**Worker 3**

$f_3$

Receive $x_t$ from the server

$x_{3,t} = x_t$

$$\begin{aligned} x_{3,t+1} &= x_{3,t} - \gamma\nabla f_3(x_{3,t}) \\ x_{3,t+2} &= x_{3,t+1} - \gamma\nabla f_3(x_{3,t+1}) \\ &\vdots \\ x_{3,t+K} &= x_{3,t+K-1} - \gamma\nabla f_3(x_{3,t+K-1}) \end{aligned}$$

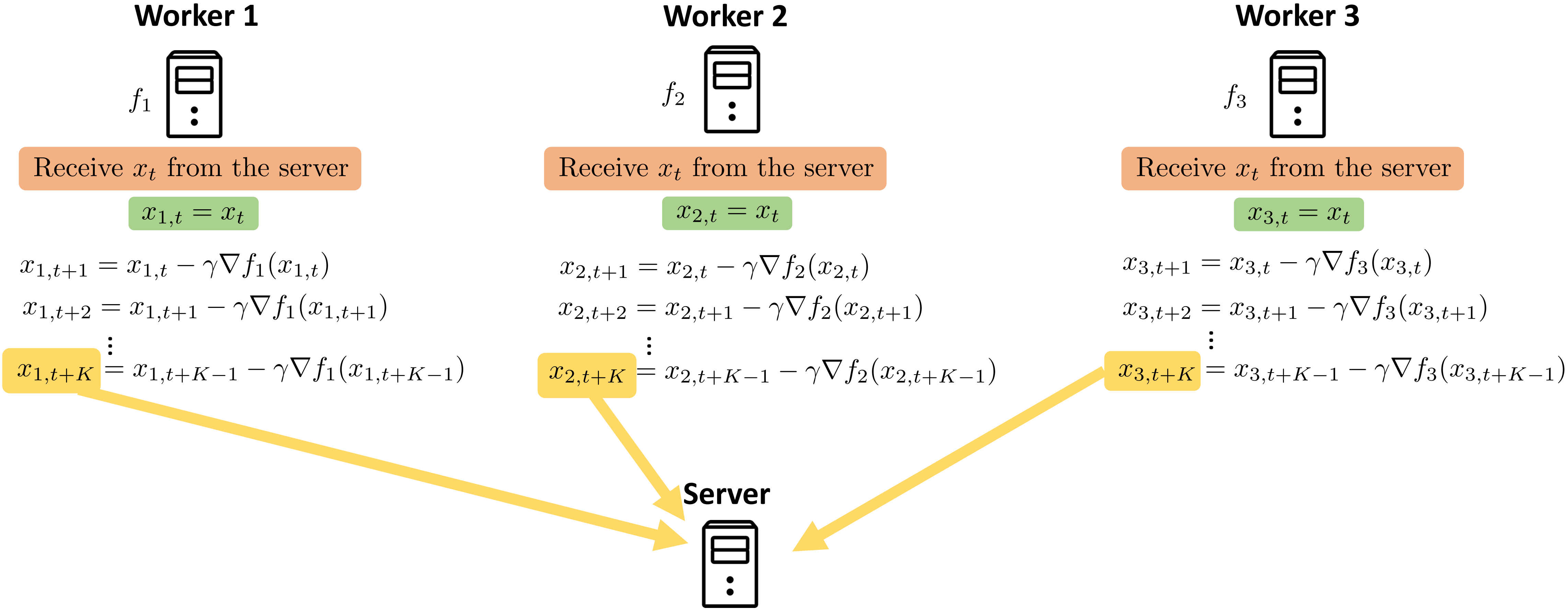
**Server**

# Distributed Local Gradient Descent

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

**Optimization problem:**

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



# Distributed Local Gradient Descent

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

**Optimization problem:**

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

**Worker 1**

$f_1$

Receive $x_t$ from the server

$x_{1,t} = x_t$

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

**Worker 2**

$f_2$

Receive $x_t$ from the server

$x_{2,t} = x_t$

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

**Worker 3**

$f_3$

Receive $x_t$ from the server

$x_{3,t} = x_t$

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

**Server**

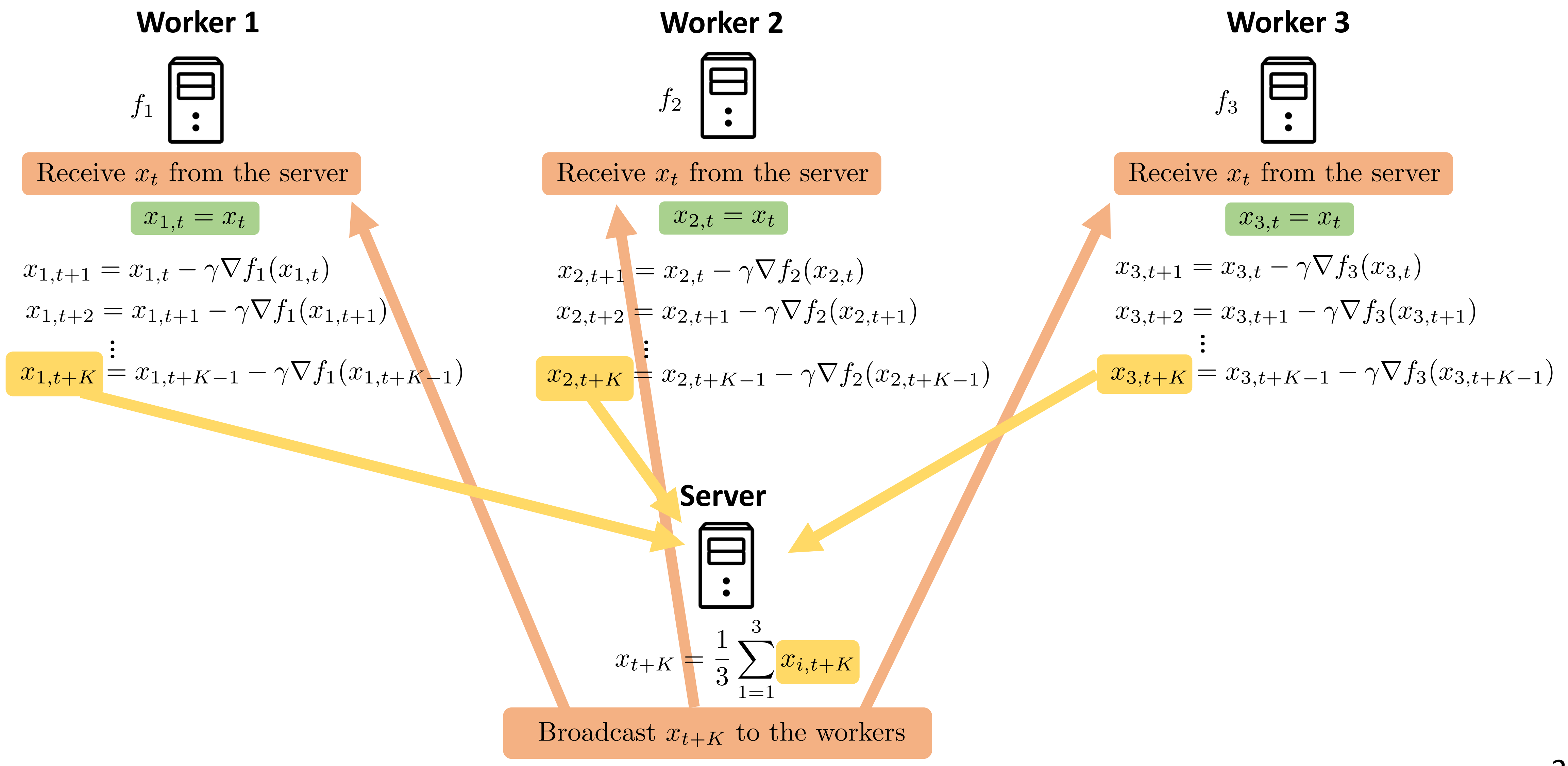
$$x_{t+K} = \frac{1}{3} \sum_{1=1}^{3} x_{i,t+K}$$

# Distributed Local Gradient Descent

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

**Optimization problem:**

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



# Key Open Problem in Federated Learning

# Key Open Problem in Federated Learning

**Local training is of key importance in FL: in practice, it significantly improves communication efficiency.**

# Key Open Problem in Federated Learning

**Local training is of key importance in FL: in practice, it significantly improves communication efficiency.**

**However, there is no theoretical result explaining this!**

# Key Open Problem in Federated Learning

**Local training is of key importance in FL: in practice, it 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?**

# Consensus Reformulation

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

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

# 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 $\mathbb{R}^{nd}$

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

# 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 $\mathbb{R}^{nd}$

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

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

# Consensus Reformulation

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

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

$$\Updownarrow$$

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

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

**Bad:** Non-differentiable function

**Good:** Indicator function of a nonempty closed convex set

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

# Key Method: Proximal Gradient Descent

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

# Key Method: Proximal Gradient Descent

stepsize

$$x_t - \boxed{\gamma} \nabla f(x_t)$$

# Key Method: Proximal Gradient Descent

$$\underbrace{x_t - \overbrace{\boxed{\gamma}}^{\text{stepsize}} \nabla f(x_t)}$$

gradient operator

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

# Key Method: Proximal Gradient Descent

stepsize

$$x_{t+1} = \boxed{\mathrm{prox}_{\gamma\psi}} \underbrace{\left(x_t - \gamma \nabla f(x_t)\right)}$$

gradient operator

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

# Key Method: Proximal Gradient Descent

stepsize

$$x_{t+1} = \underbrace{\mathrm{prox}_{\gamma\psi}}_{\text{proximal operator}}\underbrace{(x_t - \gamma \nabla f(x_t))}_{\text{gradient operator}}$$

proximal operator

$x \mapsto \mathrm{prox}_{\gamma\psi}(x)$

gradient operator

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

# Key Method: Proximal Gradient Descent

proximal operator:

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

stepsize

$$x_{t+1} = \underbrace{\mathrm{prox}_{\gamma\psi}}_{\text{proximal operator}}\underbrace{(x_t - \gamma\nabla f(x_t))}_{\text{gradient operator}}$$

proximal operator

$$x \mapsto \mathrm{prox}_{\gamma\psi}(x)$$

gradient operator

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

# Key Method: Proximal Gradient Descent

proximal operator:

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

stepsize

$$x_{t+1} = \underbrace{\mathrm{prox}_{\gamma\psi}}_{\substack{\text{proximal operator} \\ x \mapsto \mathrm{prox}_{\gamma\psi}(x)}} \underbrace{(x_t - \gamma \nabla f(x_t))}_{\substack{\text{gradient operator} \\ x \mapsto x - \gamma\nabla f(x)}}$$

**Key Observation: Prox = Communication!**

# Proximal Gradient Descent: Theory

**Theorem:**

# Proximal Gradient Descent: Theory

**Theorem:**

$$t \geq \frac{L}{\mu} \log \frac{1}{\varepsilon}$$

# Proximal Gradient Descent: Theory

**Theorem:**

$$\boxed{t} \geq \frac{L}{\mu} \log \frac{1}{\varepsilon}$$

# iterations

# Proximal Gradient Descent: Theory

$f$ is $\mu$-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:**

$$\boxed{t} \geq \boxed{\frac{L}{\mu}} \log \frac{1}{\varepsilon}$$

# iterations

# Proximal Gradient Descent: Theory

$f$ is $\mu$-convex and $L$-smooth:

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

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

**Theorem:**

$$t \ge \frac{L}{\mu} \log \frac{1}{\varepsilon}$$

# iterations

Error tolerance

# Proximal Gradient Descent: Theory

$f$ is $\mu$-convex and $L$-smooth:

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

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

**Theorem:**

$$t \ge \frac{L}{\mu}\log\frac{1}{\varepsilon} \quad \Rightarrow$$

\# iterations

Error tolerance

# Proximal Gradient Descent: Theory

$f$ is $\mu$-convex and $L$-smooth:

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

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

**Theorem:**

$$\boxed{t} \ge \boxed{\frac{L}{\mu}} \log \frac{1}{\boxed{\varepsilon}} \quad \Longrightarrow$$

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

# iterations

Error tolerance

# Proximal Gradient Descent: Theory

$f$ is $\mu$-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:**

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

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

# iterations

Error tolerance

# Proximal Gradient Descent: Theory

$f$ is $\mu$-convex and $L$-smooth:

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

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

**Theorem:**

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

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

\# iterations

Error tolerance

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

# ProxSkip: “Do Proximal Gradient Descent, but Skip Most of the Prox Evaluations!”

# ProxSkip: “Do Proximal Gradient Descent, but Skip Most of the Prox Evaluations!”

1

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

# ProxSkip: “Do Proximal Gradient Descent, but Skip Most of the Prox Evaluations!”

1

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

# ProxSkip: "Do Proximal Gradient Descent, but Skip Most of the Prox Evaluations!"

1

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

# ProxSkip: "Do Proximal Gradient Descent, but Skip Most of the Prox Evaluations!"

1

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

# ProxSkip: "Do Proximal Gradient Descent, but Skip Most of the Prox Evaluations!"

1

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

2a with probability $1 - p$ do

$1 - p \approx 1$

2b with probability $p$ do

$p \approx 0$

# ProxSkip: “Do Proximal Gradient Descent, but Skip Most of the Prox Evaluations!”

1

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

2a

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

$1-p \approx 1$

2b

with probability $p$ do

$p \approx 0$

# ProxSkip: "Do Proximal Gradient Descent, but Skip Most of the Prox Evaluations!"

1

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

2a

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

$1 - p \approx 1$

2b

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

with probability $p$ do $\quad x_{t+1} = ? \quad h_{t+1} = ?$

$p \approx 0$

# ProxSkip: Bounding the # of Iterations

**Theorem:**

# ProxSkip: Bounding the # of Iterations

**Theorem:**

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

# ProxSkip: Bounding the # of Iterations

**Theorem:**

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

# iterations

# ProxSkip: Bounding the # of Iterations

**Theorem:**

$f$ is $\mu$-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 $f$

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

# iterations

# ProxSkip: Bounding the # of Iterations

**Theorem:**

$f$ is $\mu$-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 $f$

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

# iterations

*p* = probability of evaluating the prox

# ProxSkip: Bounding the # of Iterations

**Theorem:**

$f$ is $\mu$-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 $f$

$$t \geq \max\left\{\frac{L}{\mu}, \frac{1}{p^2}\right\} \log \frac{1}{\varepsilon} \quad \Rightarrow$$

# iterations

$p$ = probability of evaluating the prox

# ProxSkip: Bounding the # of Iterations

**Theorem:**

$f$ is $\mu$-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

# ProxSkip: Bounding the # of Iterations

**Theorem:**

$f$ is $\mu$-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:

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

# 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:

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

# 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:

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

# 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:

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

# 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:

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

# 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 $f$

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

# 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 $p \cdot \frac{L}{\mu} = \frac{1}{p}$

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

# 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:

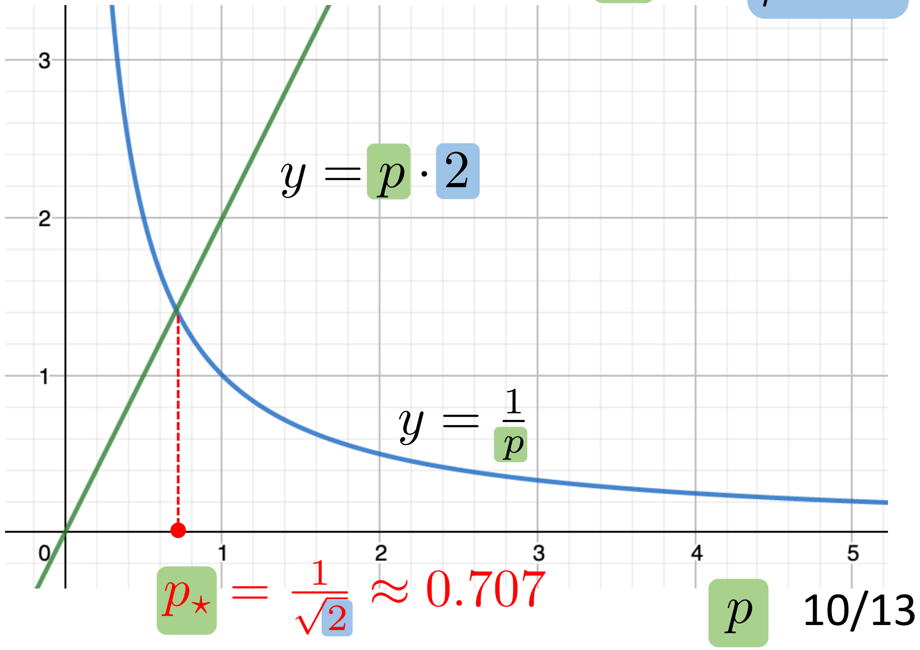
$\frac{L}{\mu}$ is the condition number of $f$

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

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 $\frac{L}{\mu} = 2$



# 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) | $\tau$ | $d$ | $\frac{1}{\tau L}$ | ✗ | $\mathcal{O}\left(\frac{G^2}{\mu n \tau \varepsilon}\right)$ [d] | ✗ |
| Scaffold (Karimireddy et al., 2020) | $\tau$ | $2d$ | $\frac{1}{\tau L}$ [e] | ✓ | $\tilde{\mathcal{O}}(\kappa)$ [c] | ✗ |
| S-Local-GD [a] (Gorbunov et al., 2021) | $\tau$ | $d < \# < 2d$ [f] | $\frac{1}{\tau L}$ | ✓ | $\tilde{\mathcal{O}}(\kappa)$ | ✗ |
| FedLin [b] (Mitra et al., 2021) | $\tau_i$ | $2d$ | $\frac{1}{\tau_i L}$ | ✓ | $\tilde{\mathcal{O}}(\kappa)$ [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.

# Scaffnew (=ProxSkip applied to FL) vs Baselines

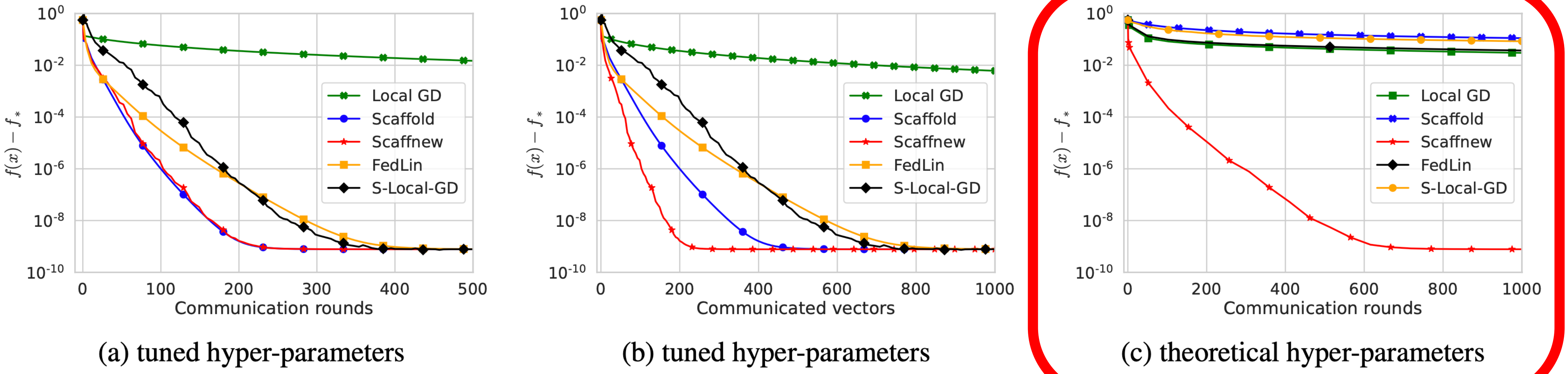



*Figure 1.* **Deterministic Problem**. Comparison of Scaffnew to other local update methods that tackle data-heterogeneity and to LocalGD. In (a) we compare communication rounds with optimally tuned hyper-parameters. In (b) we compare communicated vectors (Scaffold, FedLin and S-Local-GD require transmission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

**L2-regularized logistic regression:**

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

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

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

# Scaffnew (=ProxSkip applied to FL) vs Nesterov