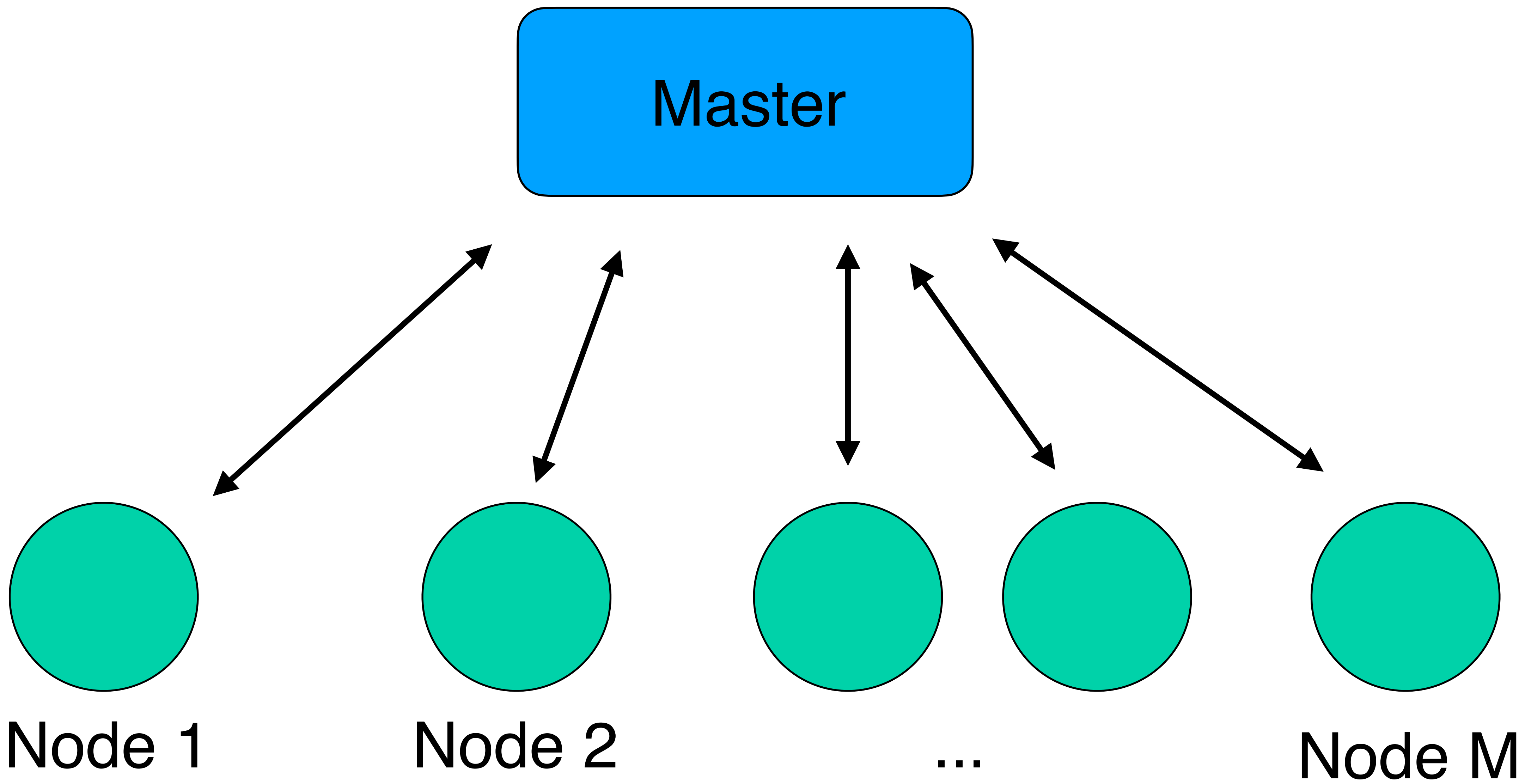
# Communication-efficient distributed optimization algorithms

**Laurent Condat,** Peter Richtárik

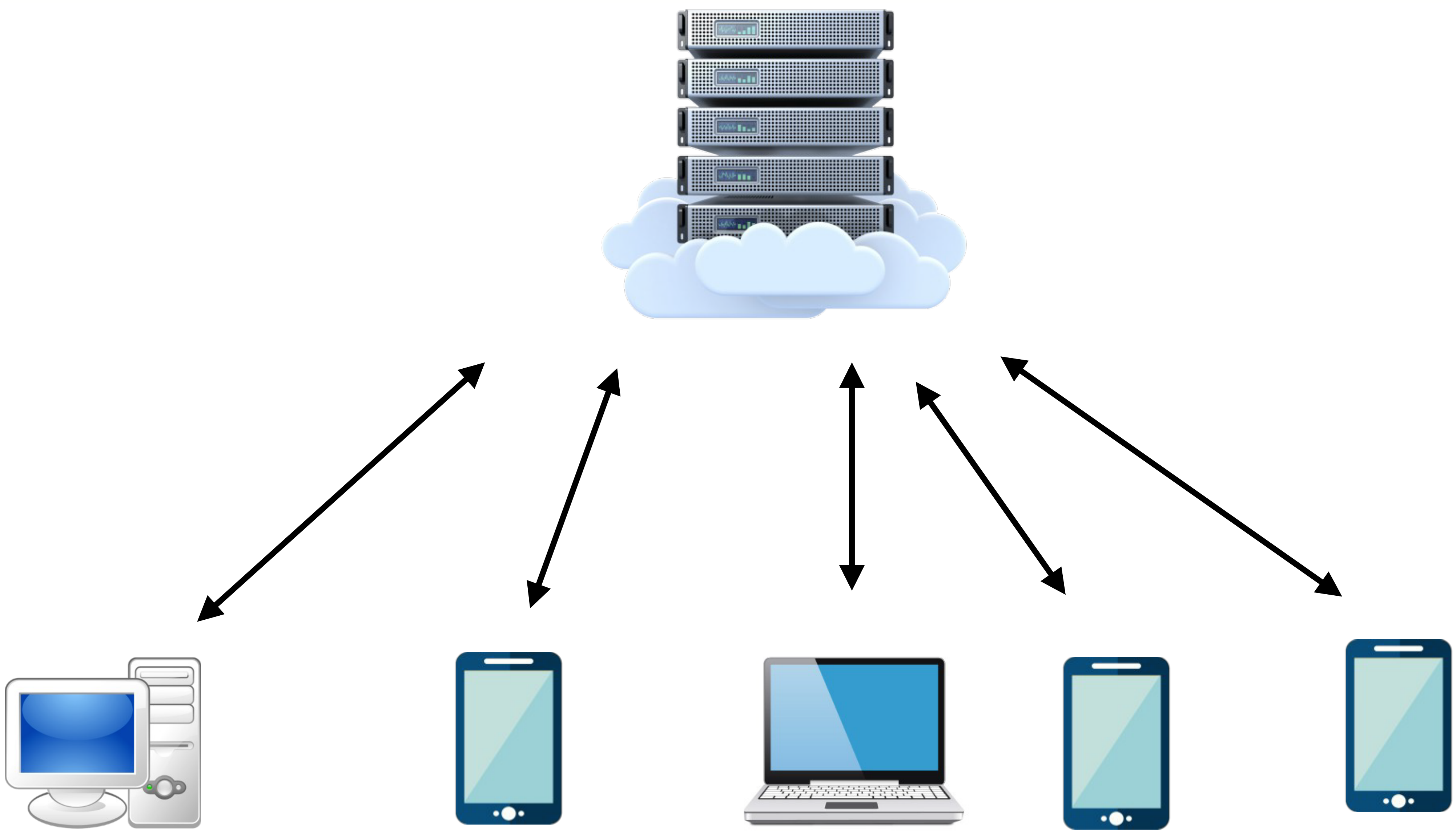
King Abdullah University of Science and Technology
(KAUST)
Thuwal, Saudi Arabia

4th IMA Conf. Mathematical
Challenges of Big Data, Sept. 2022

# Distributed computing



# Federated learning



# Federated learning

# Convex optimization problem

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

- every function $F_m$ is convex and $L$-smooth, for some $L > 0$, and $\mu$-strongly convex, for some $\mu > 0$, i.e. $F - \frac{\mu}{2}\|\cdot\|^2$ is convex.

# Convex optimization problem

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

- every function $F_m$ is convex and $L$-smooth, for some $L > 0$, and $\mu$-strongly convex, for some $\mu > 0$, i.e. $F - \frac{\mu}{2}\|\cdot\|^2$ is convex.
- $R : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$ is a proper, closed, convex function whose proximity operator

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

is easy to compute

# Convex optimization problem

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

**Prox-GD**:

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

# Convex optimization problem

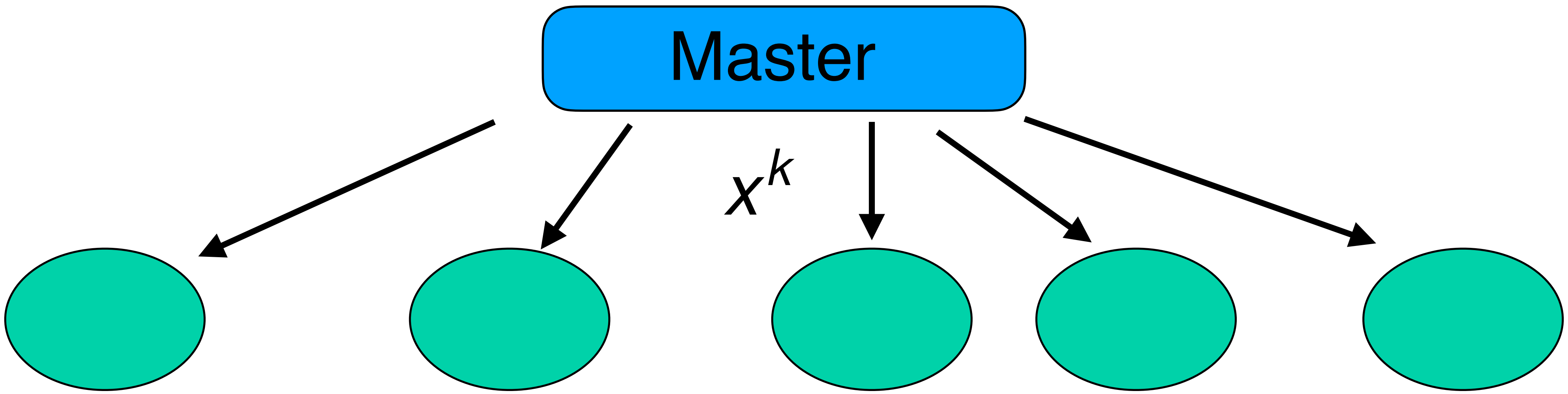
$$\operatorname*{minimize}_{x \in \mathbb{R}^d} \; R(x) + \frac{1}{M} \sum_{m=1}^{M} F_m(x)$$

**Prox-GD**:

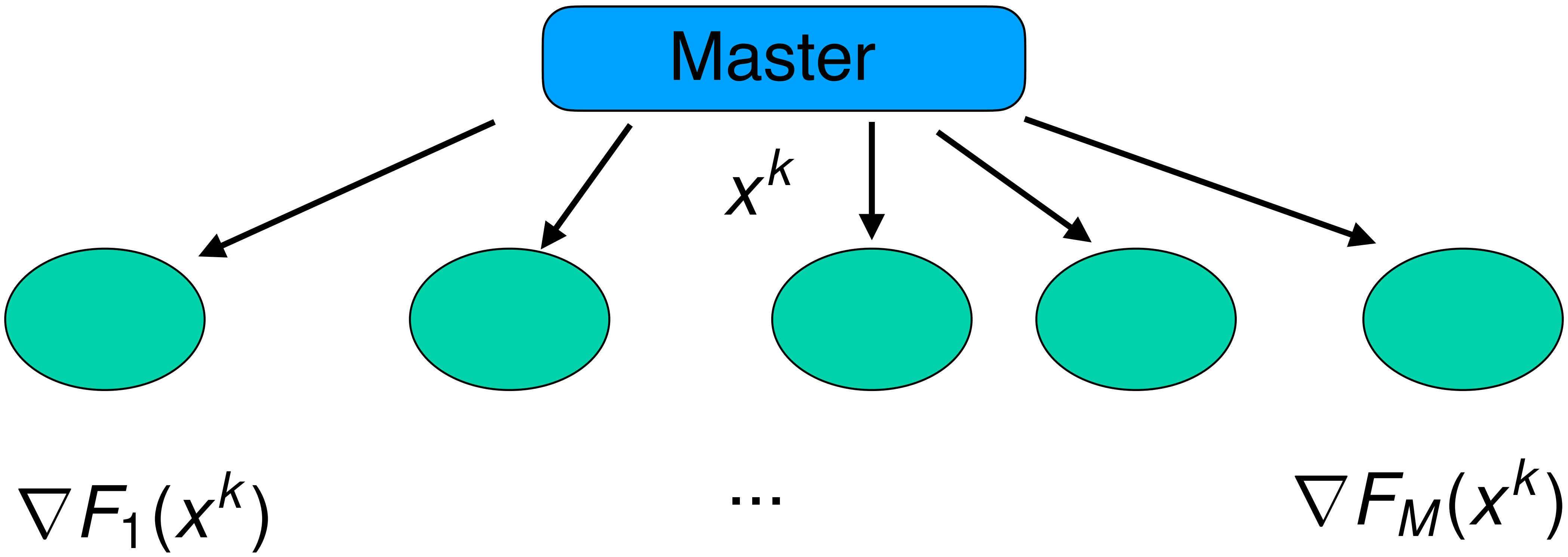
$$x^{k+1} := \operatorname{prox}_{\gamma R}\left( x^k - \frac{\gamma}{M} \sum_{m=1}^{M} \nabla F_m(x^k) \right)$$

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

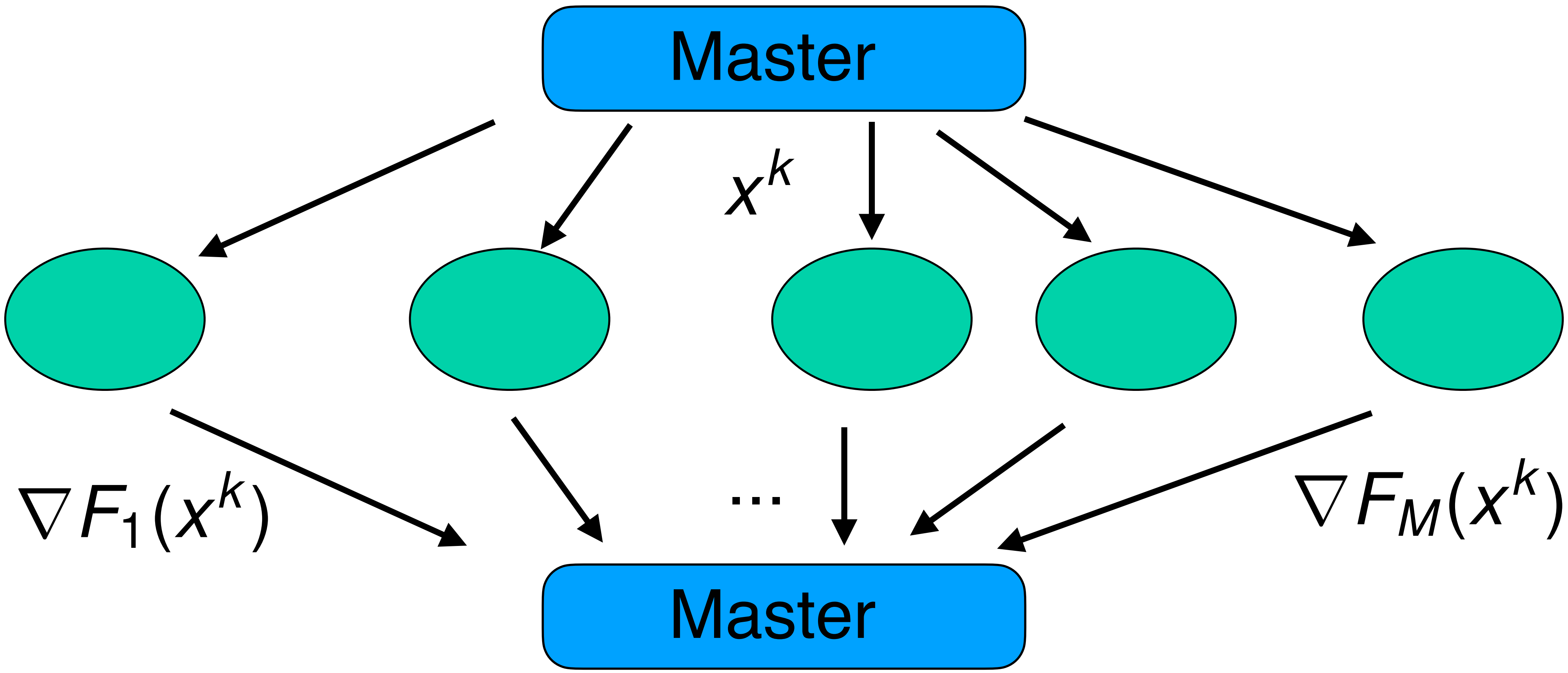
# Distributed prox. GD



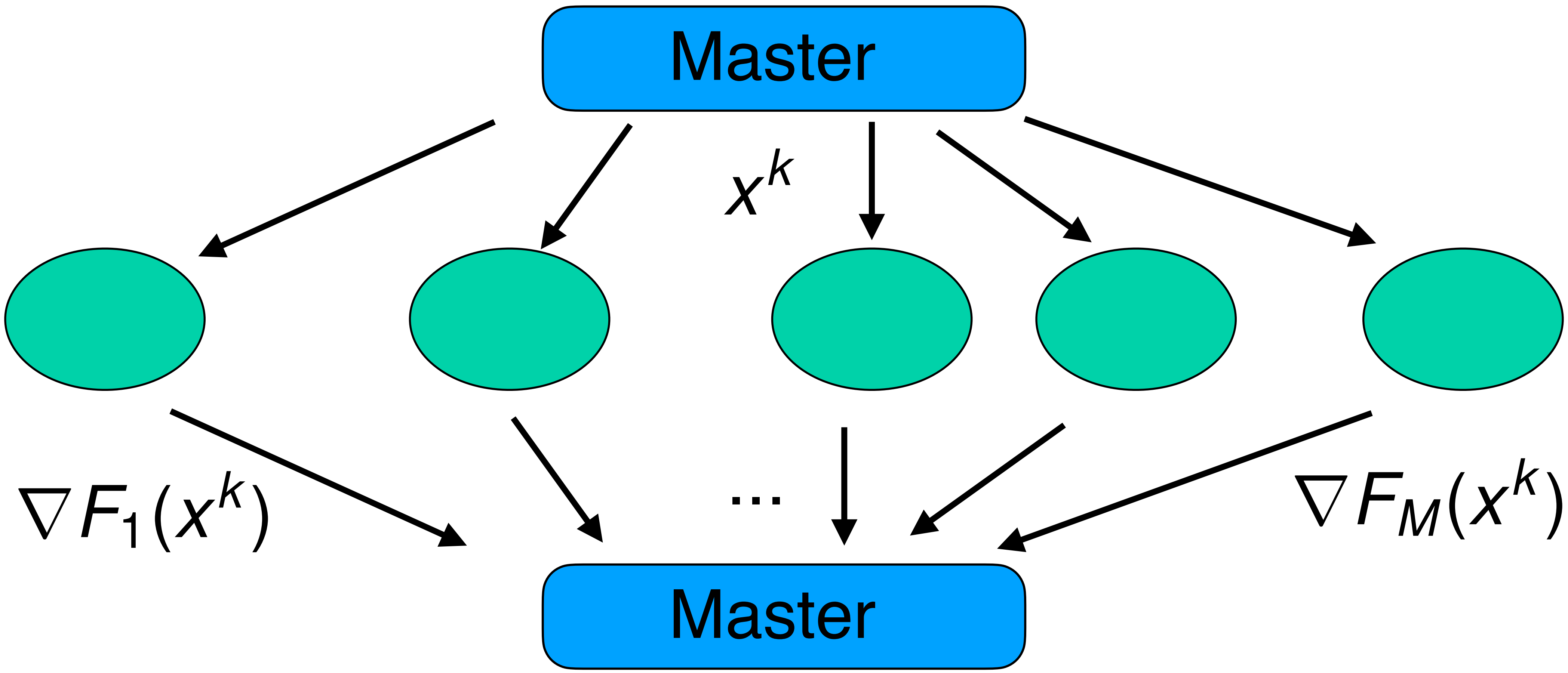
# Distributed prox. GD



# Distributed prox. GD

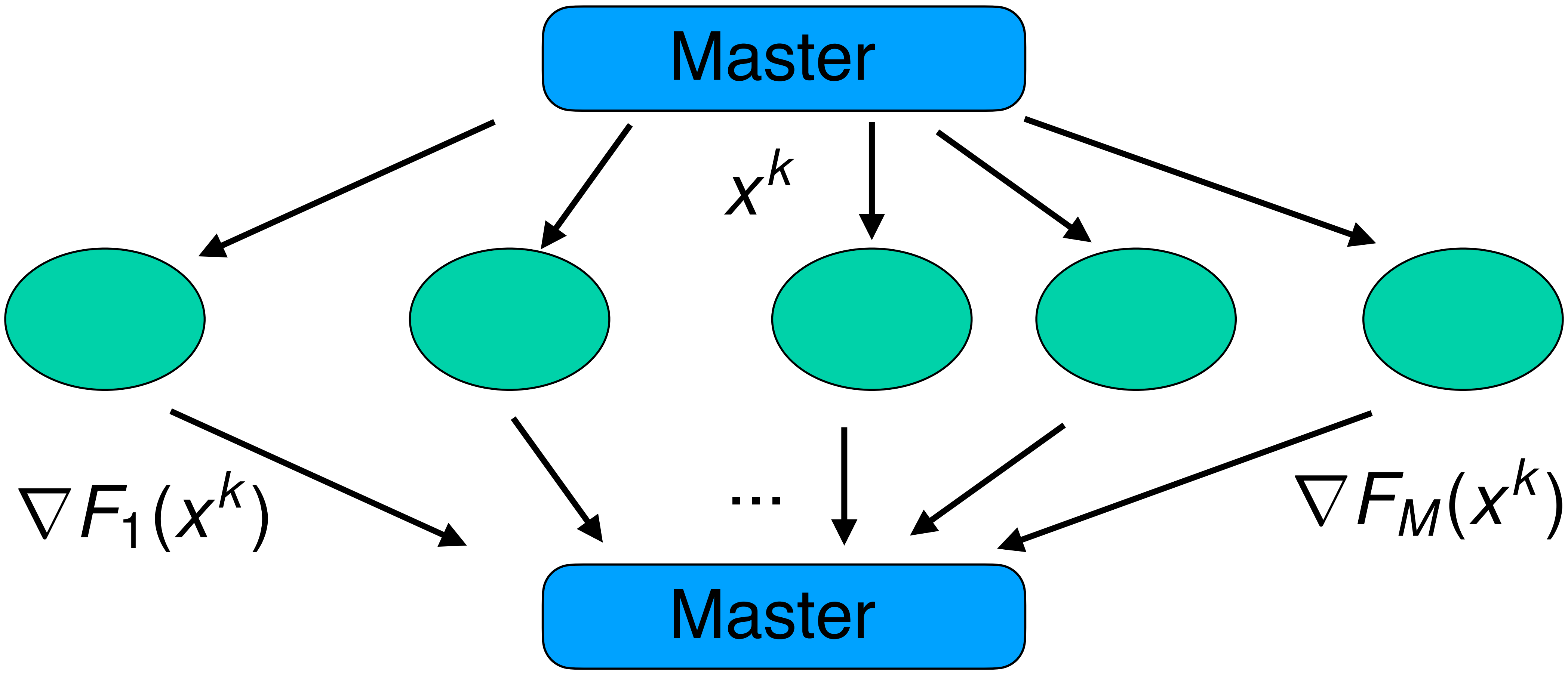


# Distributed prox. GD



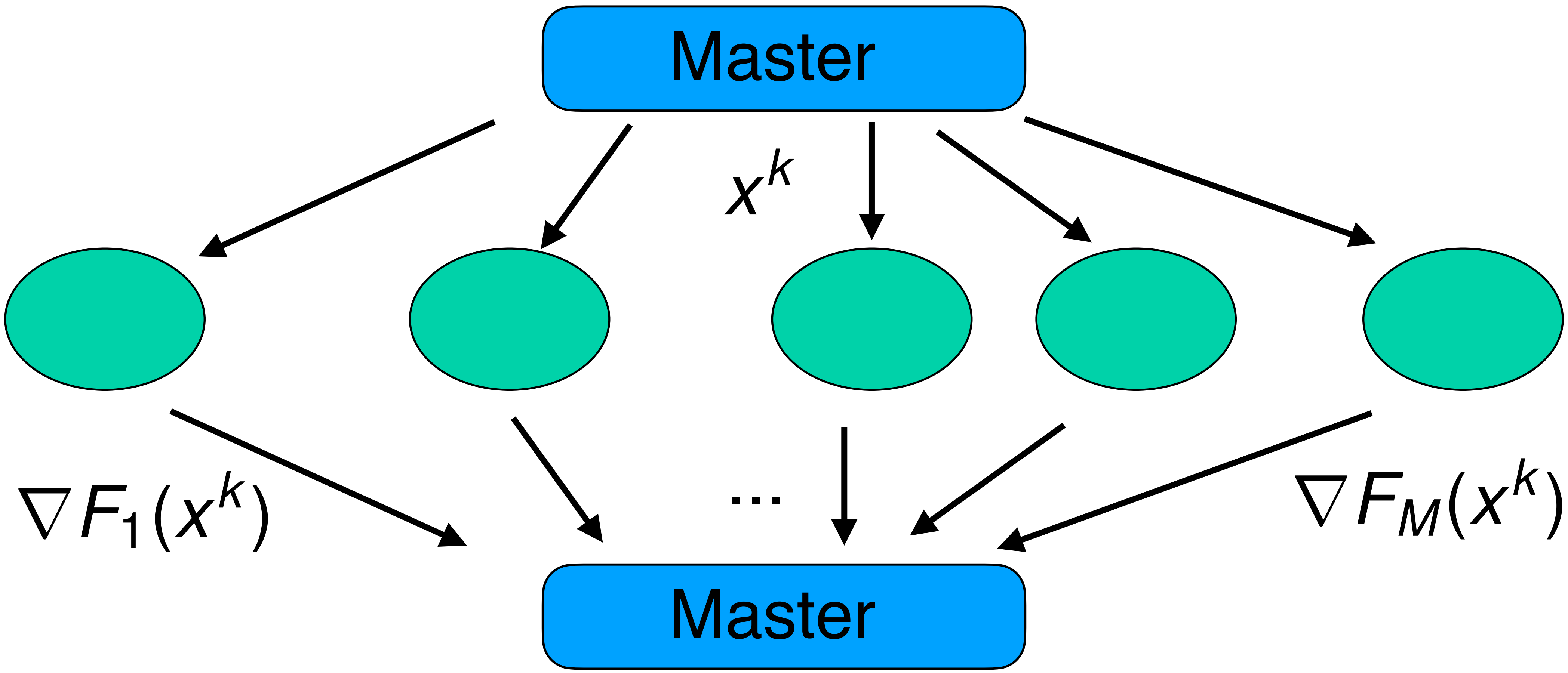

$$\frac{1}{M}\sum_{m=1}^{M}\nabla F_m(x^k)$$

# Distributed prox. GD



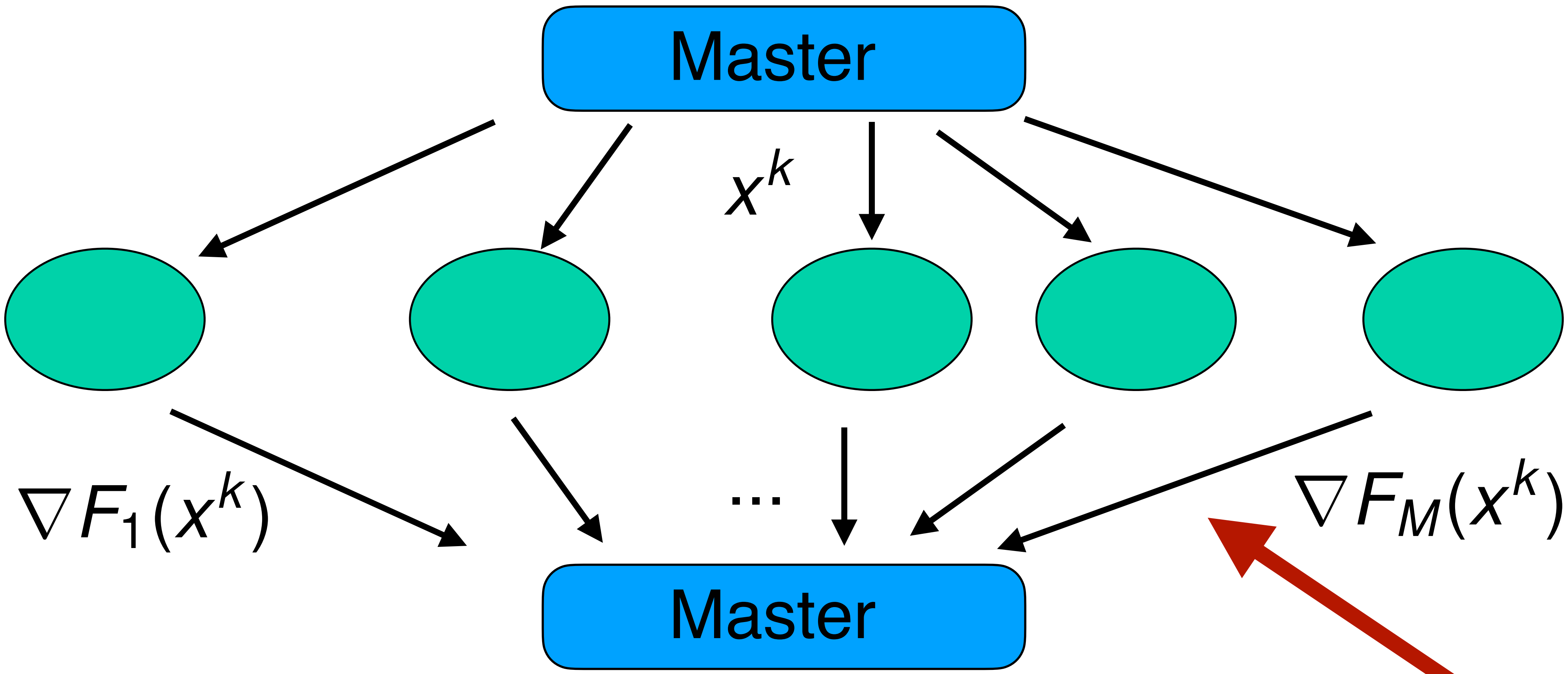

$$x^k - \frac{\gamma}{M}\sum_{m=1}^{M} \nabla F_m(x^k)$$

# Distributed prox. GD

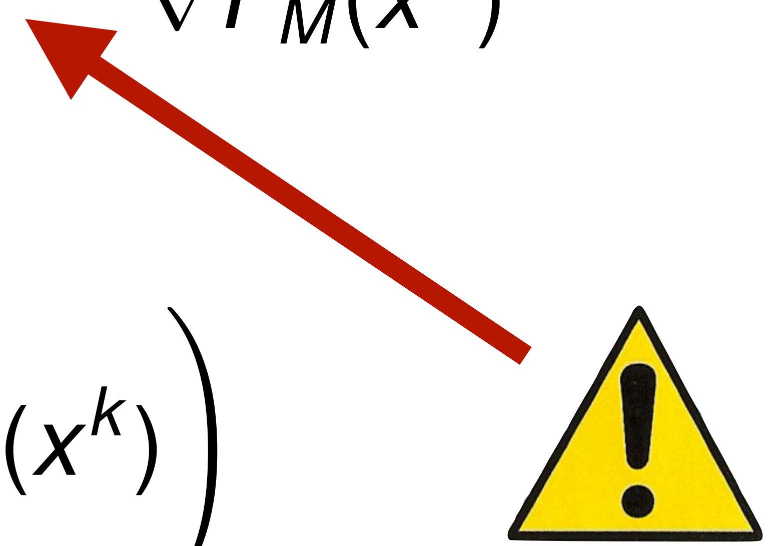



$$x^{k+1} := \mathrm{prox}_{\gamma R}\left(x^k - \frac{\gamma}{M}\sum_{m=1}^{M} \nabla F_m(x^k)\right)$$

# Distributed prox. GD

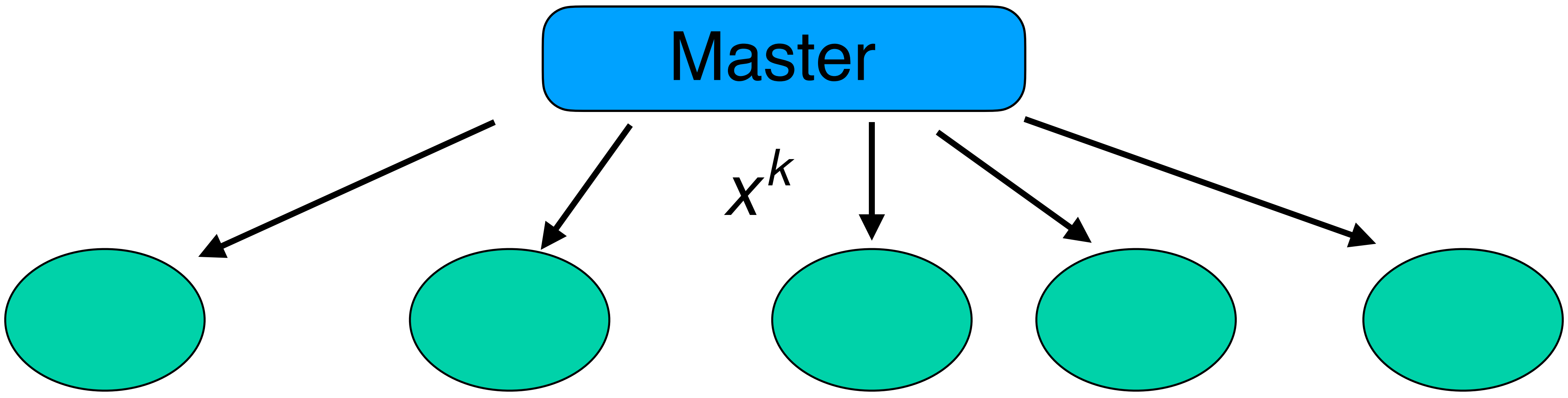



$$x^{k+1} := \mathrm{prox}_{\gamma R}\left(x^k - \frac{\gamma}{M}\sum_{m=1}^{M} \nabla F_m(x^k)\right)$$



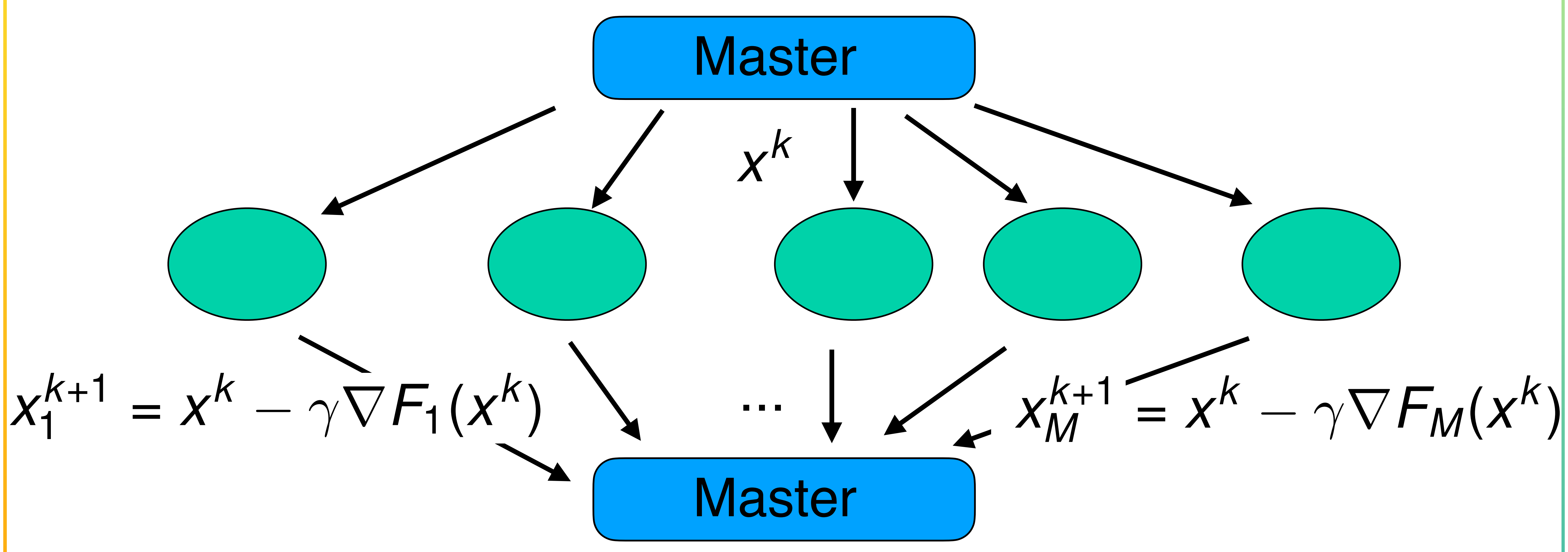
# *1) local computations*

# Distributed prox. GD



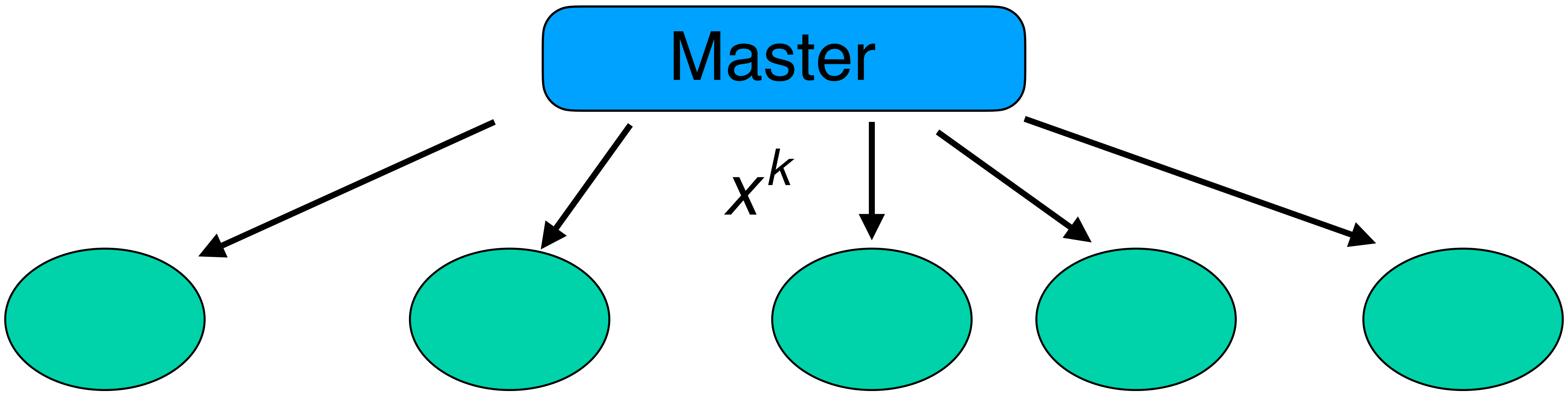

$x_1^{k+1} = x^k - \gamma \nabla F_1(x^k)$ $\qquad\cdots\qquad$ $x_M^{k+1} = x^k - \gamma \nabla F_M(x^k)$

# Distributed prox. GD




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

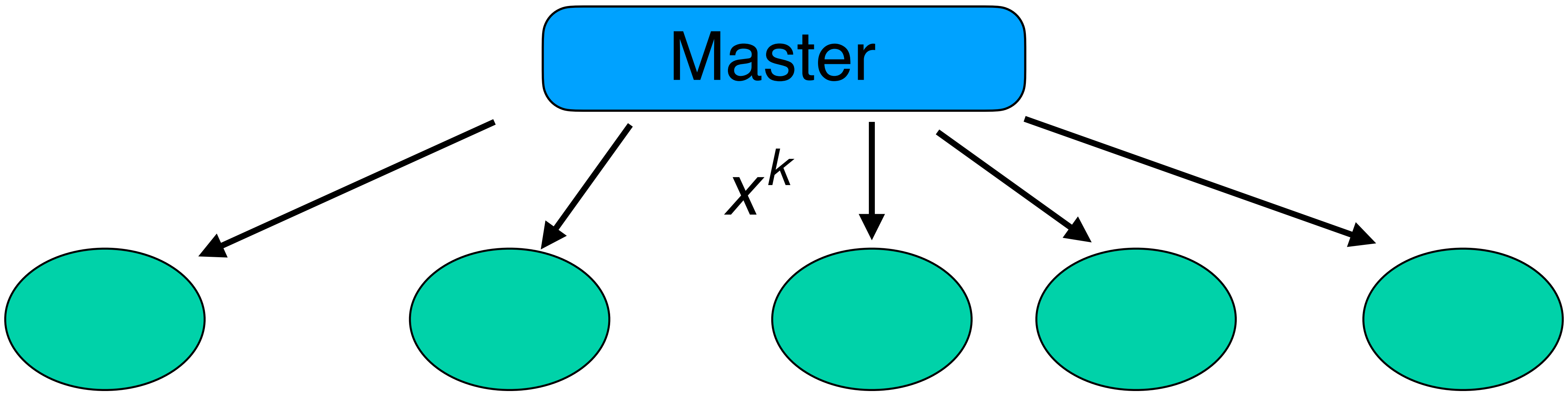
# Distributed prox. **Local** GD




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

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

# Distributed prox. **Local** GD




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

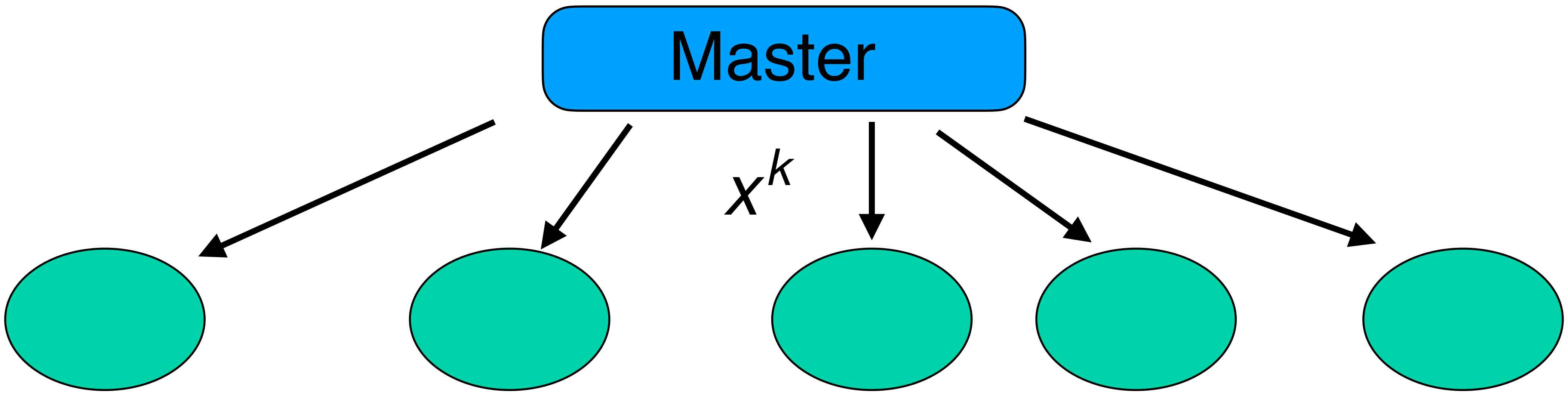
$$x_1^{k+2} = x_1^{k+1} - \gamma \nabla F_1(x_1^{k+1}) \qquad x_M^{k+2} = x_M^{k+1} - \gamma \nabla F_M(x_M^{k+1})$$

$$\cdots \qquad \cdots$$

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

$$H \geq 1$$

# Distributed prox. **Local** GD

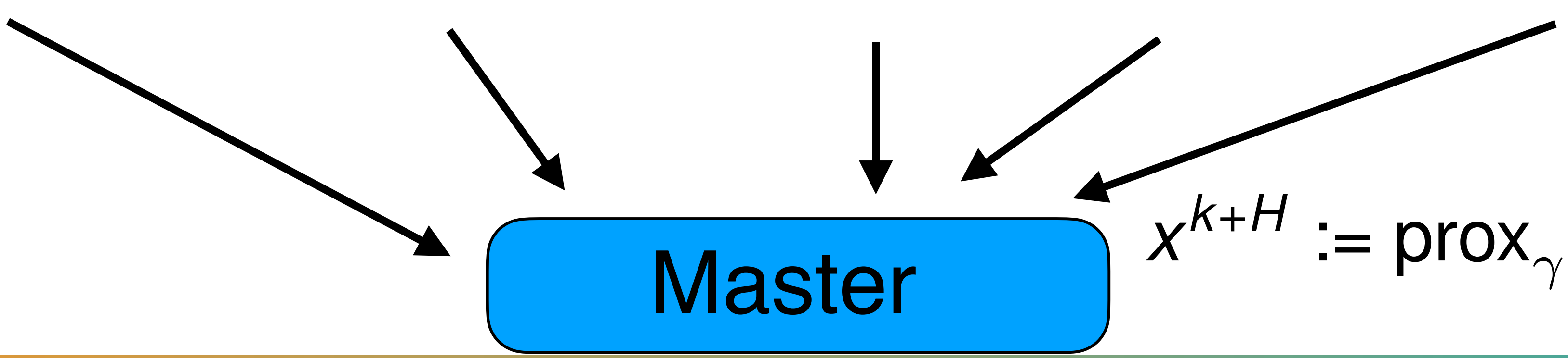



$$x_1^{k+1} = x^k - \gamma \nabla F_1(x^k) \qquad \cdots \qquad x_M^{k+1} = x^k - \gamma \nabla F_M(x^k)$$

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

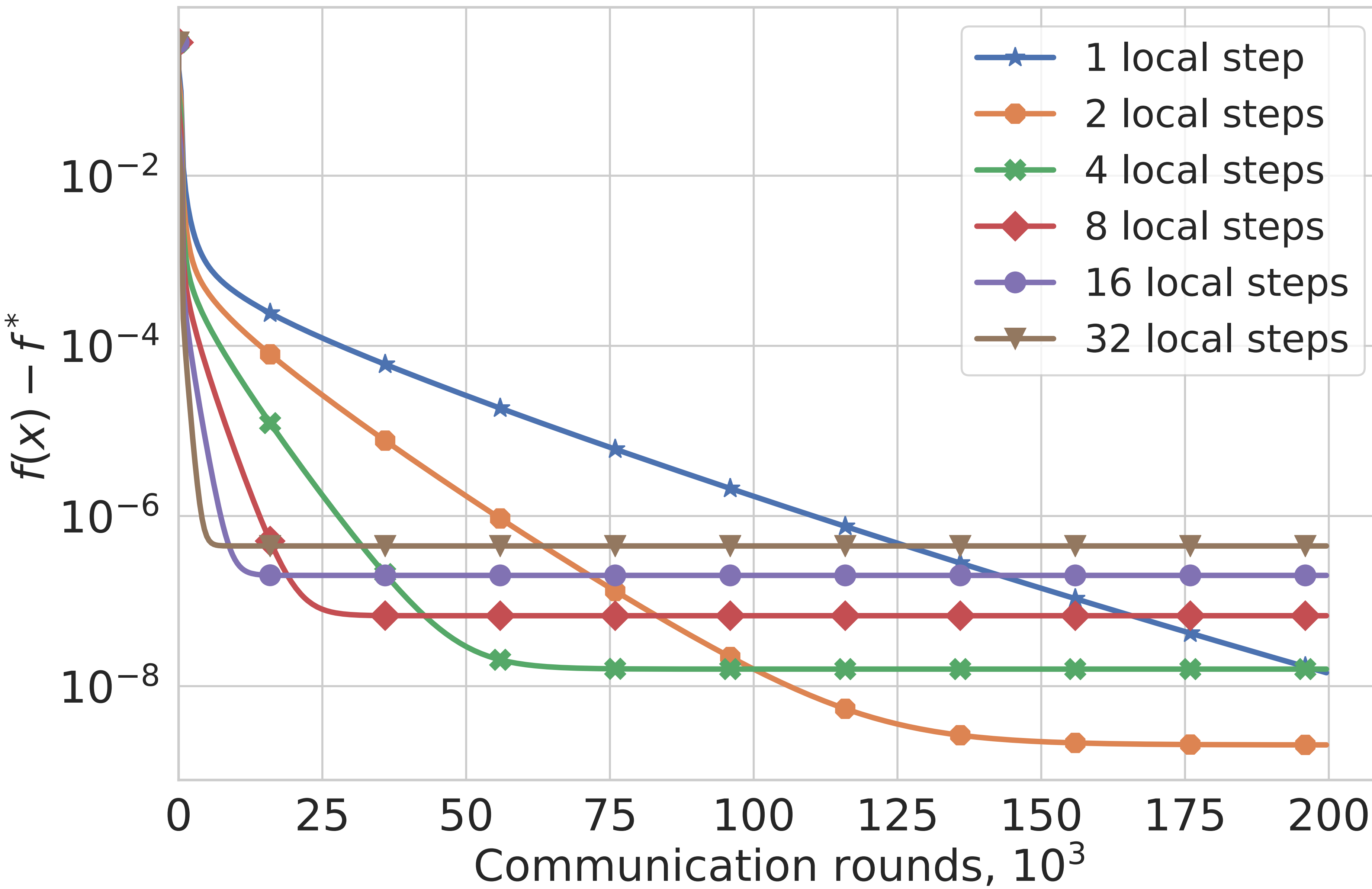
$$\cdots \qquad \cdots$$

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




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

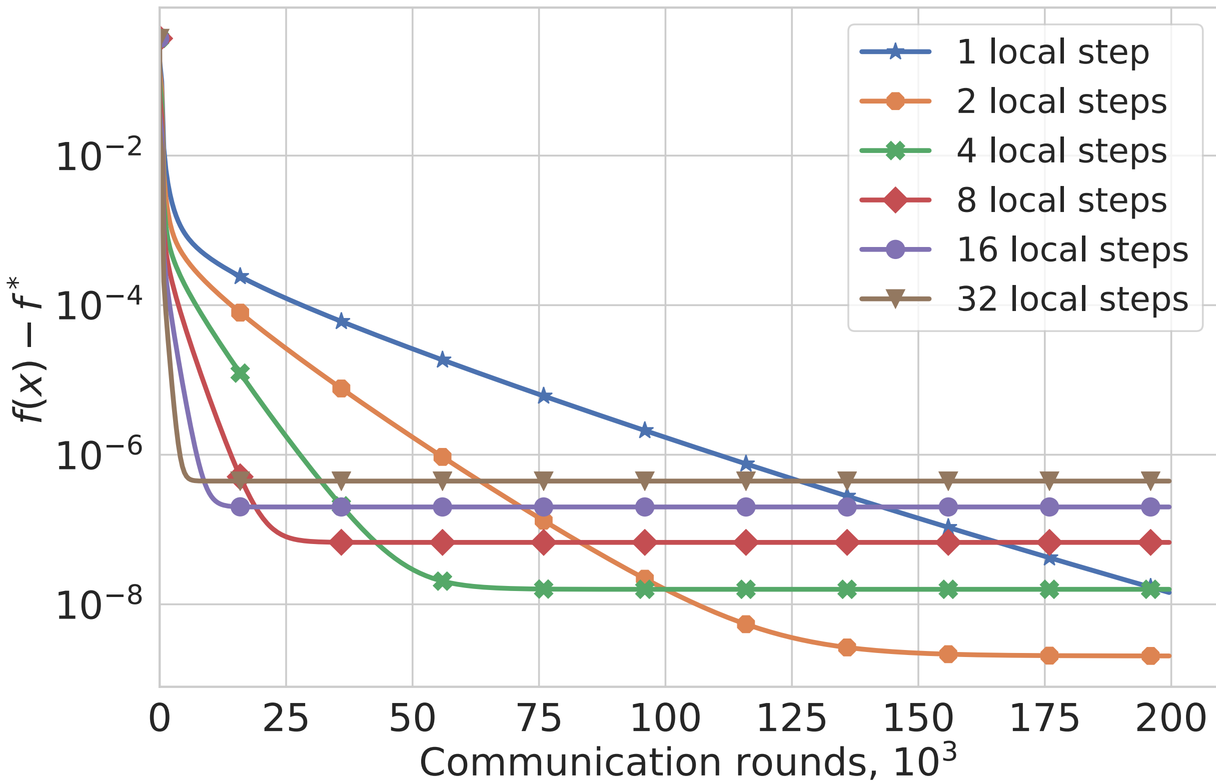
# Local GD: performance




(logistic regression)

# Local GD: analysis

* Stich, Local SGD Converges Fast and Communicates Little. *ICLR* 2019.
* Khaled, Mishchenko, and Richtárik, First analysis of local GD on heterogeneous data. *NeurIPS Workshop on Federated Learning for Data Privacy and Confidentiality*, 2019.
* Khaled, Mishchenko, and Richtárik, Tighter theory for local SGD on identical and heterogeneous data. *AISTATS* 2020.
* Ma, Konecny, Jaggi, Smith, Jordan, Richtárik, and Takác, Distributed optimization with arbitrary local solvers. *Optimization Methods and Software*, 2017.
* Haddadpour and Mahdavi, On the convergence of local descent methods in federated learning, *arXiv:1910.14425*, 2019.




# Local GD: analysis

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

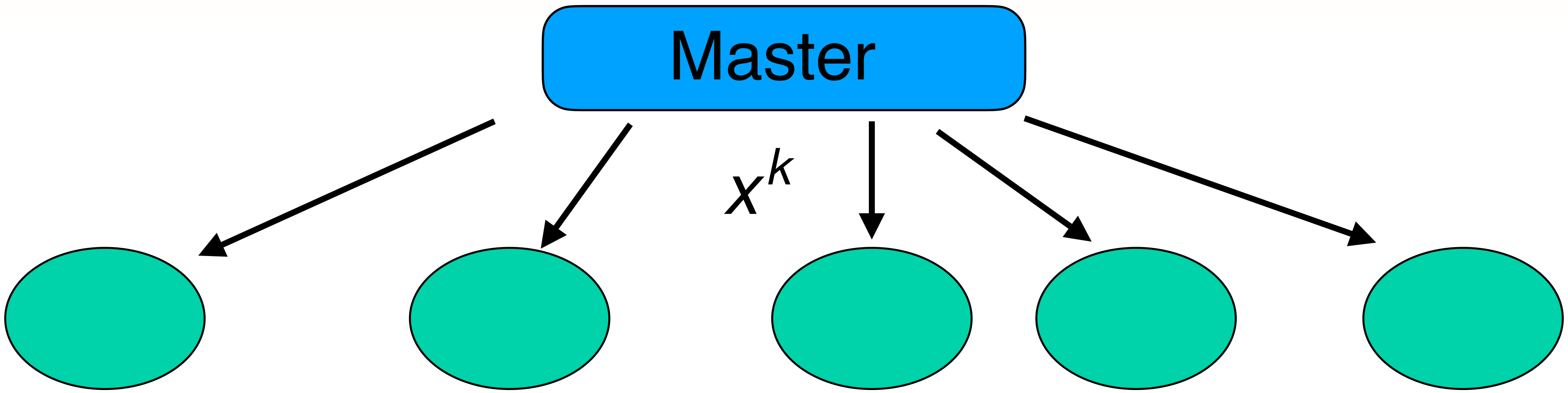
**Theorem 2.11 (linear convergence)** With $\gamma \in (0, \frac{2}{L+\mu}]$, $(x^{nH})_{n\geq 0}$ converges linearly to $x^\dagger$ with rate $\xi^H$, where $\xi = 1 - \gamma\mu$, and

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

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




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

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

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

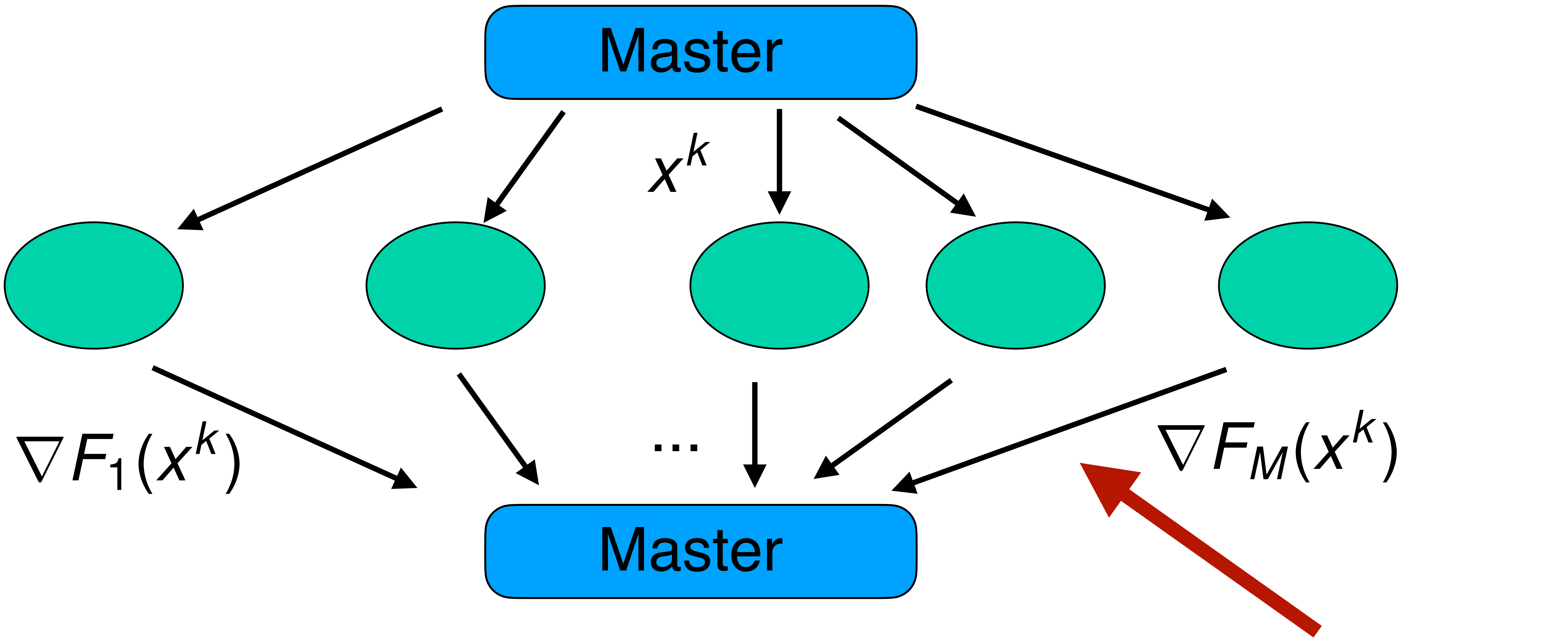
$$x_1^{k+H} = x_1^{k+H-1} - \gamma \nabla F_1(x_1^{k+H-1}) + h_1^k \qquad x_M^{k+H} = x_M^{k+H-1} - \gamma \nabla F_M(x_M^{k+H-1}) + h_M^k$$

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,” arXiv:2207.12891, 2022

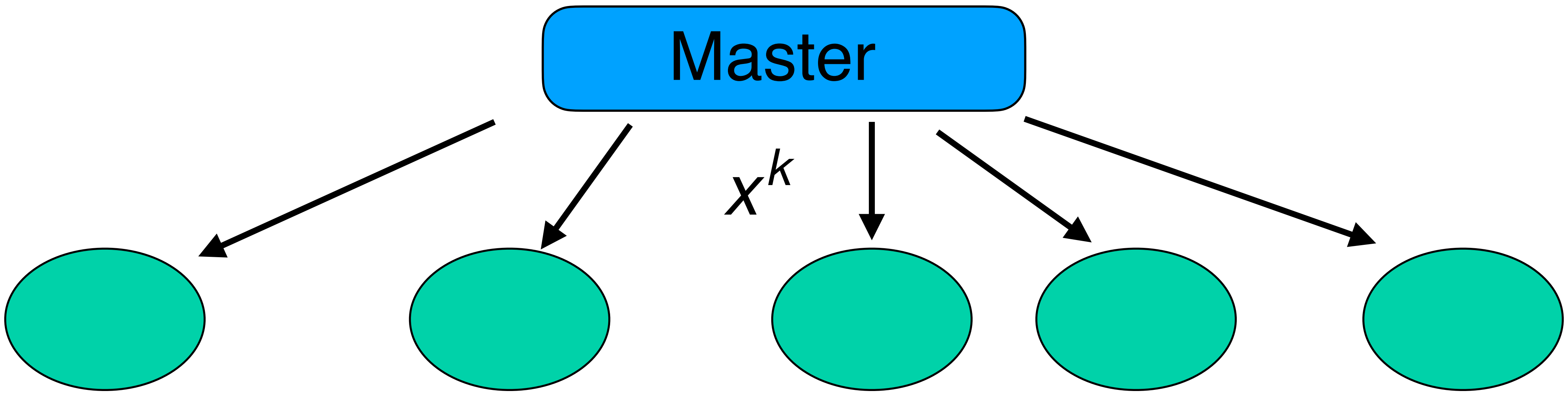
# 2) compression

# Dist. prox. GD



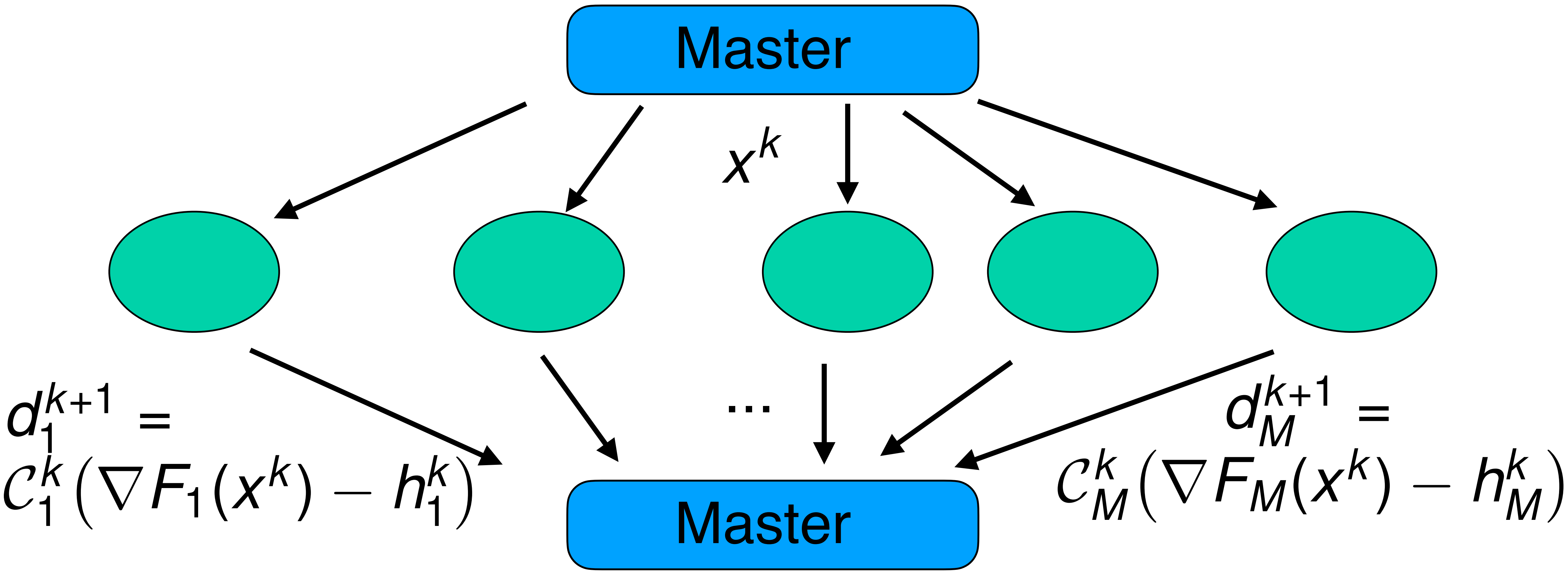

$$x^{k+1} := \mathrm{prox}_{\gamma R}\left(x^k - \frac{\gamma}{M}\sum_{m=1}^{M}\nabla F_m(x^k)\right)$$

# Dist. prox. GD with compression

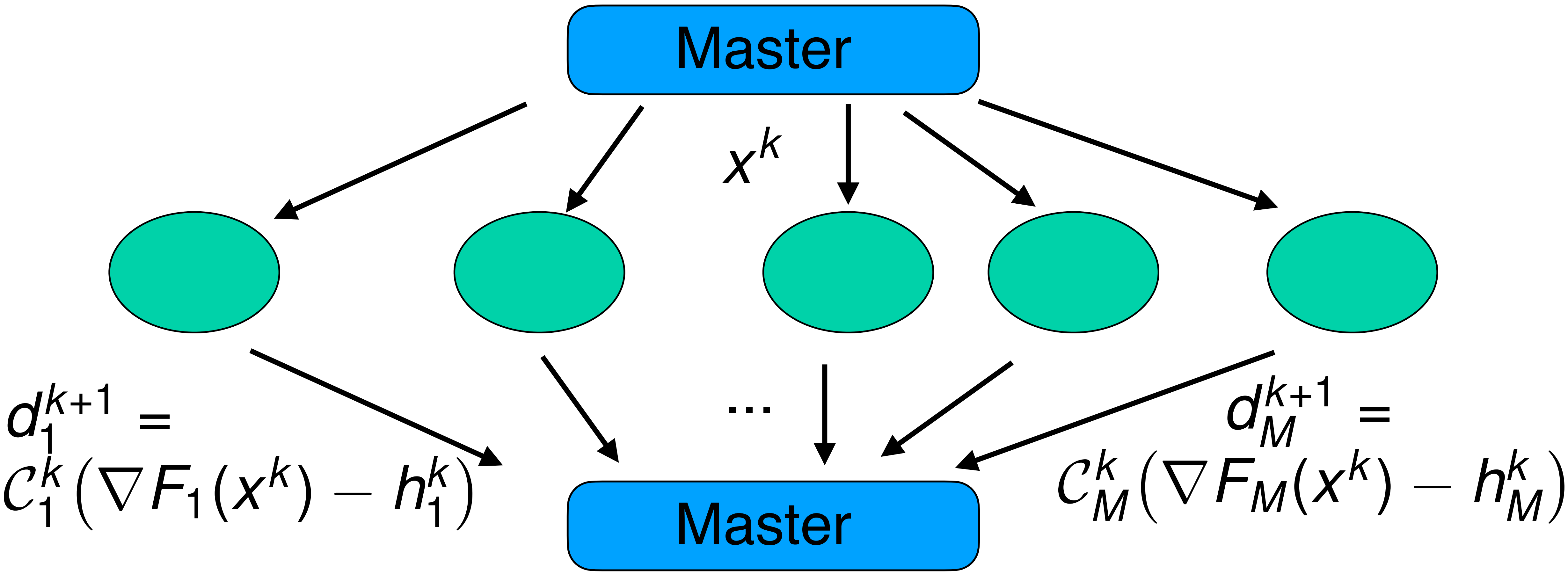



$d_1^{k+1} = \mathcal{C}_1^k\big(\nabla F_1(x^k) - h_1^k\big)$ ... $d_M^{k+1} = \mathcal{C}_M^k\big(\nabla F_M(x^k) - h_M^k\big)$

# Dist. prox. GD with compression

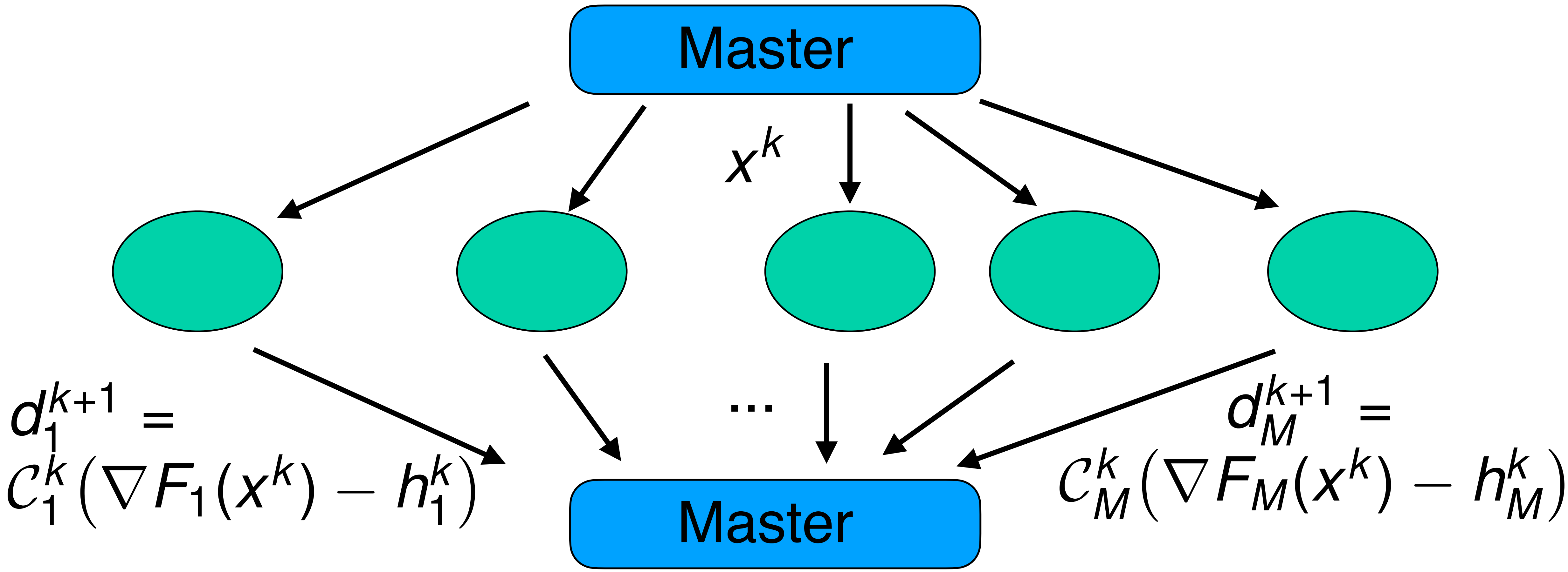


# Dist. prox. GD with compression




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

# Dist. prox. GD with compression




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

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

# Dist. prox. GD with compression

**Algorithm 1** (EF-BV)

```
input: parameters γ > 0, λ > 0, ν > 0,
initial vectors x^0 ∈ ℝ^d and h_m^0 ∈ ℝ^d
h^0 := (1/M) Σ_{m=1}^M h_m^0
for k = 0, 1, ... do
    for m = 1, ..., M in parallel do
        d_m^{k+1} := C_m^k(∇F_m(x^k) − h_m^k)
        h_m^{k+1} := h_m^k + λ d_m^{k+1}
    end for
    // at master:
    d^{k+1} := (1/M) Σ_{m=1}^M d_m^{k+1}
    x^{k+1} := prox_{γR}(x^k − γ(h^k + ν d^{k+1}))
    h^{k+1} := h^k + λ d^{k+1}
end for
```

**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**

    $d_m^{k+1} := \mathcal{C}_m^k\big(\nabla F_m(x^k) - h_m^k\big)$

    $h_m^{k+1} := h_m^k + \lambda d_m^{k+1}$

  **end for**

  // at master:

  $d^{k+1} := \frac{1}{M}\sum_{m=1}^{M} d_m^{k+1}$

  $x^{k+1} := \mathrm{prox}_{\gamma R}\big(x^k - \gamma(h^k + \nu d^{k+1})\big)$

  $h^{k+1} := h^k + \lambda d^{k+1}$

**end for**

Condat, Yi, Richtárik, “EF-BV: A unified theory of error feedback and variance reduction…”, NeurIPS 2022

# Dist. prox. GD with compression

**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**
    $d_m^{k+1} := \mathcal{C}_m^k\big(\nabla F_m(x^k) - h_m^k\big)$
    $h_m^{k+1} := h_m^k + \lambda d_m^{k+1}$
  **end for**
  // at master:
  $d^{k+1} := \frac{1}{M}\sum_{m=1}^{M} d_m^{k+1}$
  $x^{k+1} := \mathrm{prox}_{\gamma R}\big(x^k - \gamma(h^k + \nu d^{k+1})\big)$
  $h^{k+1} := h^k + \lambda d^{k+1}$
**end for**

Condat, Yi, Richtárik, "EF-BV: A unified theory of error feedback and variance reduction…", NeurIPS 2022

$\nu = 1$ and unbiased compressors: DIANA [Mishchenko et al. 2019] generalized in:

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

# DIANA

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

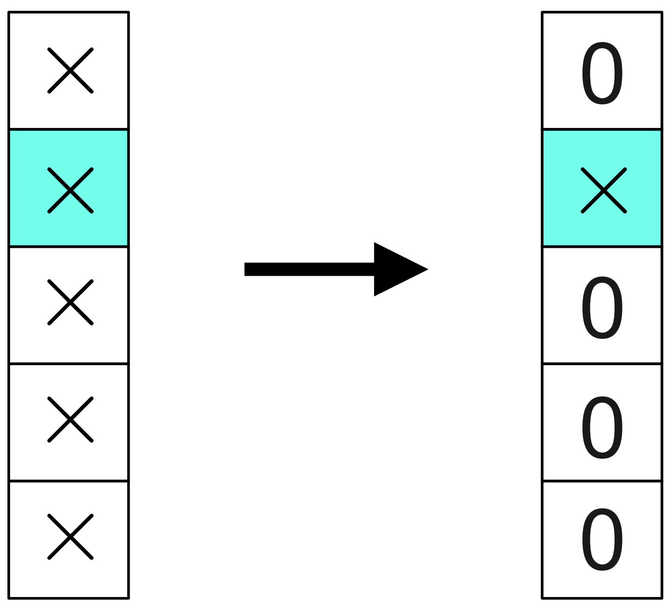
with stochastic gradients

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

which are unbiased: $\mathbb{E}\big[g_m^k\big] = \nabla F_m(x^k)$

# Examples of unbiased compressors

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



# Examples of unbiased compressors

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

# Examples of unbiased compressors

- `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

# Variance of unbiased compressors

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

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

# Variance of unbiased compressors

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

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

We also define $\omega_{\mathrm{av}} \in [0, \omega]$ and $\zeta \in [0, \omega_{\mathrm{av}}]$ such that, for any $(v_m)_{m=1}^M$,

$$\mathbb{E}\left[\left\|\frac{1}{M}\sum_{m=1}^{M}\left(\mathcal{C}_m^k(v_m) - v_m\right)\right\|^2\right] \leq \frac{\omega_{\mathrm{av}}}{M}\sum_{m=1}^{M}\|v_m\|^2 - \zeta\left\|\frac{1}{M}\sum_{m=1}^{M} v_m\right\|^2$$

☞ If the $(\mathcal{C}_m^k)_{m=1}^M$ are independent, $\omega_{\mathrm{av}} = \frac{\omega}{M}$

# DIANA: convergence

**Theorem** [MURANA] In DIANA, set $\lambda := \frac{1}{1+\omega}$ and suppose that

$$0 < \gamma < \frac{2}{L}\frac{1}{a + 4\omega_{\mathrm{av}}},$$

where $a := \max\left(1 - 2\zeta, 0\right)$. Choose $b > 1$ s.t. $\eta := 1 - \gamma\left(\frac{2}{L}\frac{1}{a+(1+b)^2\omega_{\mathrm{av}}}\right)^{-1} \in (0, 1)$. Define the Lyapunov function, for every $k \geq 0$,

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

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

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

# DIANA: convergence

**Theorem** [MURANA] In DIANA, set $\lambda \coloneqq \frac{1}{1+\omega}$ and suppose that

$$0 < \gamma < \frac{2}{L} \frac{1}{a + 4\omega_{\text{av}}},$$

☞ DIANA achieves $\epsilon$-accuracy with iteration complexity

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

# DIANA: convergence

**Theorem** [MURANA] In DIANA, set $\lambda \coloneqq \frac{1}{1+\omega}$ and suppose that

$$0 < \gamma < \frac{2}{L} \frac{1}{a + 4\omega_{\mathrm{av}}},$$

☞ Typically, the communication complexity can be reduced from

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

# Conclusion

2 ideas to reduce communication:

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

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

# Conclusion

2 ideas to reduce communication:

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

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

Combining the 2 ideas: work in progress!