Communication-efficient distributed optimization algorithms

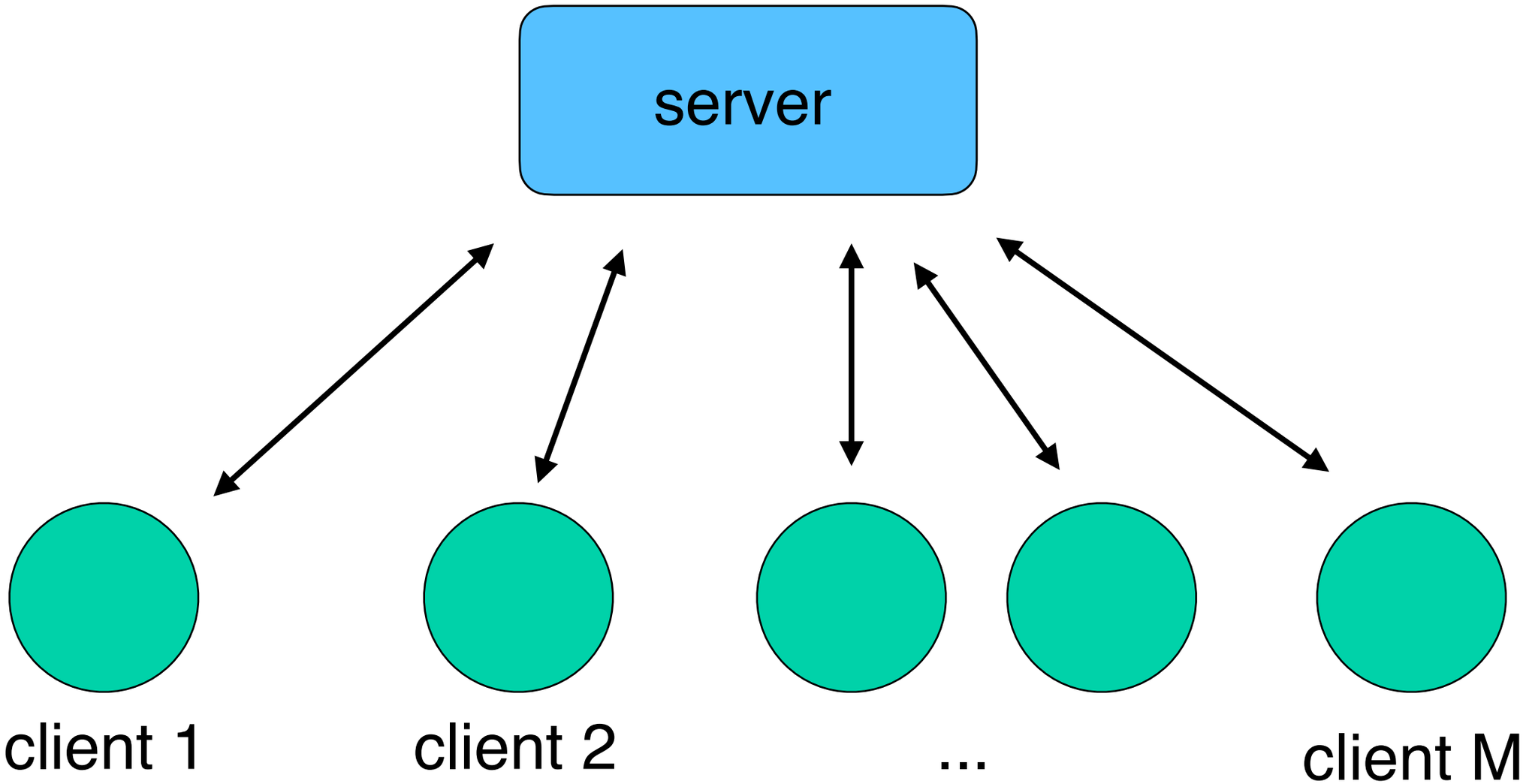
Laurent Condat

Peter Richtárik

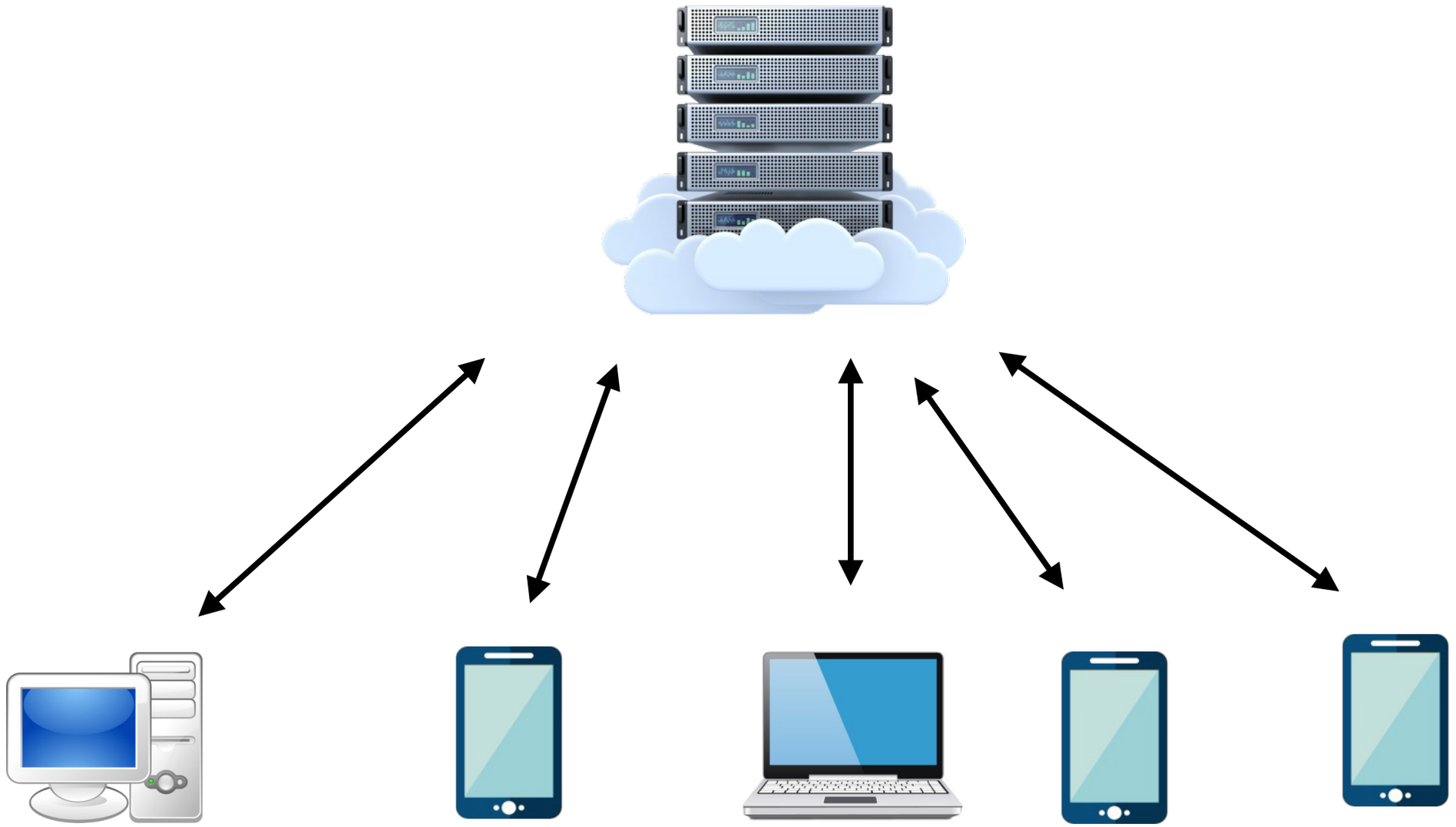
King Abdullah Univ. of Science and Technology (KAUST)
Saudi Arabia

IFORS, July 2023

Distributed optimization



Federated learning



Federated learning

Convex problem

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \; R(x) + \frac{1}{M} \sum_{m=1}^{M} F_m(x)$$

- F_m is L-smooth and μ-strongly convex ($F - \frac{\mu}{2} \| \cdot \|^2$ convex), with $L \geq \mu > 0$

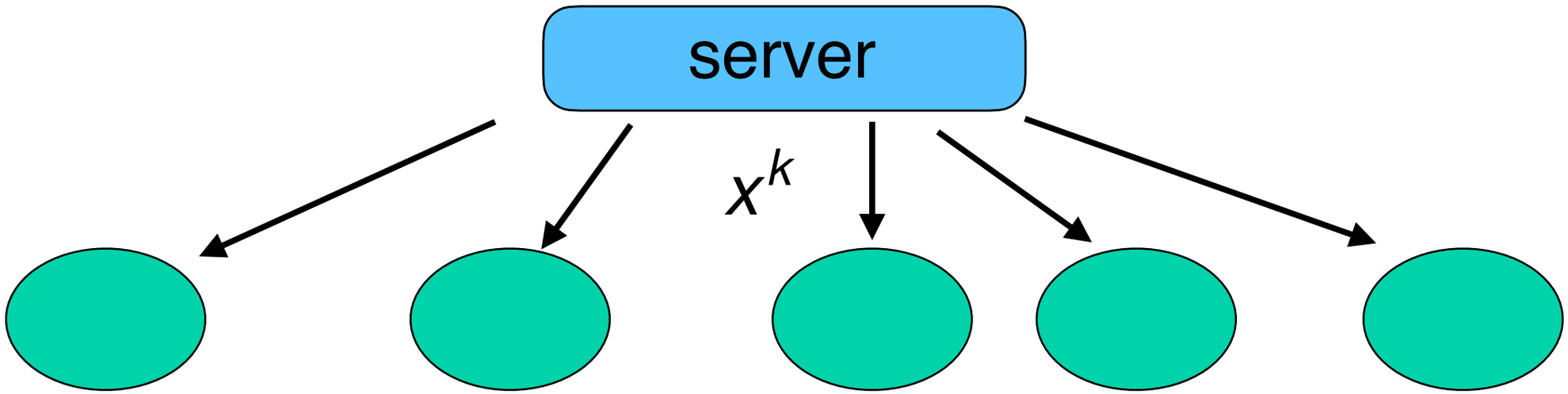
Convex problem

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \; R(x) + \frac{1}{M} \sum_{m=1}^{M} F_m(x)$$

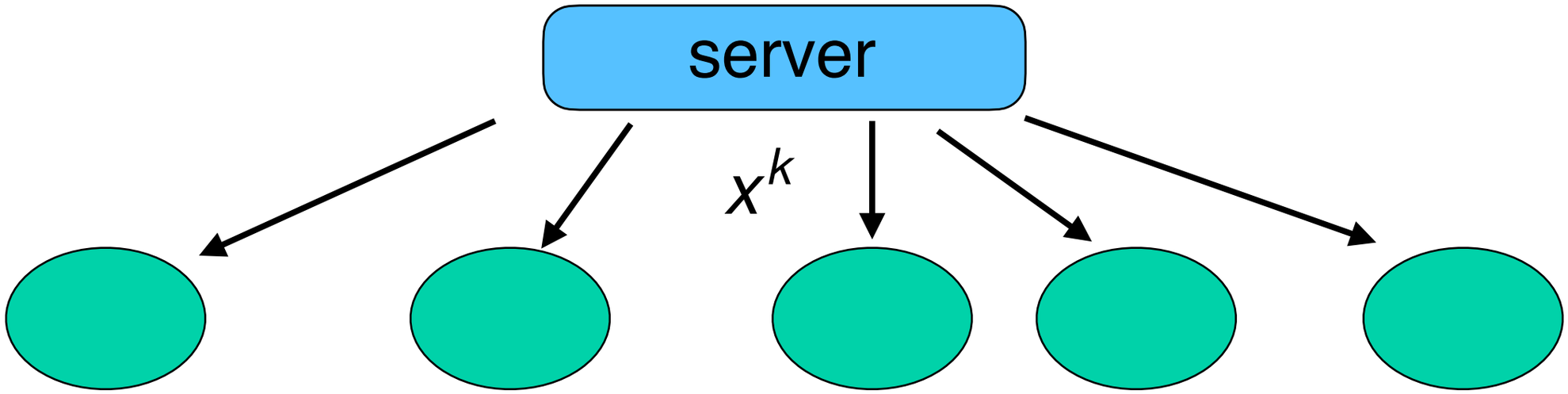
- F_m is L-smooth and μ-strongly convex ($F - \frac{\mu}{2} \| \cdot \|^2$ convex), with $L \geq \mu > 0$
- $R : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$ is a proper, closed, convex function with proximity operator

$$\text{prox}_{\gamma R} : x \mapsto \arg\min_w \left(\gamma R(w) + \frac{1}{2} \|x - w\|^2 \right)$$

Distributed prox. GD

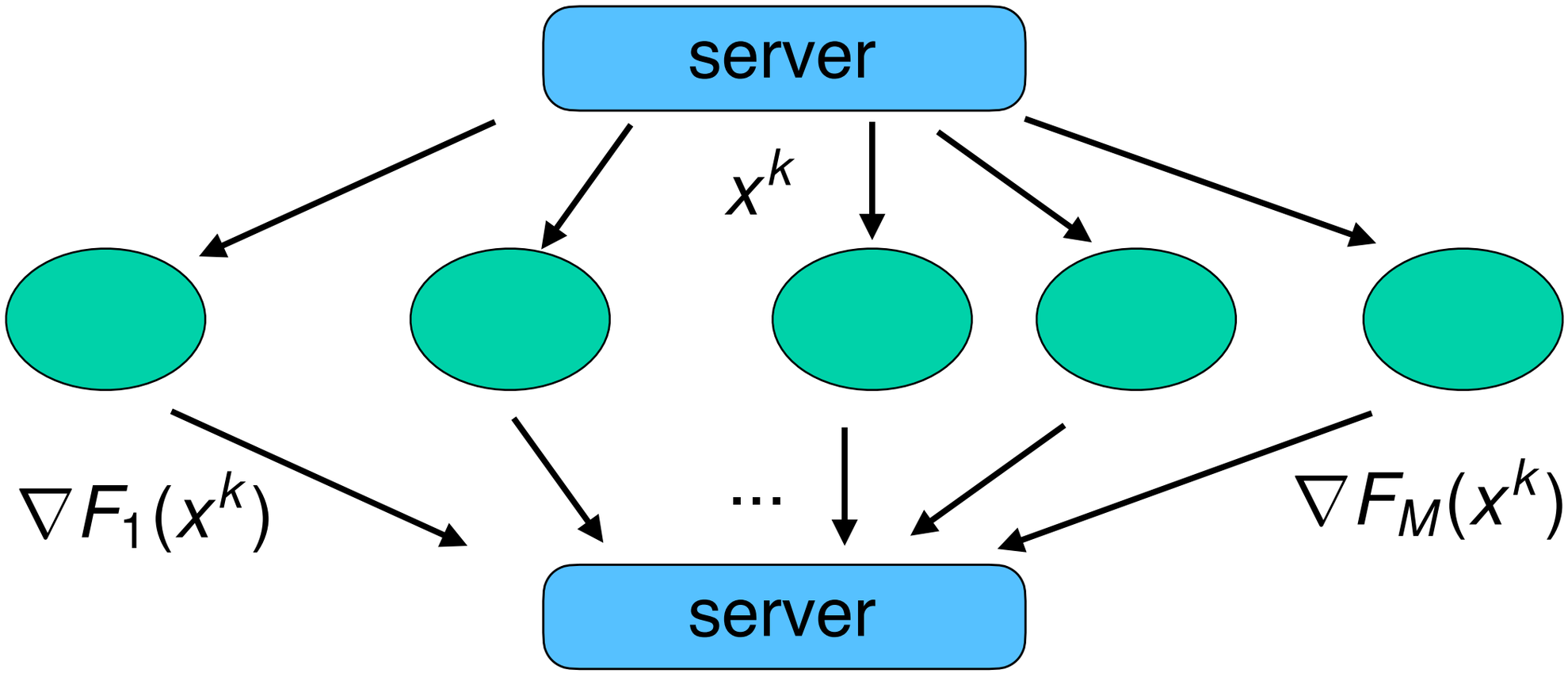


Distributed prox. GD

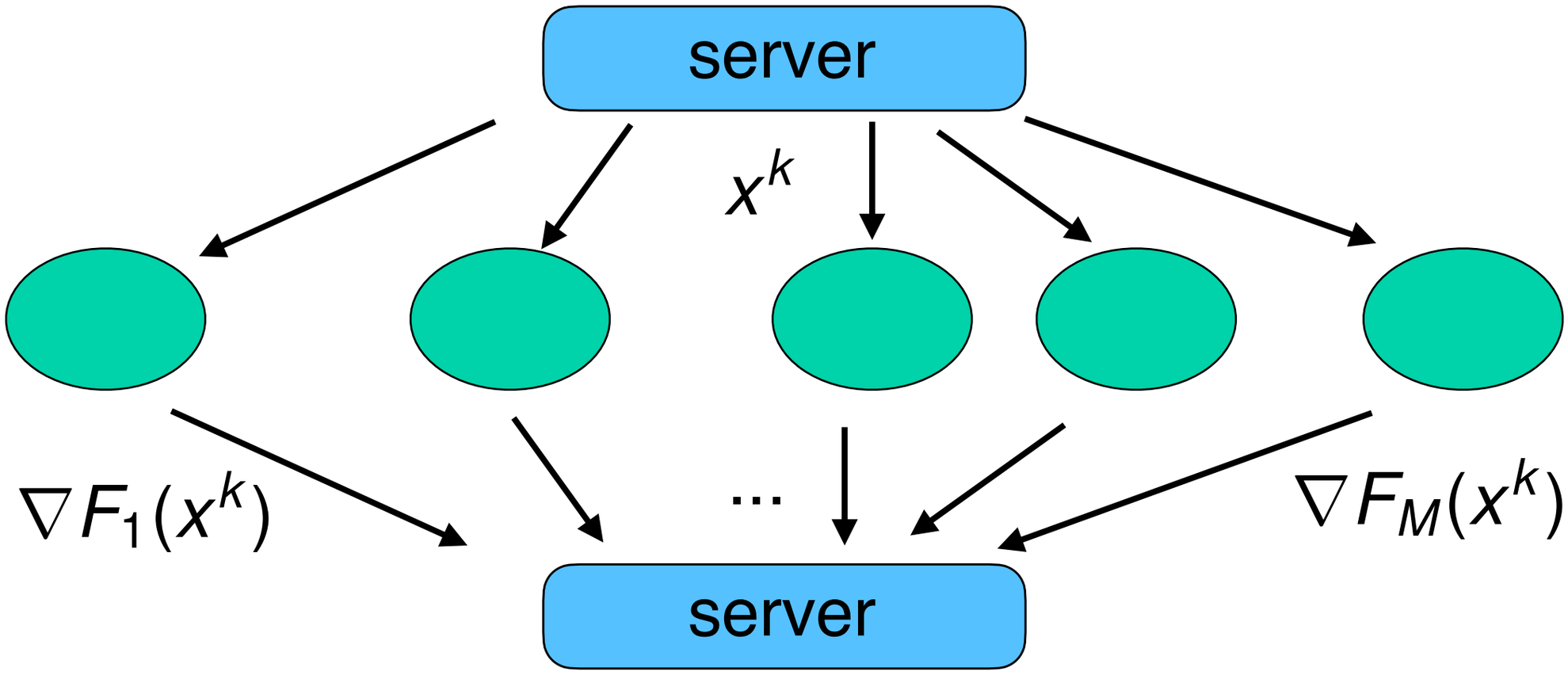


$\nabla F_1(x^k)$ ... $\nabla F_M(x^k)$

Distributed prox. GD

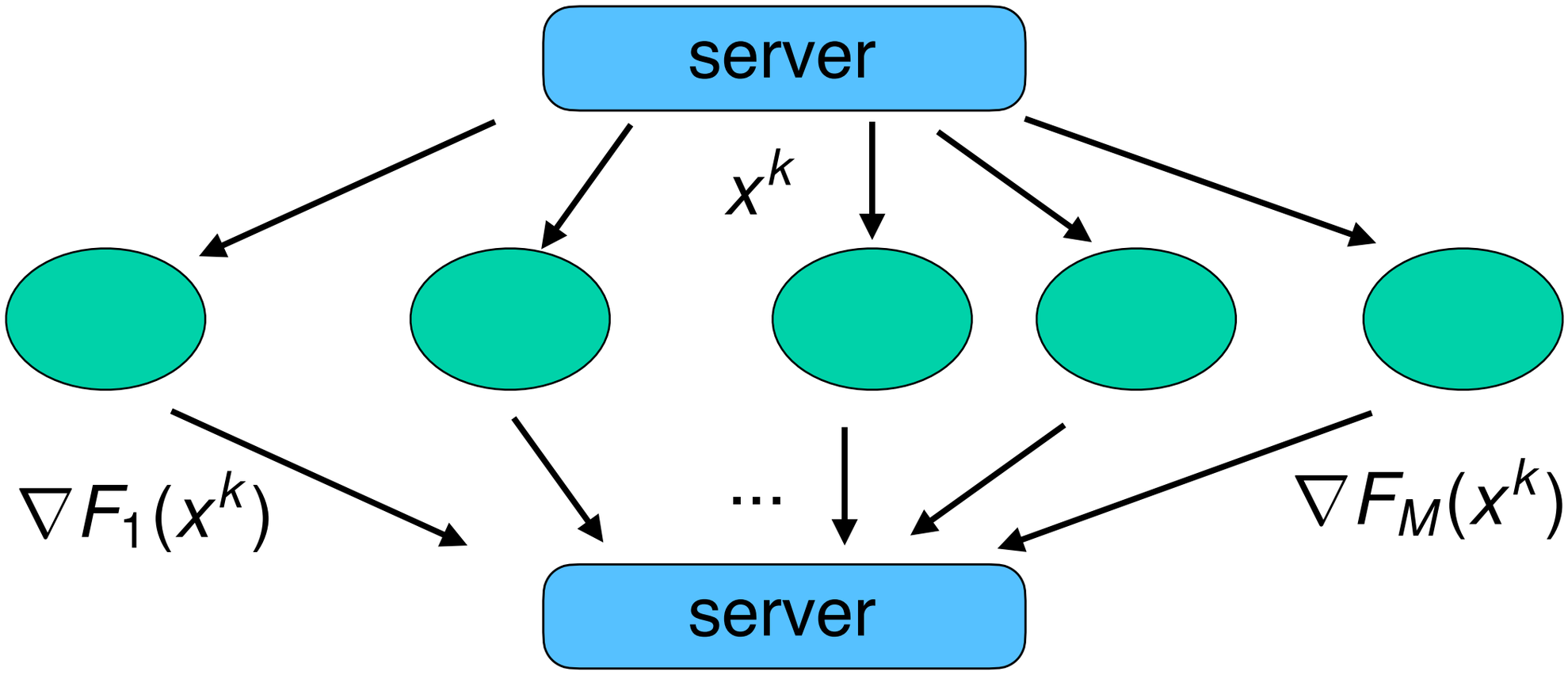


Distributed prox. GD



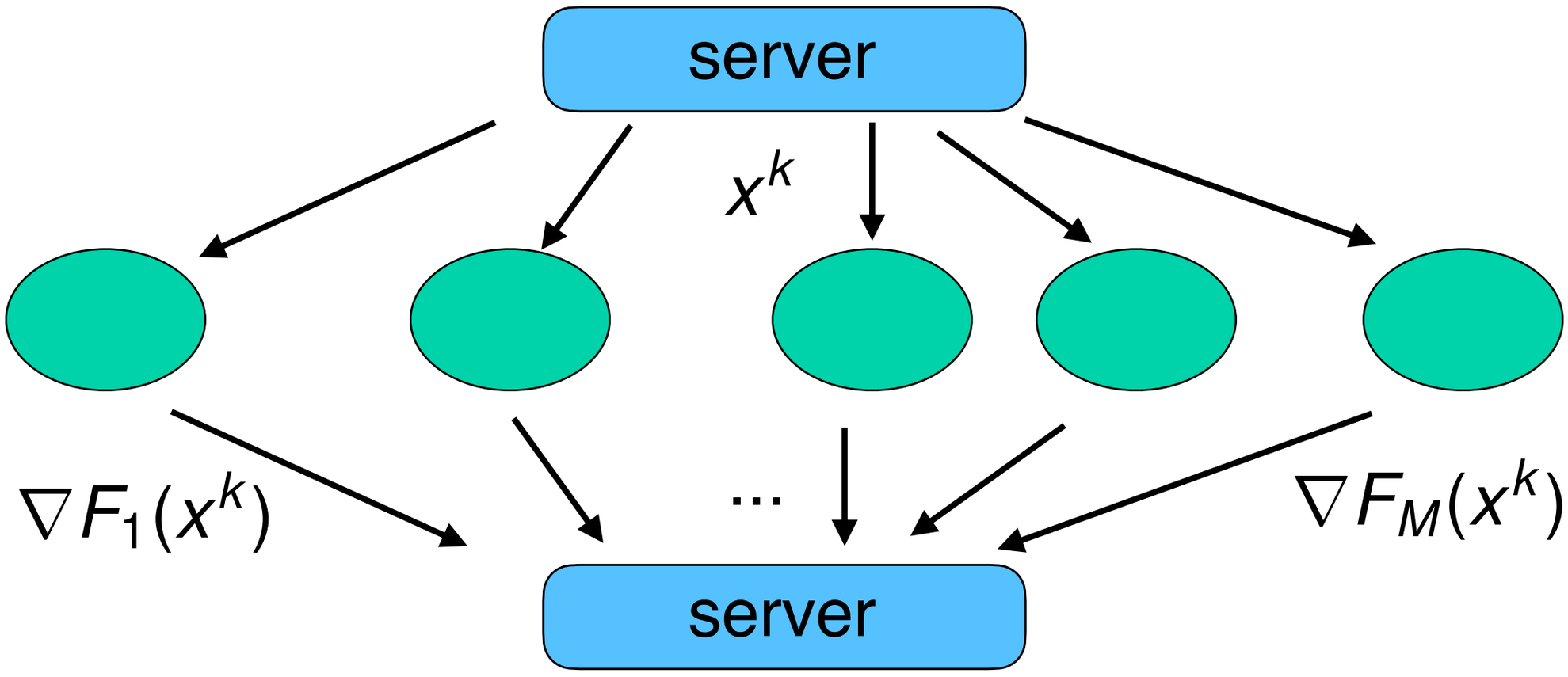
$$\frac{1}{M}\sum_{m=1}^{M}\nabla F_m(x^k)$$

Distributed prox. GD



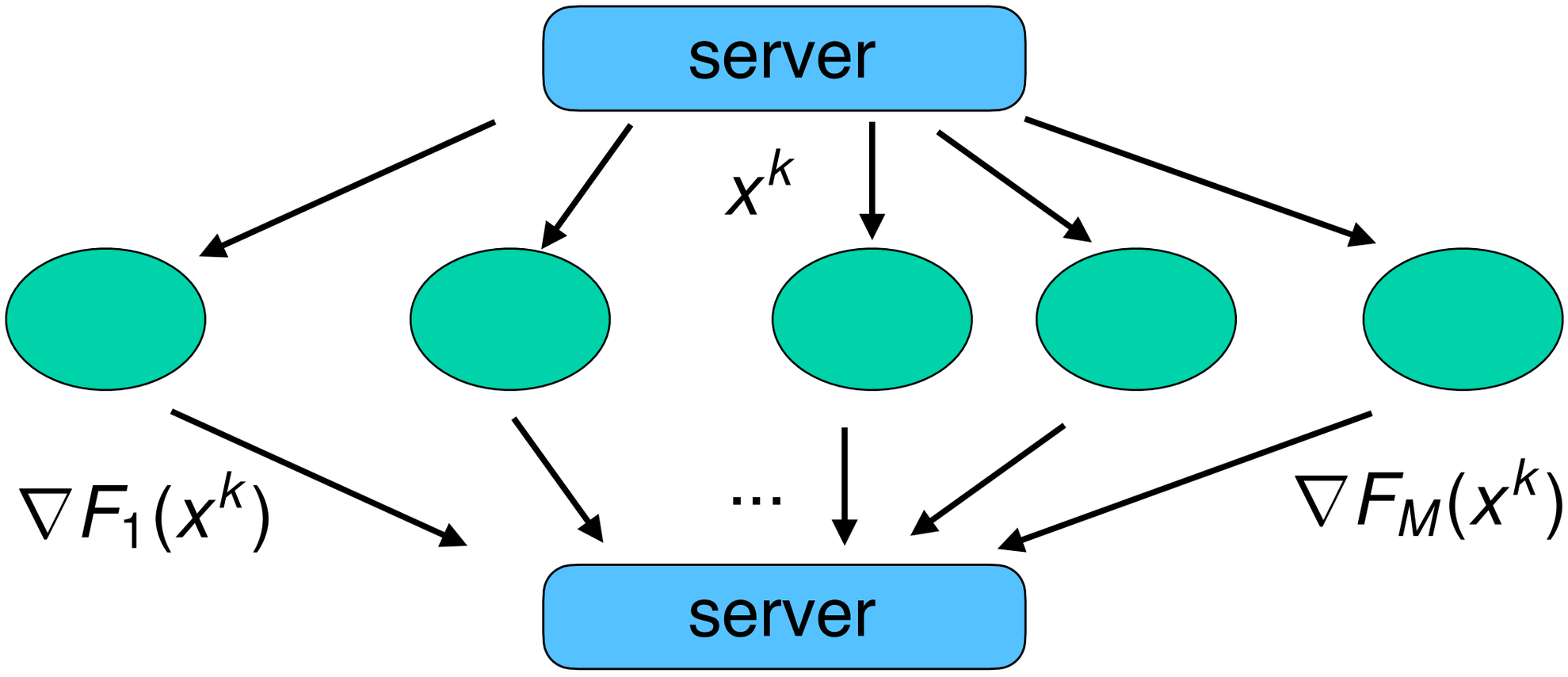
$$x^k - \frac{\gamma}{M} \sum_{m=1}^{M} \nabla F_m(x^k)$$

Distributed prox. GD



$$x^{k+1} := \mathrm{prox}_{\gamma R}\left(x^k - \frac{\gamma}{M}\sum_{m=1}^{M}\nabla F_m(x^k)\right)$$

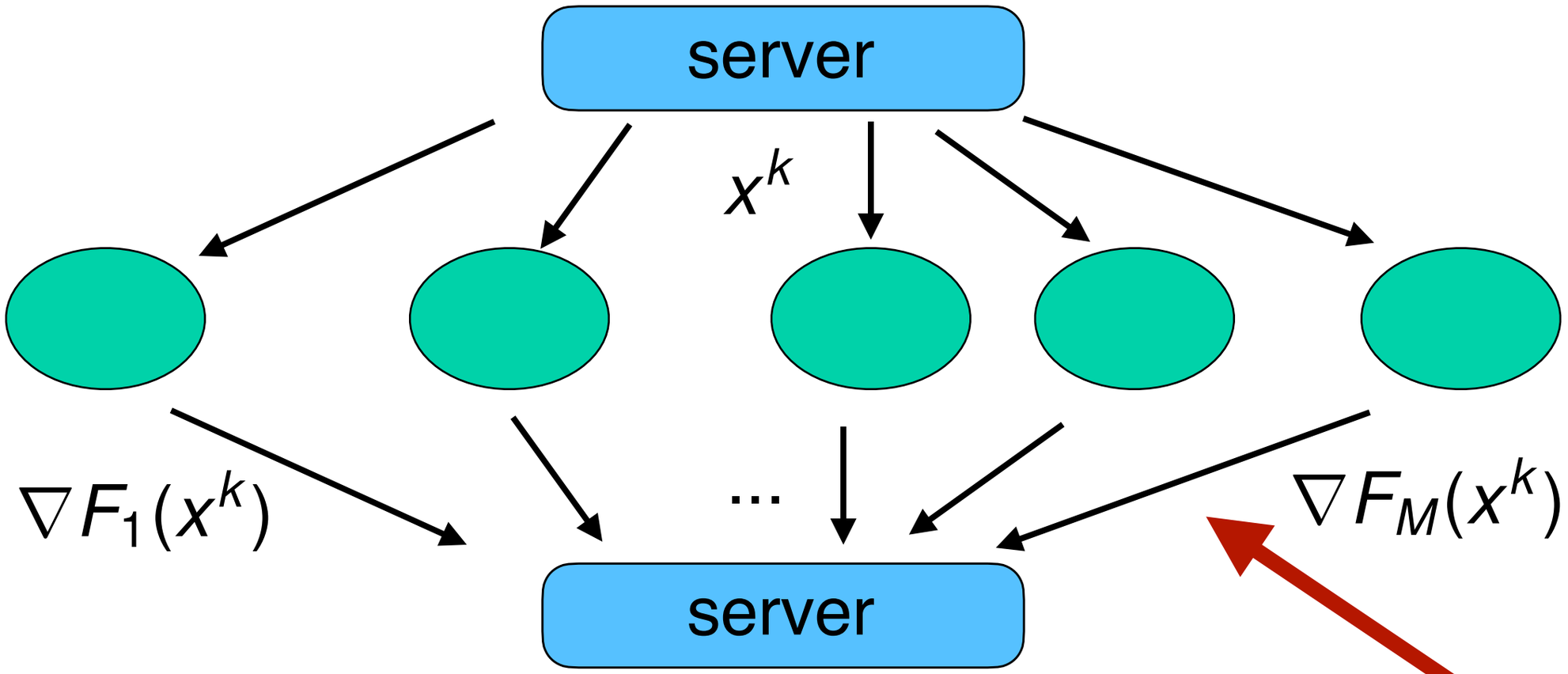
Distributed prox. GD



$$x^{k+1} := \mathrm{prox}_{\gamma R}\left(x^k - \frac{\gamma}{M}\sum_{m=1}^{M}\nabla F_m(x^k)\right)$$

$$0 < \gamma \leq \frac{2}{L+\mu} \quad \text{☞} \quad \|x^k - x^\star\| \leq (1-\gamma\mu)^k \|x^0 - x^\star\|$$

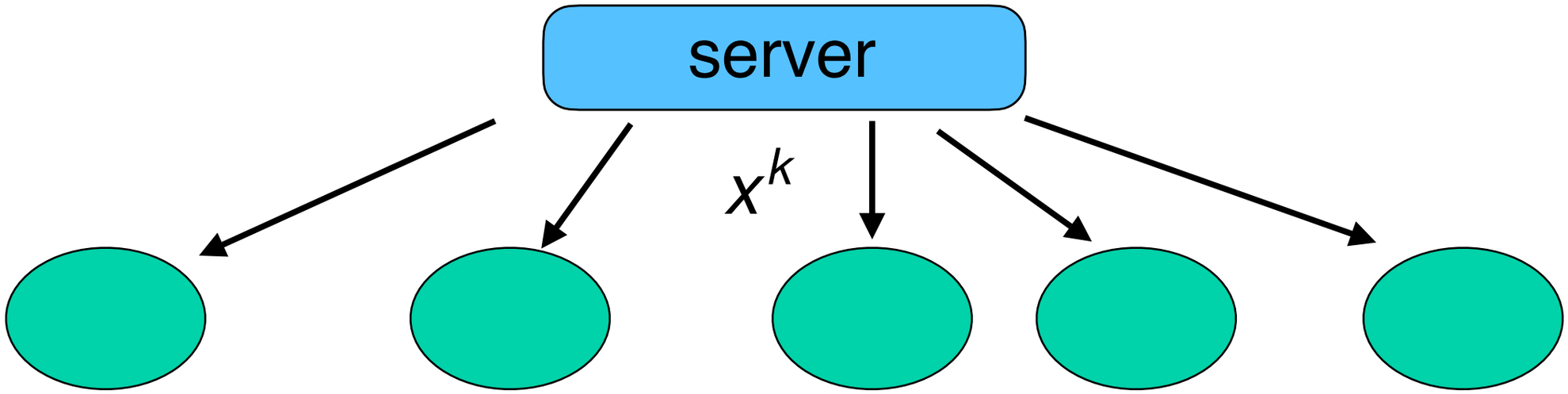
Distributed prox. GD



$$x^{k+1} := \mathrm{prox}_{\gamma R}\left(x^k - \frac{\gamma}{M}\sum_{m=1}^{M}\nabla F_m(x^k)\right)$$

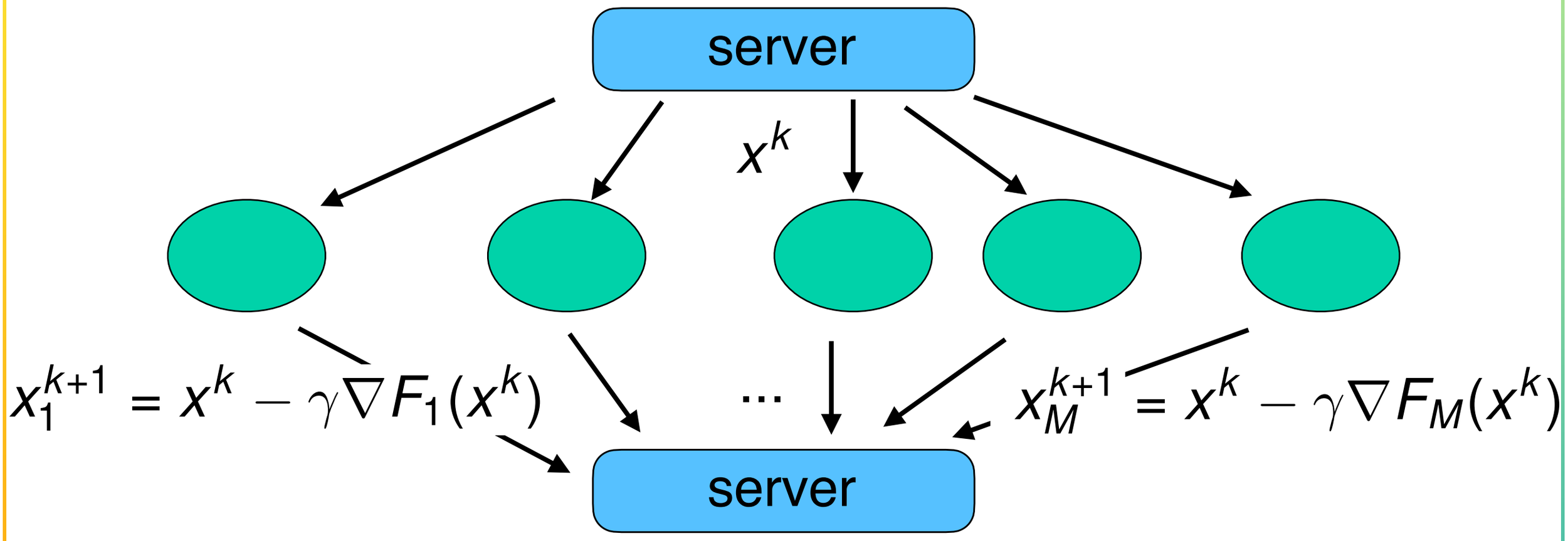
1) local computations

Distributed GD



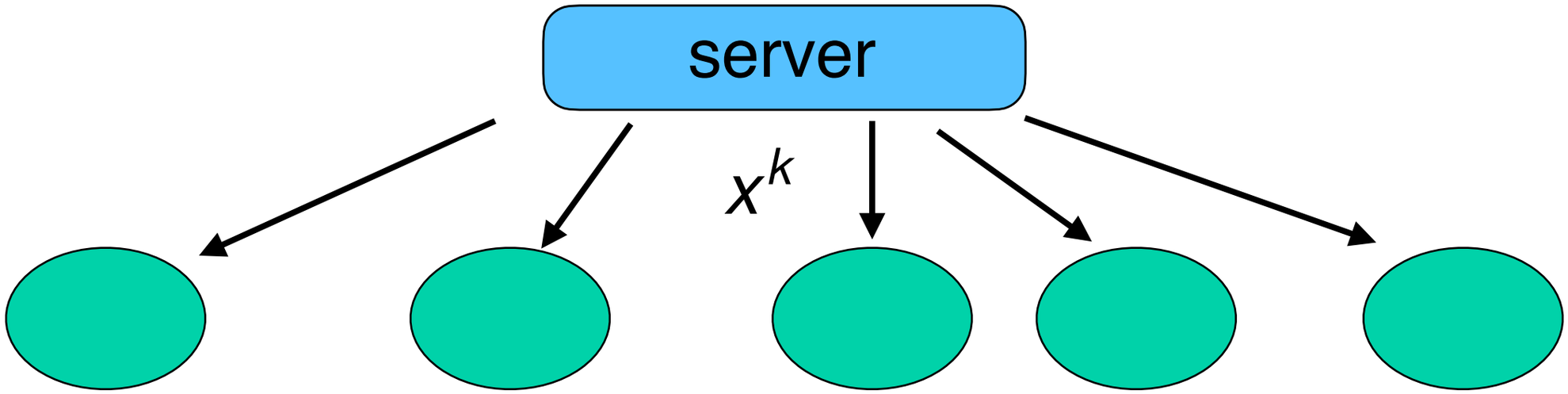
$$x_1^{k+1} = x^k - \gamma \nabla F_1(x^k) \qquad \cdots \qquad x_M^{k+1} = x^k - \gamma \nabla F_M(x^k)$$

Distributed GD



$$x^{k+1} := \mathrm{prox}_{\gamma R}\left(\frac{1}{M}\sum_{m=1}^{M} x_m^{k+1}\right)$$

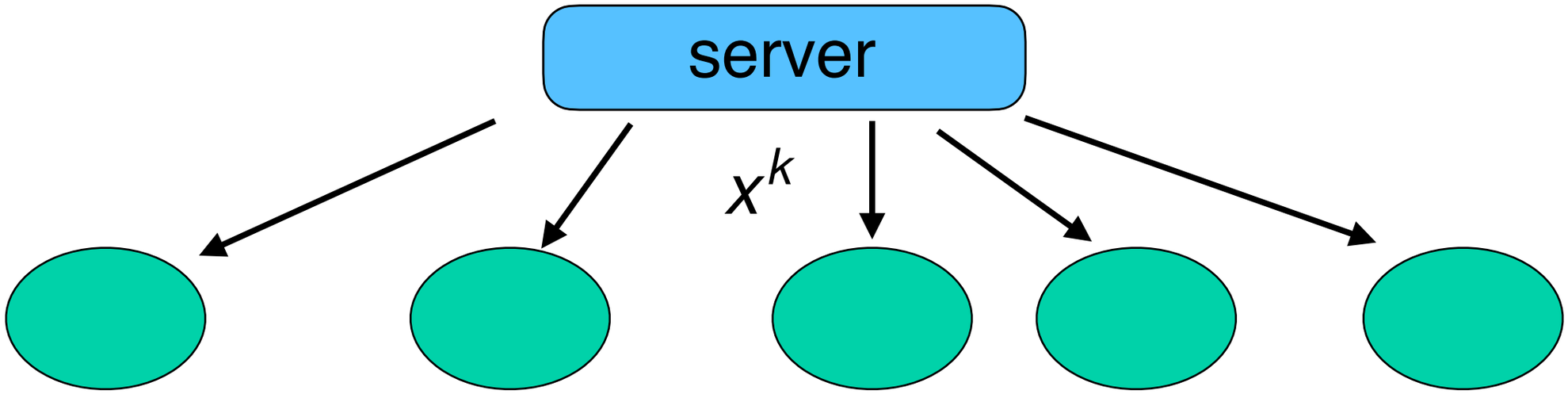
Distributed **Local** GD



$$x_1^{k+1} = x^k - \gamma \nabla F_1(x^k) \qquad \cdots \qquad x_M^{k+1} = x^k - \gamma \nabla F_M(x^k)$$

$$x_1^{k+2} = x_1^{k+1} - \gamma \nabla F_1(x_1^{k+1}) \qquad x_M^{k+2} = x_M^{k+1} - \gamma \nabla F_M(x_M^{k+1})$$

Distributed **Local** GD



$$x_1^{k+1} = x^k - \gamma \nabla F_1(x^k) \qquad \cdots \qquad x_M^{k+1} = x^k - \gamma \nabla F_M(x^k)$$

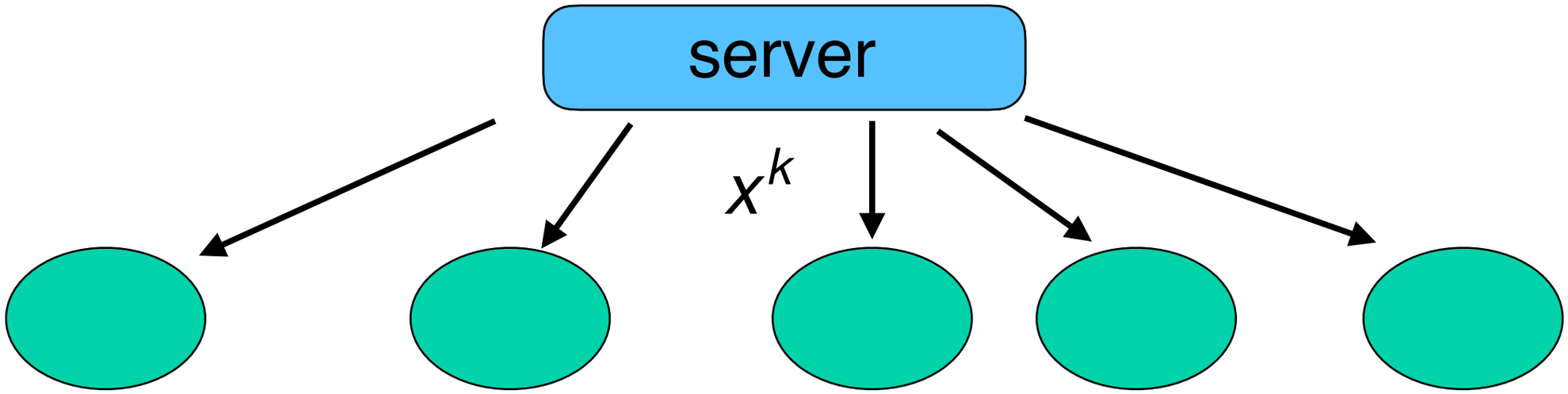
$$x_1^{k+2} = x_1^{k+1} - \gamma \nabla F_1(x_1^{k+1}) \qquad x_M^{k+2} = x_M^{k+1} - \gamma \nabla F_M(x_M^{k+1})$$

$$\cdots \qquad\qquad\qquad\qquad \cdots$$

$$x_1^{k+H} = x_1^{k+H-1} - \gamma \nabla F_1(x_1^{k+H-1}) \qquad x_M^{k+H} = x_M^{k+H-1} - \gamma \nabla F_M(x_M^{k+H-1})$$

$$H \geq 1$$

Distributed **Local** GD

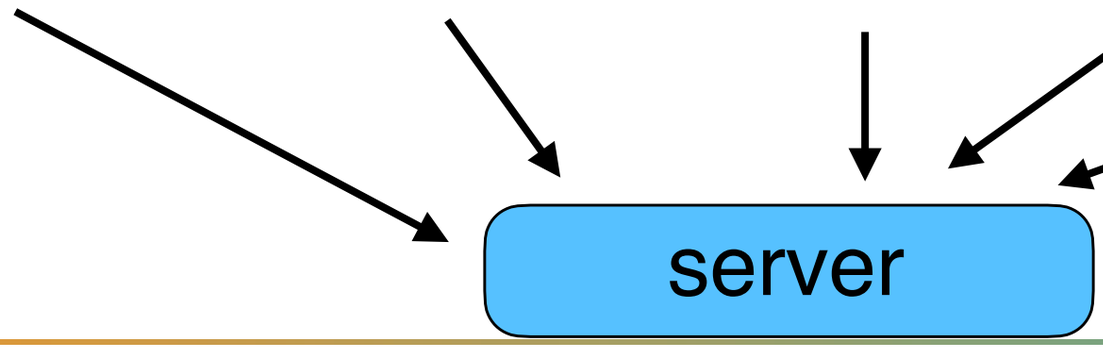


$$x_1^{k+1} = x^k - \gamma \nabla F_1(x^k) \qquad \cdots \qquad x_M^{k+1} = x^k - \gamma \nabla F_M(x^k)$$

$$x_1^{k+2} = x_1^{k+1} - \gamma \nabla F_1(x_1^{k+1}) \qquad x_M^{k+2} = x_M^{k+1} - \gamma \nabla F_M(x_M^{k+1})$$

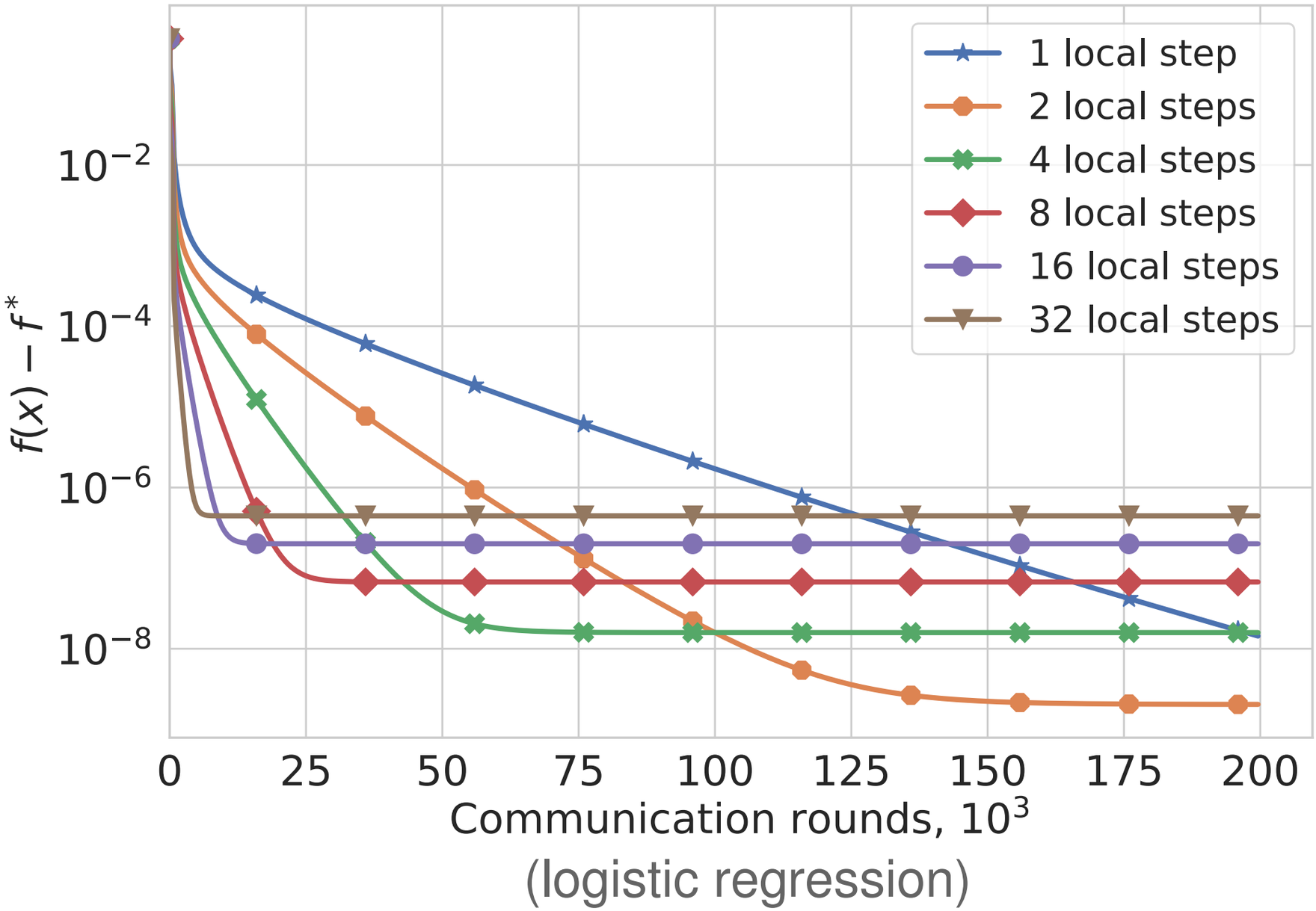
$$\cdots \qquad\qquad\qquad\qquad \cdots$$

$$x_1^{k+H} = x_1^{k+H-1} - \gamma \nabla F_1(x_1^{k+H-1}) \qquad x_M^{k+H} = x_M^{k+H-1} - \gamma \nabla F_M(x_M^{k+H-1})$$



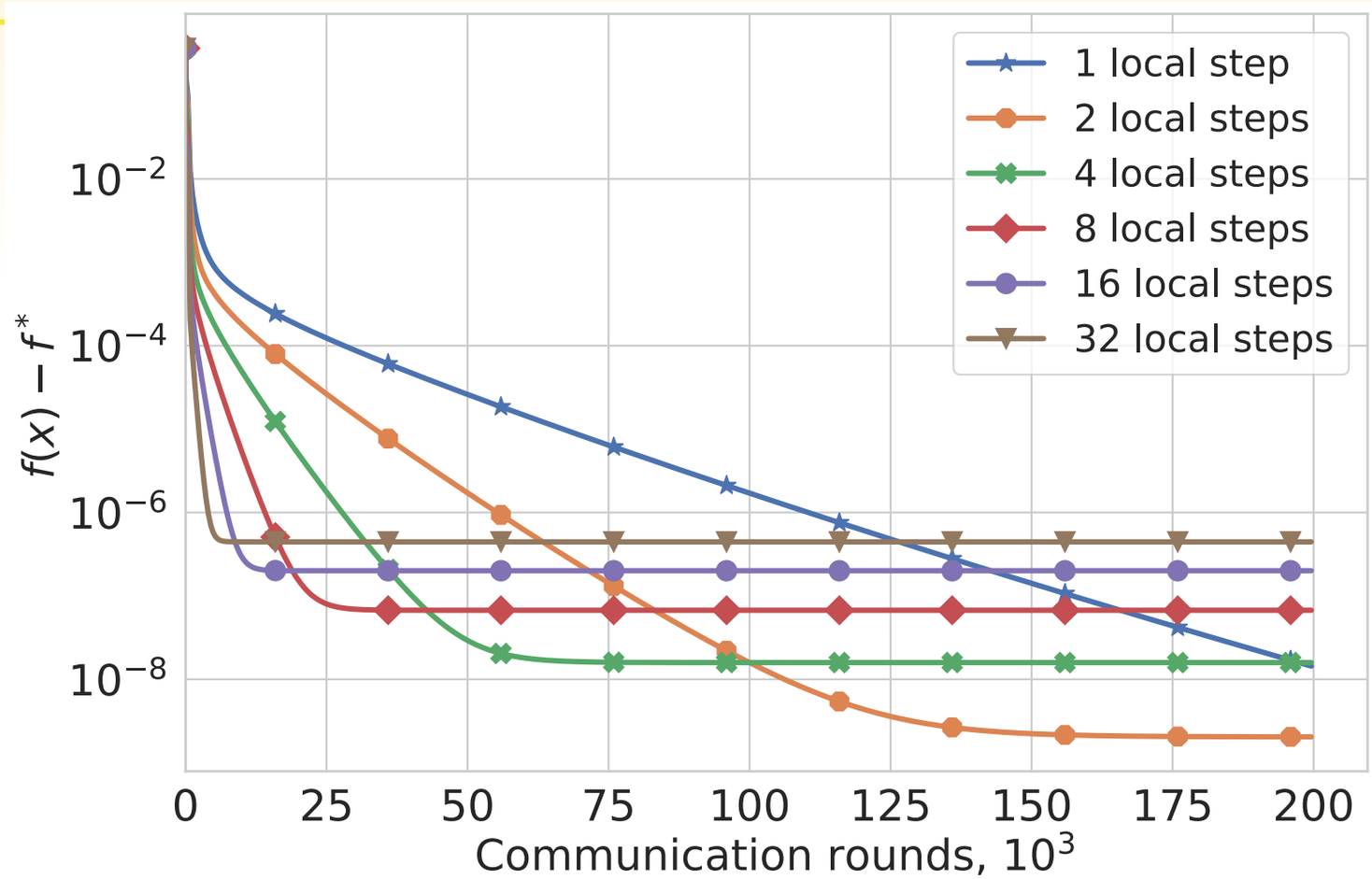
$$x^{k+H} := \mathrm{prox}_{\gamma R}\left(\frac{1}{M} \sum_{m=1}^{M} x_m^{k+H} \right)$$

Local GD: analysis



(logistic regression)

Malinovsky, Kovalev, Gasanov, Condat, Richtárik, “From local SGD to local fixed point methods for federated learning,” ICML 2020



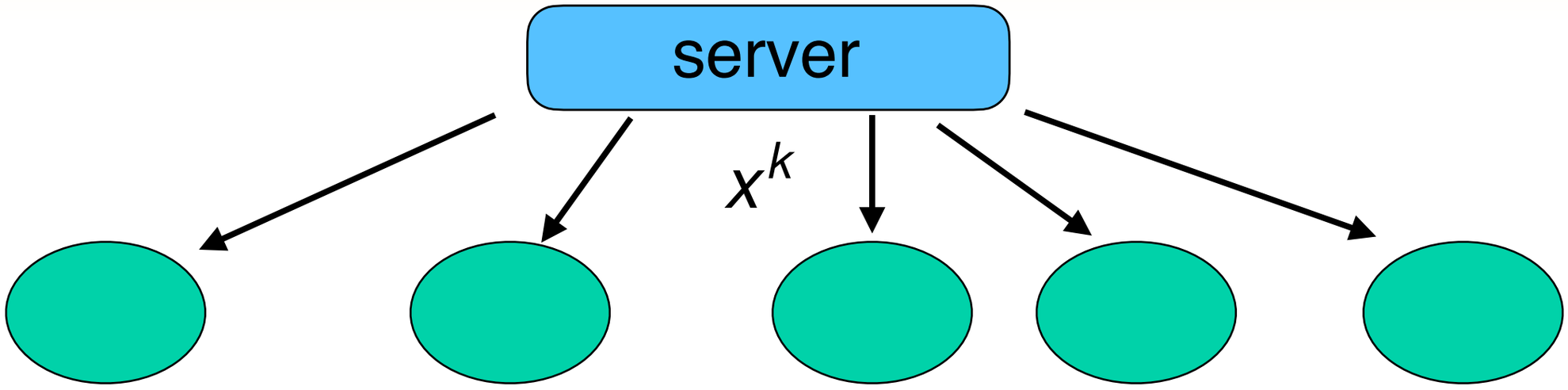
Theorem 2.11 (linear convergence) With $\gamma \in (0, \frac{2}{L+\mu}]$, $(x^{nH})_{n\geq 0}$ converges linearly to $x^\dagger$ with rate $(1-\gamma\mu)^H$ and

$$\|x^\dagger - x^\star\| \leq S,$$

where

$$S = \frac{\xi}{1-\xi}\frac{1-\xi^{H-1}}{1-\xi^H}\frac{1}{M}\sum_{m=1}^{M}\|\nabla F_m(x^\star)\|.$$

Variance-reduced local GD



$$x_1^{k+1} = x^k - \gamma \nabla F_1(x^k) + h_1^k \qquad \cdots \qquad x_M^{k+1} = x^k - \gamma \nabla F_M(x^k) + h_M^k$$

$$x_1^{k+2} = x_1^{k+1} - \gamma \nabla F_1(x_1^{k+1}) + h_1^k \qquad x_M^{k+2} = x_M^{k+1} - \gamma \nabla F_M(x_M^{k+1}) + h_M^k$$

$$\cdots \qquad\qquad\qquad\qquad \cdots$$

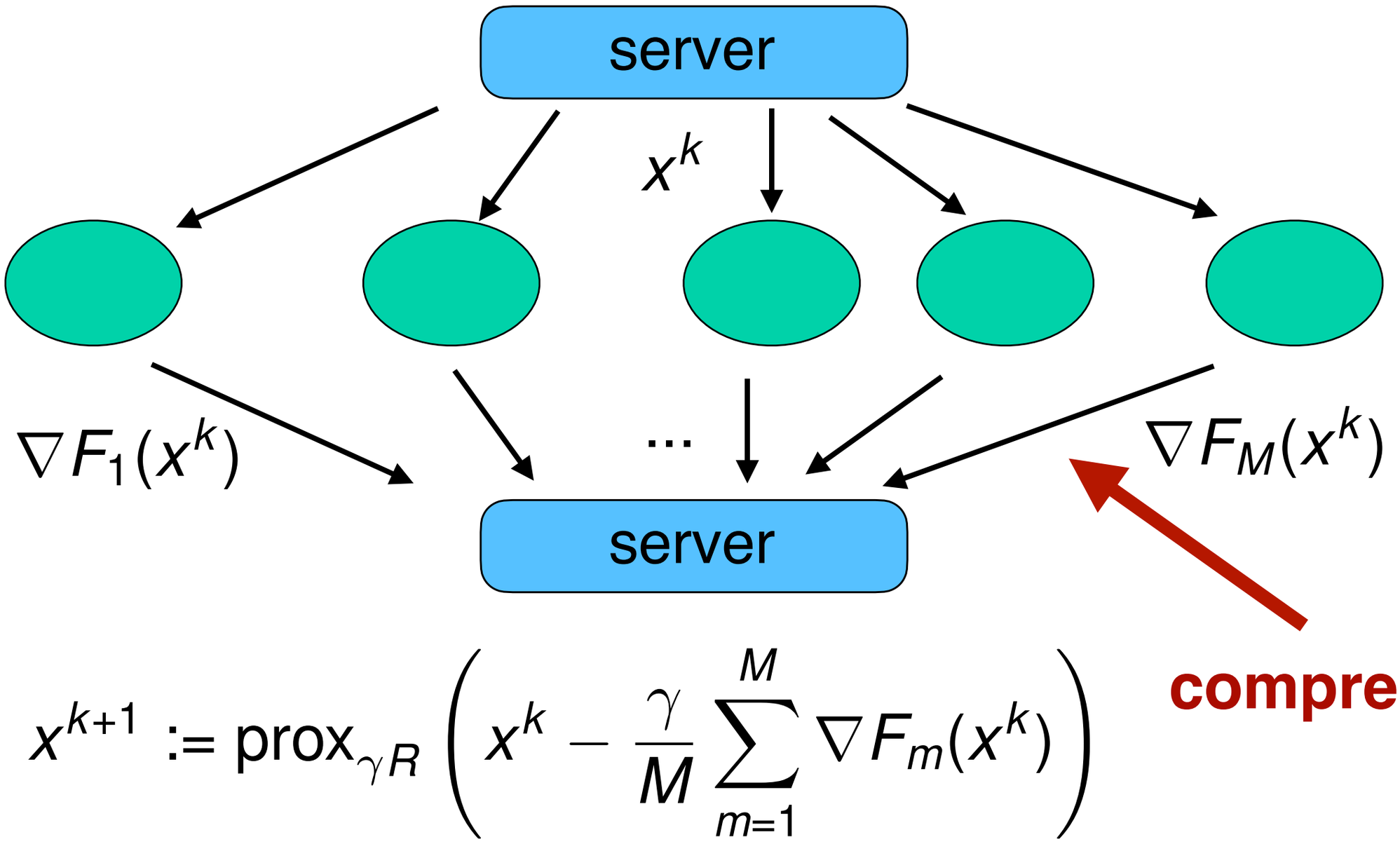
$$x_1^{k+H} = x_1^{k+H-1} - \gamma \nabla F_1(x_1^{k+H-1}) + h_1^k \qquad x_M^{k+H} = x_M^{k+H-1} - \gamma \nabla F_M(x_M^{k+H-1}) + h_M^k$$

Mishchenko, Malinovsky, Stich, Richtárik, “ProxSkip: Yes! Local Gradient Steps Provably Lead to Communication Acceleration!,” ICML 2022

Condat and Richtárik, “RandProx: Primal-Dual Optimization Algorithms with Randomized Proximal Updates,” ICLR 2023

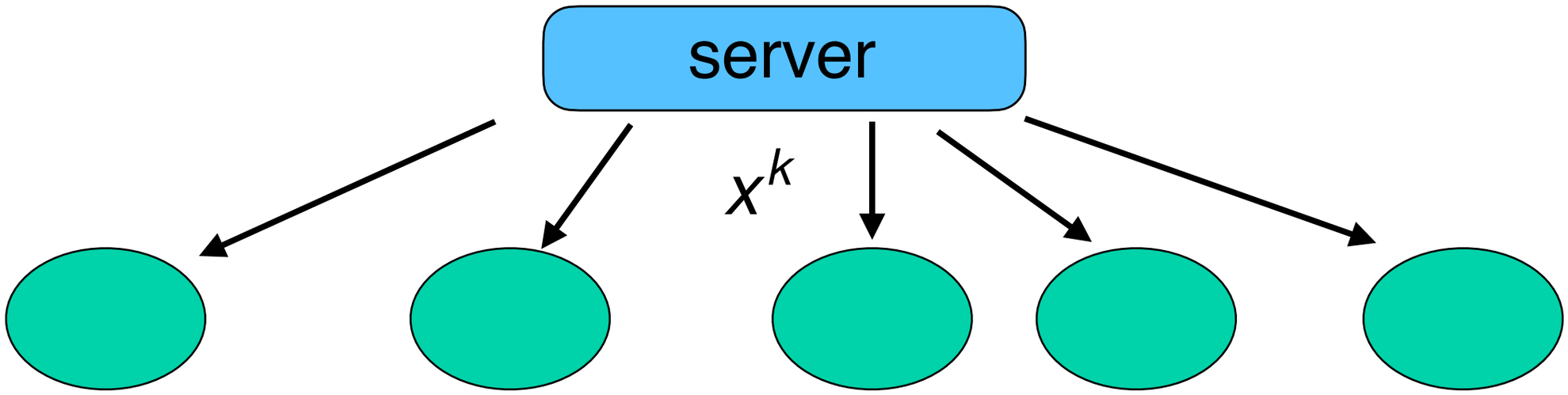
2) compression

Distributed GD



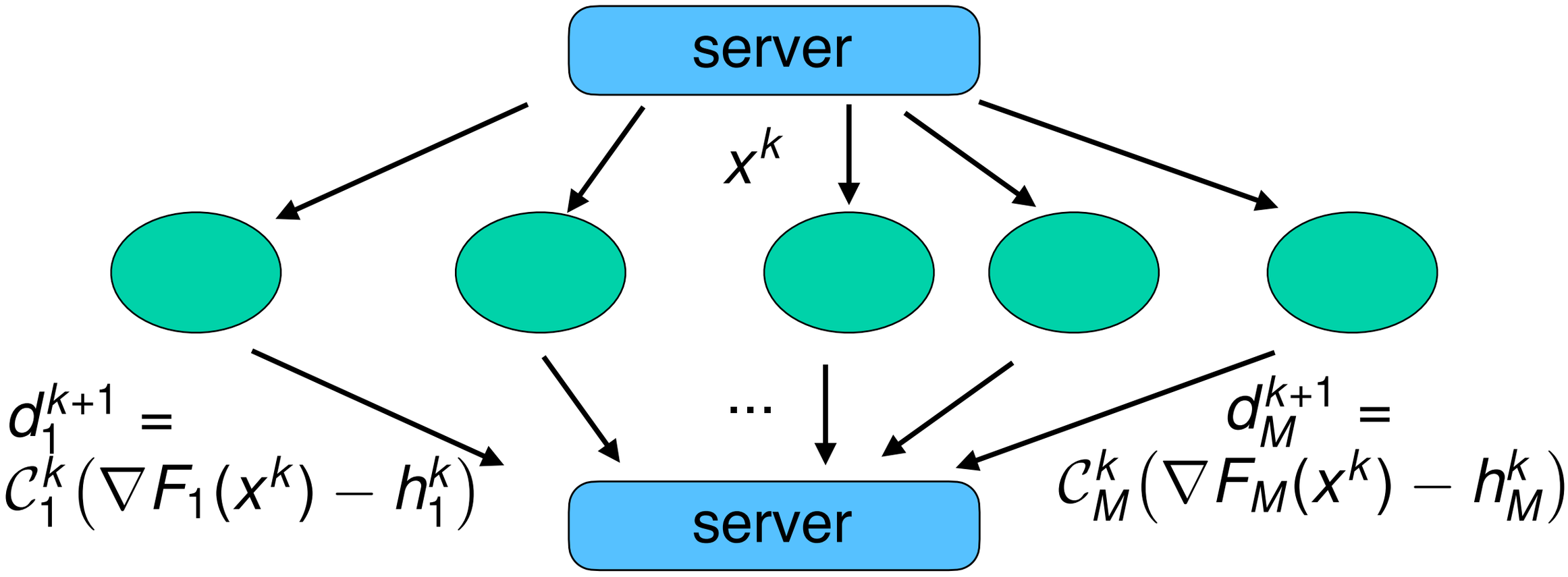
$$x^{k+1} := \mathrm{prox}_{\gamma R}\left(x^k - \frac{\gamma}{M}\sum_{m=1}^{M}\nabla F_m(x^k)\right)$$

Distributed GD with compression

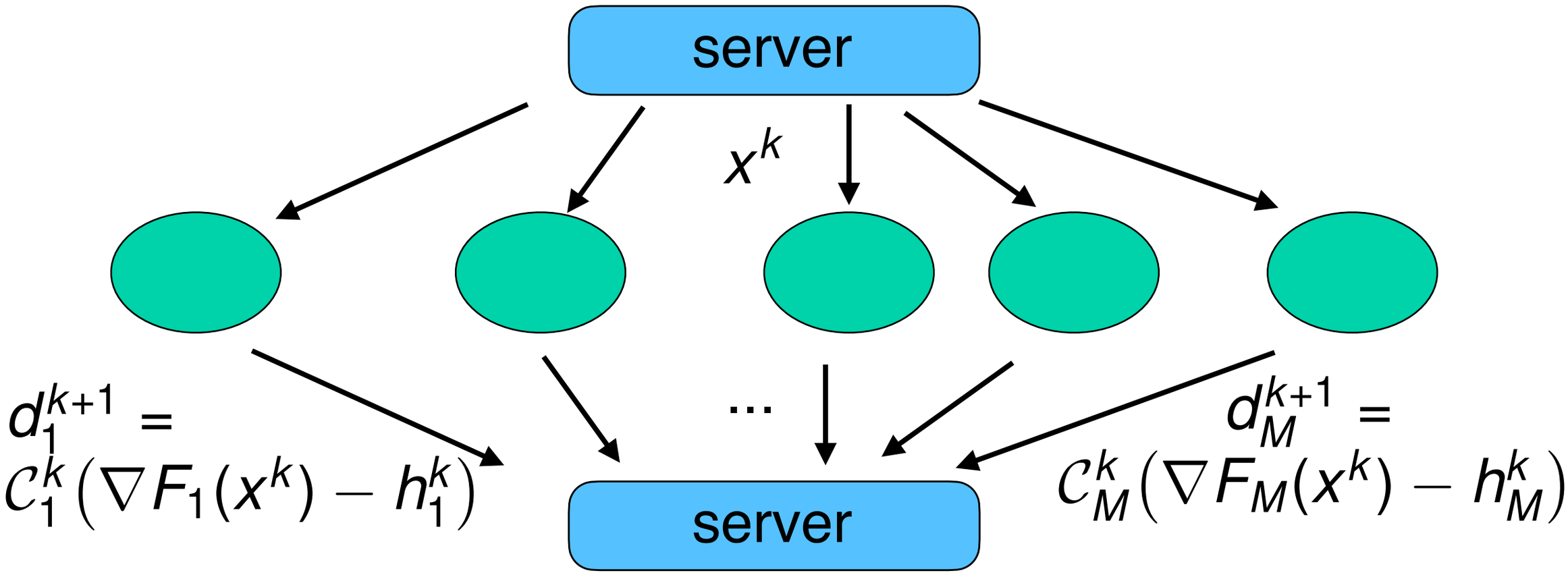


$d_1^{k+1} = \mathcal{C}_1^k\left(\nabla F_1(x^k) - h_1^k\right)$ ... $d_M^{k+1} = \mathcal{C}_M^k\left(\nabla F_M(x^k) - h_M^k\right)$

Distributed GD with compression

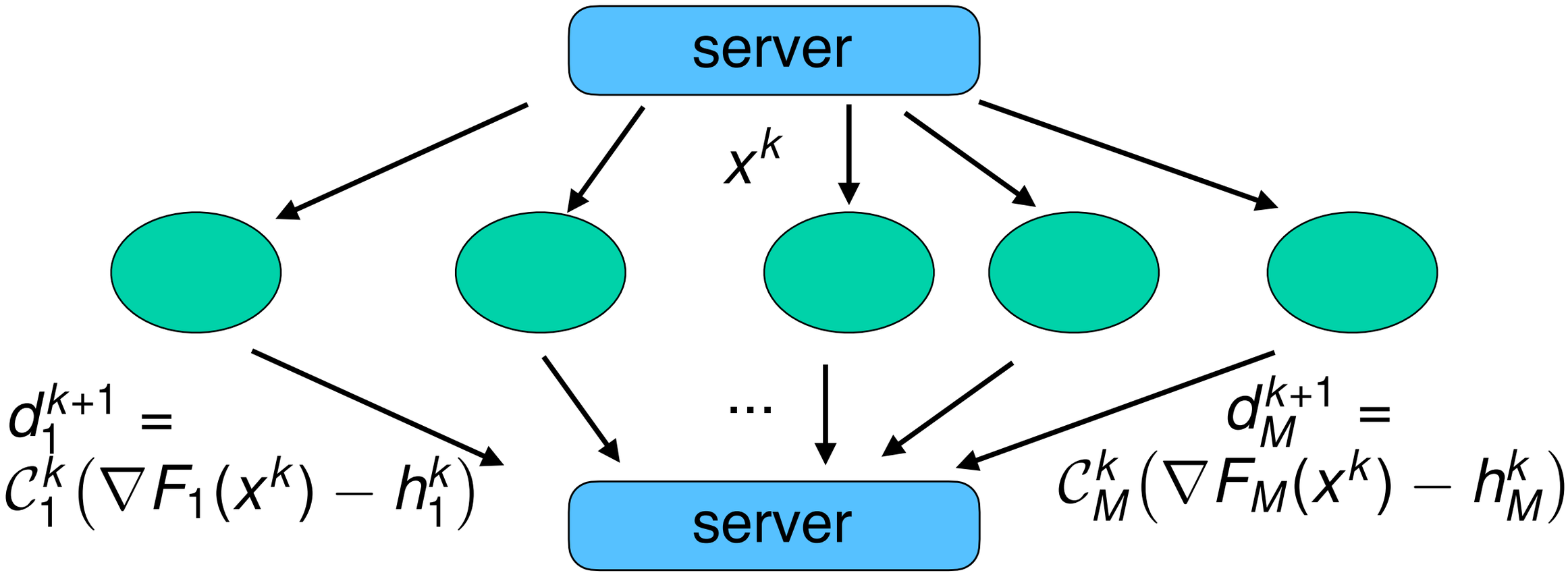


Distributed GD with compression



$$g^k = h^k + \frac{\nu}{M}\sum_{m=1}^{M} d_m^k$$

Distributed GD with compression



$$g^k = h^k + \frac{\nu}{M}\sum_{m=1}^{M} d_m^k$$

$$x^{k+1} := \mathrm{prox}_{\gamma R}\left(x^k - \gamma g^k\right)$$

EF-BV

Algorithm 1 (EF-BV)

input: parameters $\gamma > 0$, $\lambda > 0$, $\nu > 0$,
initial vectors $x^0 \in \mathbb{R}^d$ and $h_m^0 \in \mathbb{R}^d$
$h^0 := \frac{1}{M} \sum_{m=1}^{M} h_m^0$
for $k = 0, 1, \ldots$ **do**
  for $m = 1, \ldots, M$ in parallel **do**
    $d_m^{k+1} := \mathcal{C}_m^k\left(\nabla F_m(x^k) - h_m^k\right)$
    $h_m^{k+1} := h_m^k + \lambda d_m^{k+1}$
  end for
  // at master:
  $d^{k+1} := \frac{1}{M} \sum_{m=1}^{M} d_m^{k+1}$
  $x^{k+1} := \mathrm{prox}_{\gamma R}\left(x^k - \gamma(h^k + \nu d^{k+1})\right)$
  $h^{k+1} := h^k + \lambda d^{k+1}$
end for

Condat, Yi, Richtárik, "EF-BV: A unified theory of error feedback and variance reduction for biased and unbiased compression in distributed optimization," NeurIPS 2022

EF-BV: SGD-type algorithm

$$x^{k+1} := \mathrm{prox}_{\gamma R}\left(x^k - \frac{\gamma}{M}\sum_{m=1}^{M} g_m^k\right)$$

with stochastic gradients

$$g_m^k = h_m^k + \nu \mathcal{C}_m^k\left(\nabla F_m(x^k) - h_m^k\right) \approx \nabla F_m(x^k)$$

EF-BV: SGD-type algorithm

$$x^{k+1} := \mathrm{prox}_{\gamma R}\left(x^k - \frac{\gamma}{M}\sum_{m=1}^{M} g_m^k\right)$$

with stochastic gradients

$$g_m^k = h_m^k + \nu \mathcal{C}_m^k\left(\nabla F_m(x^k) - h_m^k\right) \approx \nabla F_m(x^k)$$

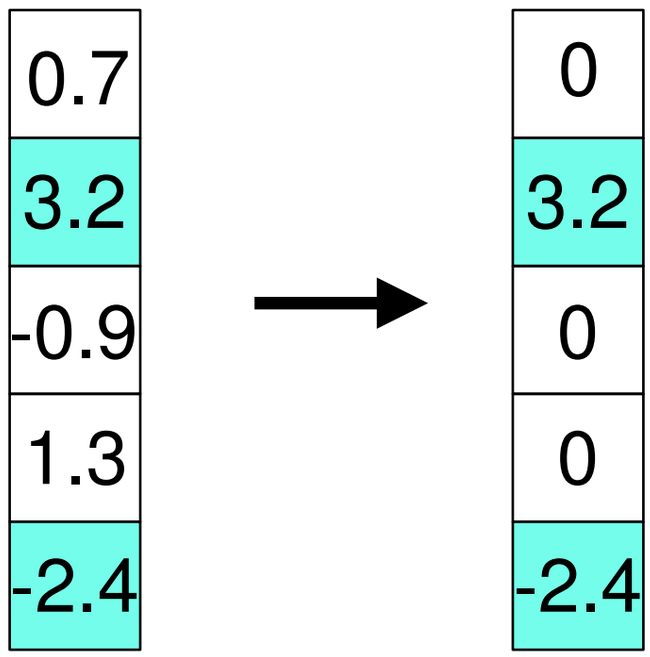
$\nu = 1$ and unbiased $\mathcal{C}_m^k \quad \Rightarrow \quad \mathbb{E}\left[g_m^k\right] = \nabla F_m(x^k)$

☞ DIANA [Mishchenko et al. 2019] generalized in:

Condat and Richtárik, “MURANA: A Generic Framework for Stochastic Variance-Reduced Optimization,” MSML 2022

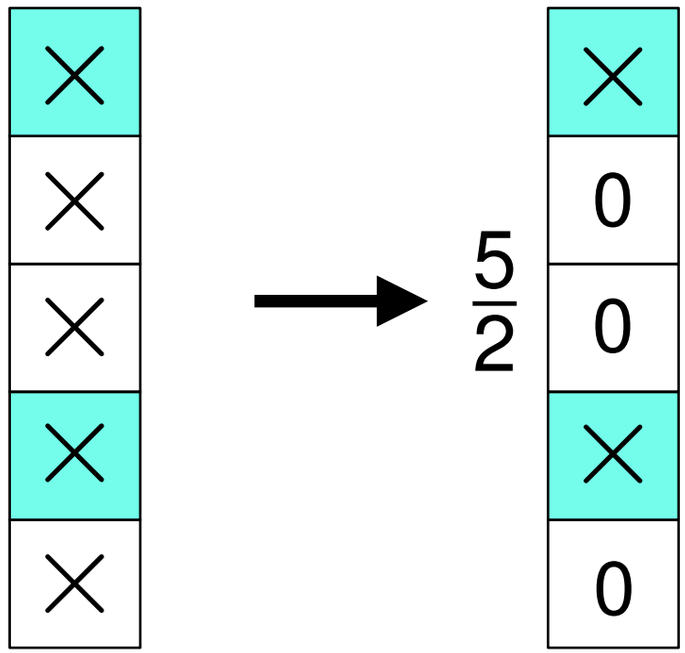
Biased compressors: example

- `top-`s: s elements with largest abs. value kept, other ones set to 0.



Unbiased compressors: examples

- `rand-`s: s elements out of d chosen unif. at random and scaled by $\frac{d}{s}$, other ones set to 0.



Unbiased compressors: examples

- `rand-`s: s elements out of d chosen unif. at random and scaled by $\frac{d}{s}$, other ones set to 0.

- quantization of the real values:

  Example: 0.2 represented by

$$\begin{cases} 0 \text{ with probability } \frac{4}{5} \\ 1 \text{ with probability } \frac{1}{5} \end{cases}$$

Unbiased compressors: examples

- `rand-`s: s elements out of d chosen unif. at random and scaled by $\frac{d}{s}$, other ones set to 0.

- quantization of the real values:

  Example: 0.2 represented by

$$\begin{cases} 0 \text{ with probability } \frac{4}{5} \\ 1 \text{ with probability } \frac{1}{5} \end{cases}$$

Albasyoni, Safaryan, Condat, Richtárik, “Optimal Gradient Compression for Distributed and Federated Learning,” 2020

Unbiased compressors: examples

- probabilistic activation:

$$\mathcal{C}_m^k(v_m) = \begin{cases} \frac{1}{p} v_m \text{ with probability } p \\ 0 \text{ with probability } 1-p \end{cases}$$

Unbiased compressors: examples

- probabilistic activation:

$$\mathcal{C}_m^k(v_m) = \begin{cases} \frac{1}{p} v_m \text{ with probability } p \\ 0 \text{ with probability } 1-p \end{cases}$$

- partial participation:

$$\mathcal{C}_m^k(v_m) = \begin{cases} \frac{M}{N} v_m \text{ if } m \in \Omega^k \\ 0 \text{ else} \end{cases}$$

for $\Omega^k \subset \{1, \dots, M\}$ of size N chosen uniformly at random

Unbiased compressors

- $\mathbb{E}\left[\mathcal{C}_m^k(v)\right] = v$

$\exists \omega \geq 0$ such that $\forall v \in \mathbb{R}^d$,

- $\mathbb{E}\left[\|\mathcal{C}_m^k(v) - v\|^2\right] \leq \omega \|v\|^2$

DIANA: convergence

Theorem [MURANA]. DIANA with independent $\mathcal{C}_m^k$, $\lambda = \frac{1}{1+\omega}$ and

$$0 < \gamma < \frac{2}{L} \frac{1}{1 + 4\frac{\omega}{M}} :$$

Choose $b > 1$ s.t. $\eta \coloneqq 1 - \gamma \left(\frac{2}{L} \frac{1}{1+(1+b)^2 \frac{\omega}{M}} \right)^{-1} \in (0, 1)$.

Define the Lyapunov function, for every $k \geq 0$,

$$\Psi^k \coloneqq \left\| x^k - x^\star \right\|^2 + (b^2 + b)\gamma^2 \frac{\omega(1+\omega)}{M^2} \sum_{m=1}^{M} \left\| h_m^k - h_m^\star \right\|^2 .$$

Then, for every $k \geq 0$, we have $\mathbb{E}\left[\Psi^k\right] \leq c^k \Psi^0$, where

$$c \coloneqq 1 - \min \left\{ 2\gamma\eta\mu, \frac{1 - b^{-2}}{1+\omega} \right\} < 1.$$

DIANA: convergence

Theorem [MURANA]. DIANA with independent $\mathcal{C}_m^k$, $\lambda = \frac{1}{1+\omega}$ and

$$0 < \gamma < \frac{2}{L} \frac{1}{1 + 4\frac{\omega}{M}} :$$

☞ iteration complexity to reach ϵ-accuracy:

$$\mathcal{O}\left(\left(\frac{L}{\mu}\left(1 + \frac{\omega}{M}\right) + \omega\right) \log \epsilon^{-1}\right)$$

DIANA: convergence

Theorem [MURANA]. DIANA with independent $\mathcal{C}_m^k$, $\lambda = \frac{1}{1+\omega}$ and

$$0 < \gamma < \frac{2}{L} \frac{1}{1 + 4\frac{\omega}{M}} :$$

typically, the communication complexity can be reduced from

$$\mathcal{O}\left(d\frac{L}{\mu}\log\epsilon^{-1}\right) \quad \text{to} \quad \mathcal{O}\left(\left(\frac{L}{\mu}+d\right)\log\epsilon^{-1}\right)$$

combining local training & compression

TAMUNA

Algorithm 1 (TAMUNA)

for $r = 0, 1, \ldots$ (rounds) **do**
  choose a subset $\Omega^{(r)}$ of size c unif. at random
  choose the number of local steps $\mathcal{L}^{(r)} \geq 1$
  for clients $m \in \Omega^{(r)}$, in parallel, **do**
    $x_m^{(r,0)} := \bar{x}^{(r)}$ (received from the server)
    for $\ell = 0, \ldots, \mathcal{L}^{(r)} - 1$ (local steps) **do**
      $x_m^{(r,\ell+1)} := x_m^{(r,\ell)} - \gamma \nabla F_m\left(x_m^{(r,\ell)}\right) + \gamma h_m^{(r)}$
    end for
    send $\mathcal{C}_m^{(r)}\left(x_m^{(r,\mathcal{L}^{(r)})}\right)$ to the server
  end for
  $\bar{x}^{(r+1)} := \frac{1}{s} \sum_{i \in \Omega^{(r)}} \mathcal{C}_m^{(r)}\left(x_m^{(r,\mathcal{L}^{(r)})}\right)$ (aggregation)
  for clients $m \in \Omega^{(r)}$, in parallel, **do**
    $h_m^{(r+1)} := h_m^{(r)} + \frac{\eta}{\gamma}\left(\mathcal{C}_m^{(r)}\left(\bar{x}^{(r+1)}\right) - \mathcal{C}_m^{(r)}\left(x_m^{(r,\mathcal{L}^{(r)})}\right)\right)$
  end for
  for clients $m \notin \Omega^{(r)}$, in parallel, **do**
    $h_m^{(r+1)} := h_m^{(r)}$ (the client is idle)
  end for
end for

Condat, Agarský, Malinovsky, Richtárik, "TAMUNA: Doubly accelerated federated learning with local training, compression, and partial participation", preprint, 2023

TAMUNA

Uplink communication complexity with $c \leq M$ participating clients:

$$\mathcal{O}\left(\left(\sqrt{d}\sqrt{\frac{L}{\mu}}\sqrt{\frac{M}{c}} + d\sqrt{\frac{L}{\mu}}\frac{\sqrt{M}}{c} + d\frac{M}{c}\right)\log\epsilon^{-1}\right)$$

TAMUNA

Uplink communication complexity with $c \leq M$ participating clients:

$$\mathcal{O}\left(\left(\sqrt{d}\sqrt{\frac{L}{\mu}}\sqrt{\frac{M}{c}} + d\sqrt{\frac{L}{\mu}\frac{\sqrt{M}}{c}} + d\frac{M}{c}\right)\log\epsilon^{-1}\right)$$

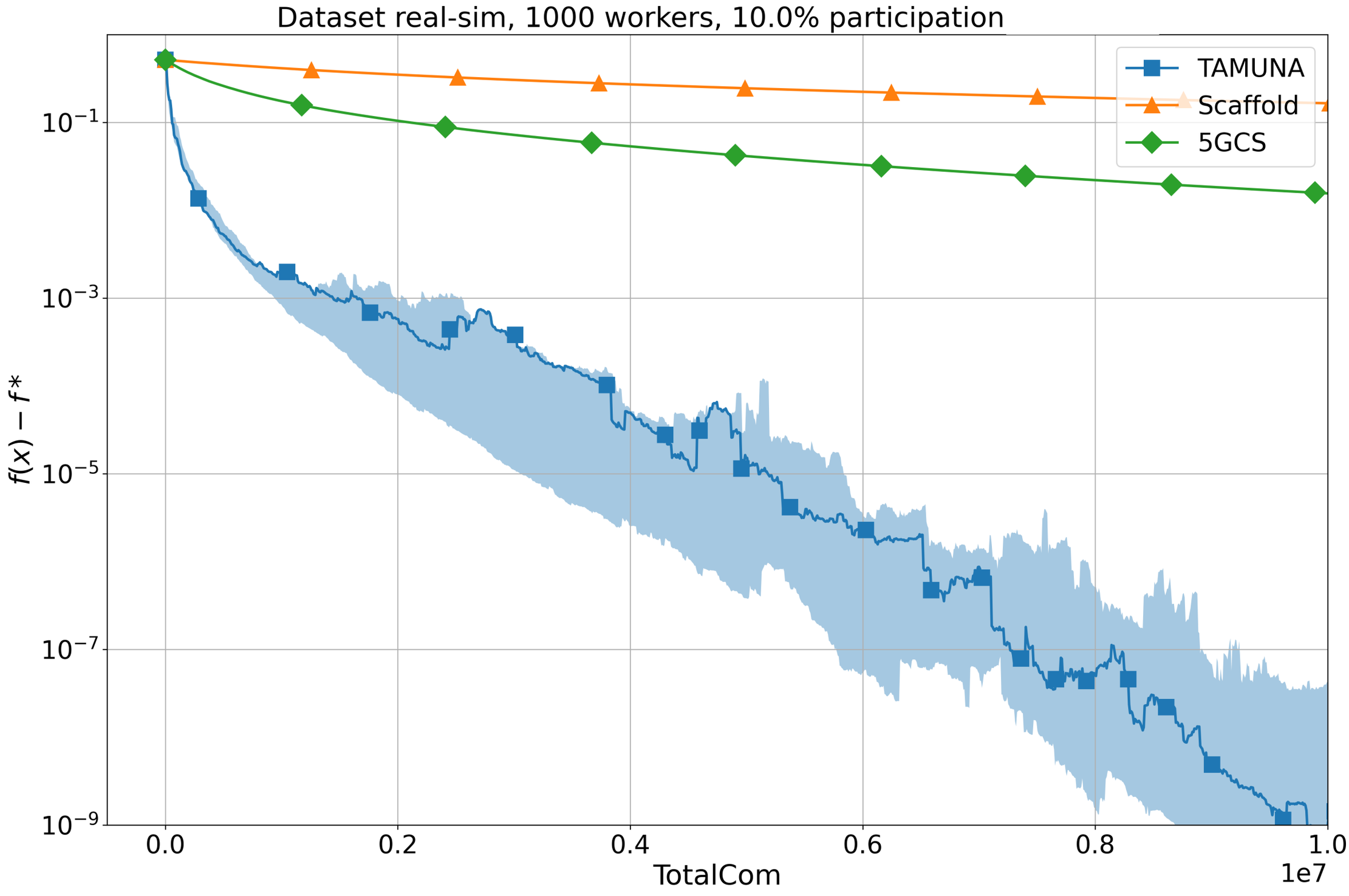
TAMUNA

Uplink communication complexity with $c \leq M$ participating clients:

$$\mathcal{O}\left(\left(\sqrt{d}\sqrt{\frac{L}{\mu}}\sqrt{\frac{M}{c}} + d\sqrt{\frac{L}{\mu}\frac{\sqrt{M}}{c}} + d\frac{M}{c}\right)\log\epsilon^{-1}\right)$$

Note: $\sqrt{d}\sqrt{\frac{L}{\mu}}$ better than $d\sqrt{\frac{L}{\mu}}$ (Proxskip) and $d + \frac{L}{\mu}$ (DIANA)

Experiment: logistic regression



Conclusion

To reduce communication:

1) **local computations**: communicate less frequently.

2) **compression**: communicate compressed vectors.

Combining the 2 ideas + designing randomized primal-dual algorithms: work in progress!