

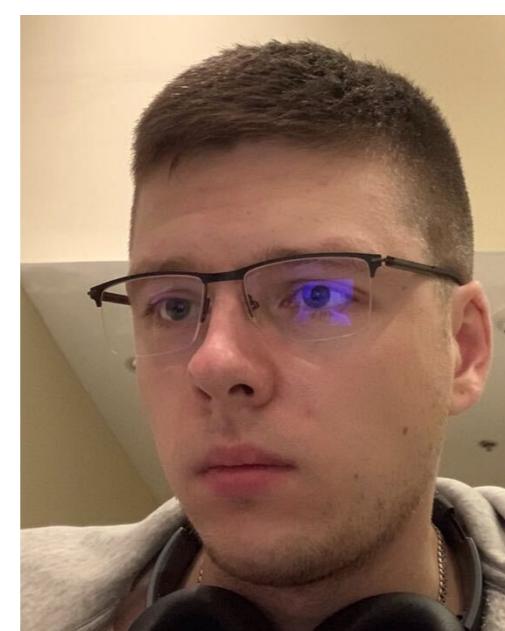
From Local SGD to Local Fixed-Point Methods for Federated Learning [ICML'20]

Laurent Condat

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



Grigory
Malinovsky



Dmitry
Kovalev



Elnur
Gasanov

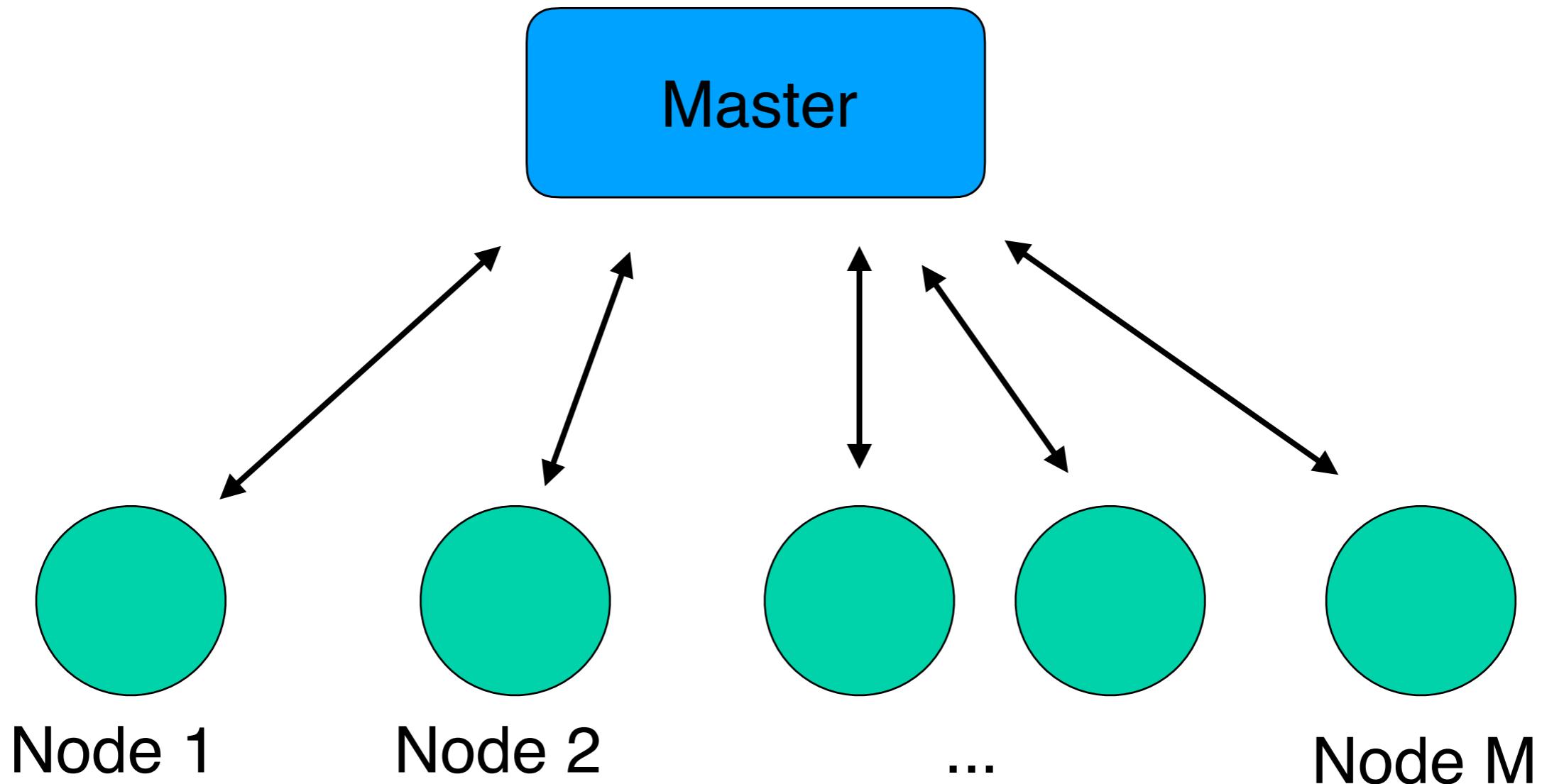


Peter
Richtárik

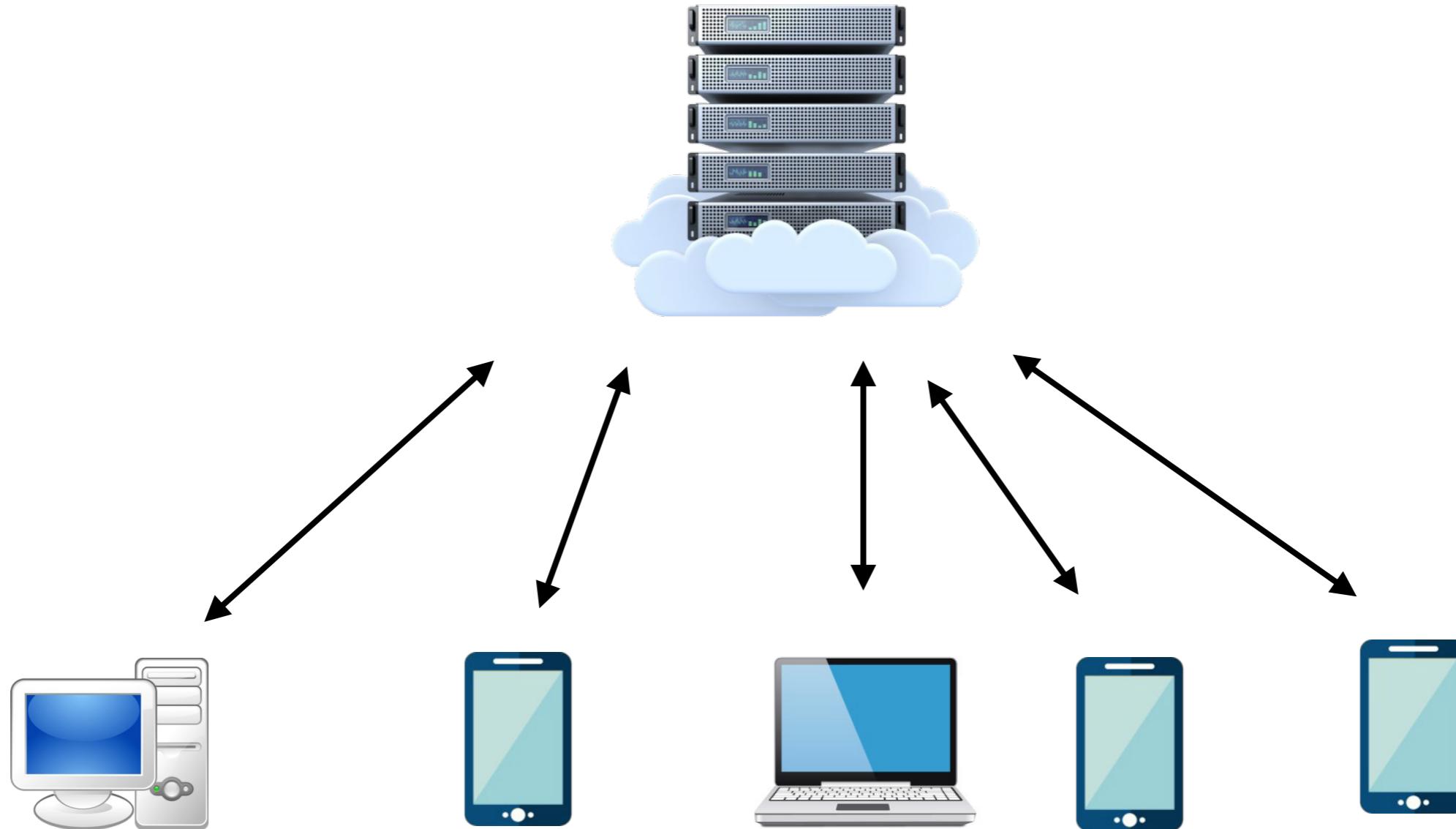
Outline

- * Context: distributed and federated learning
- * Problem formulation and motivation
- * Proposed algorithms and analysis

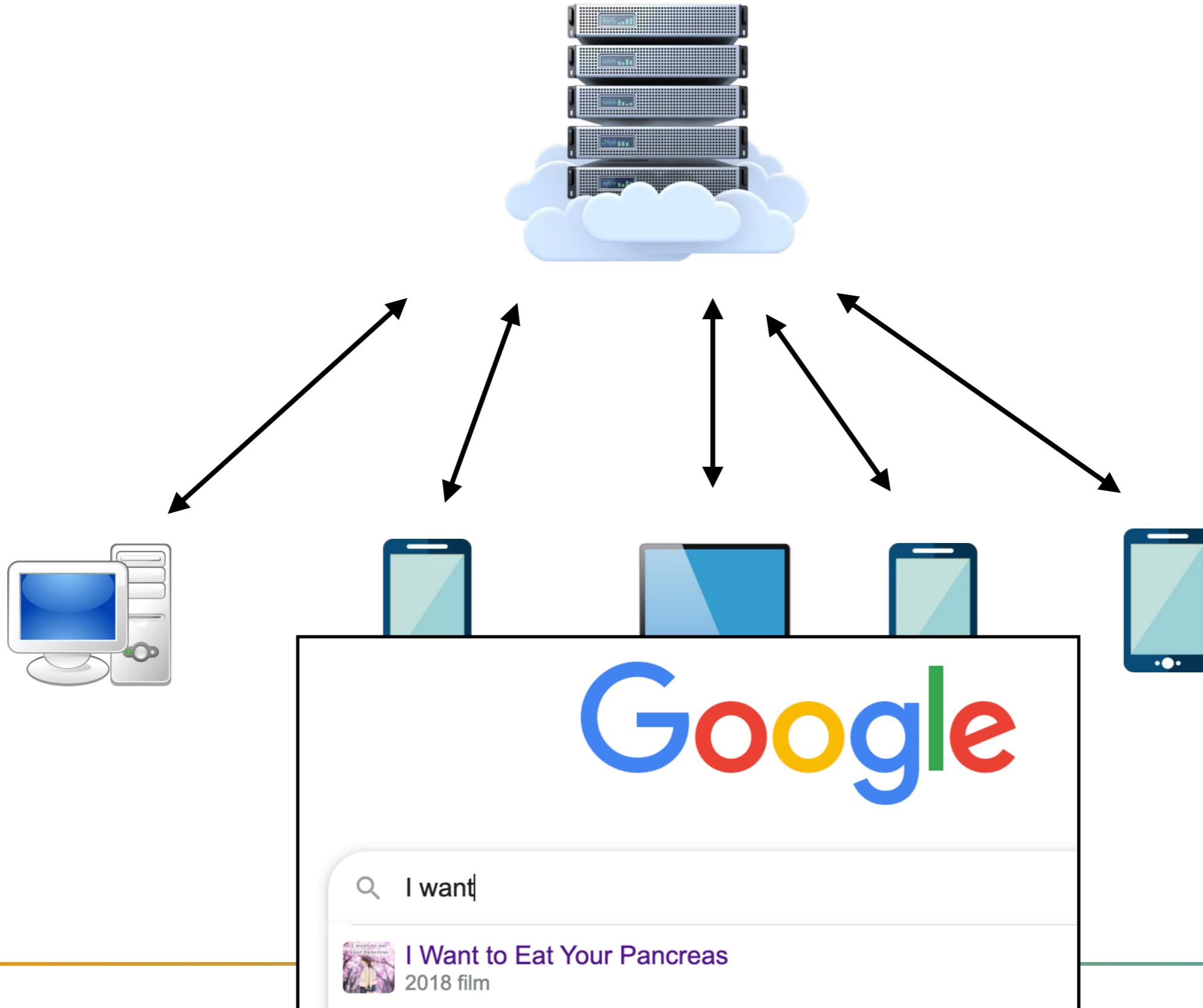
Distributed algorithms



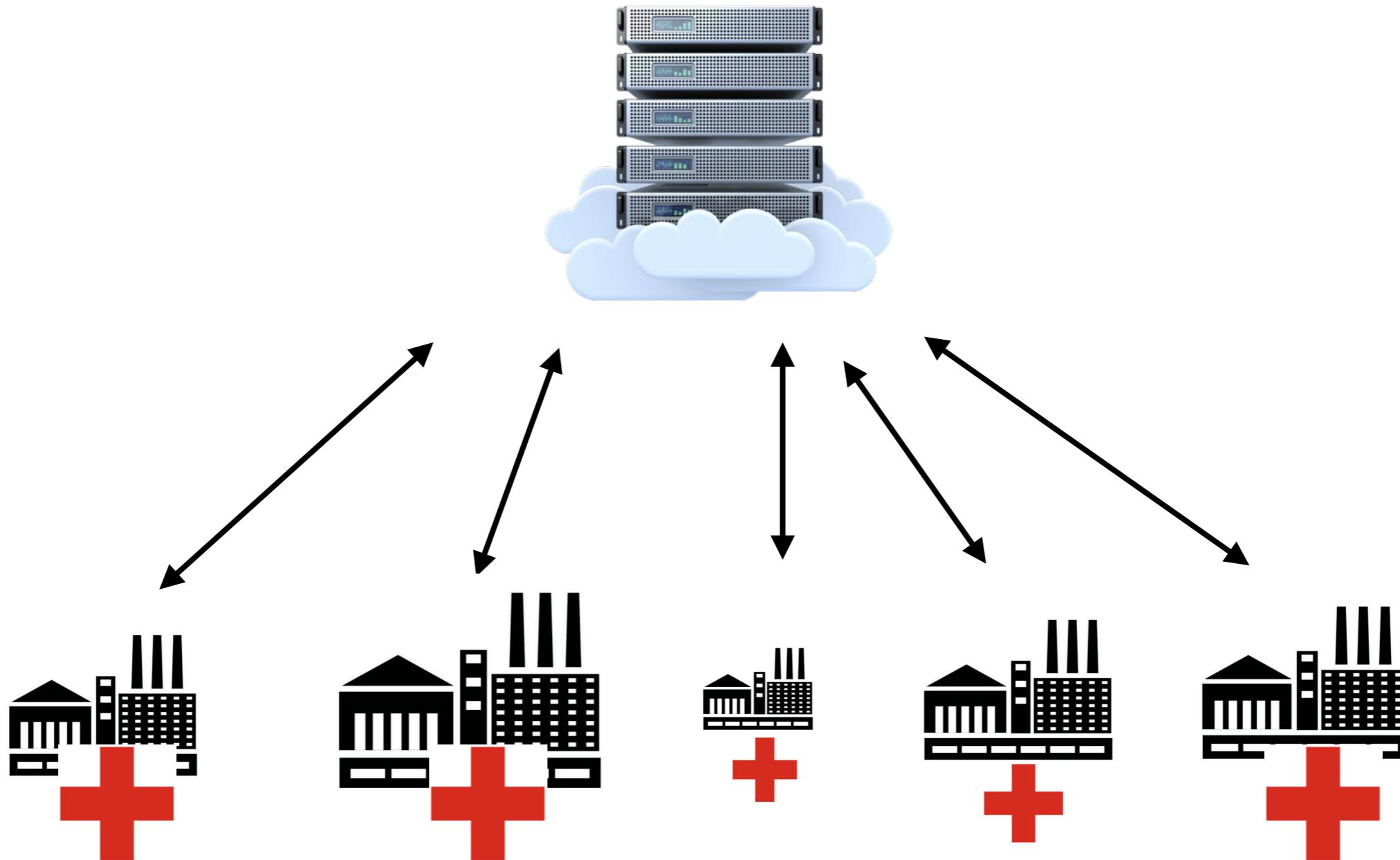
Federated learning



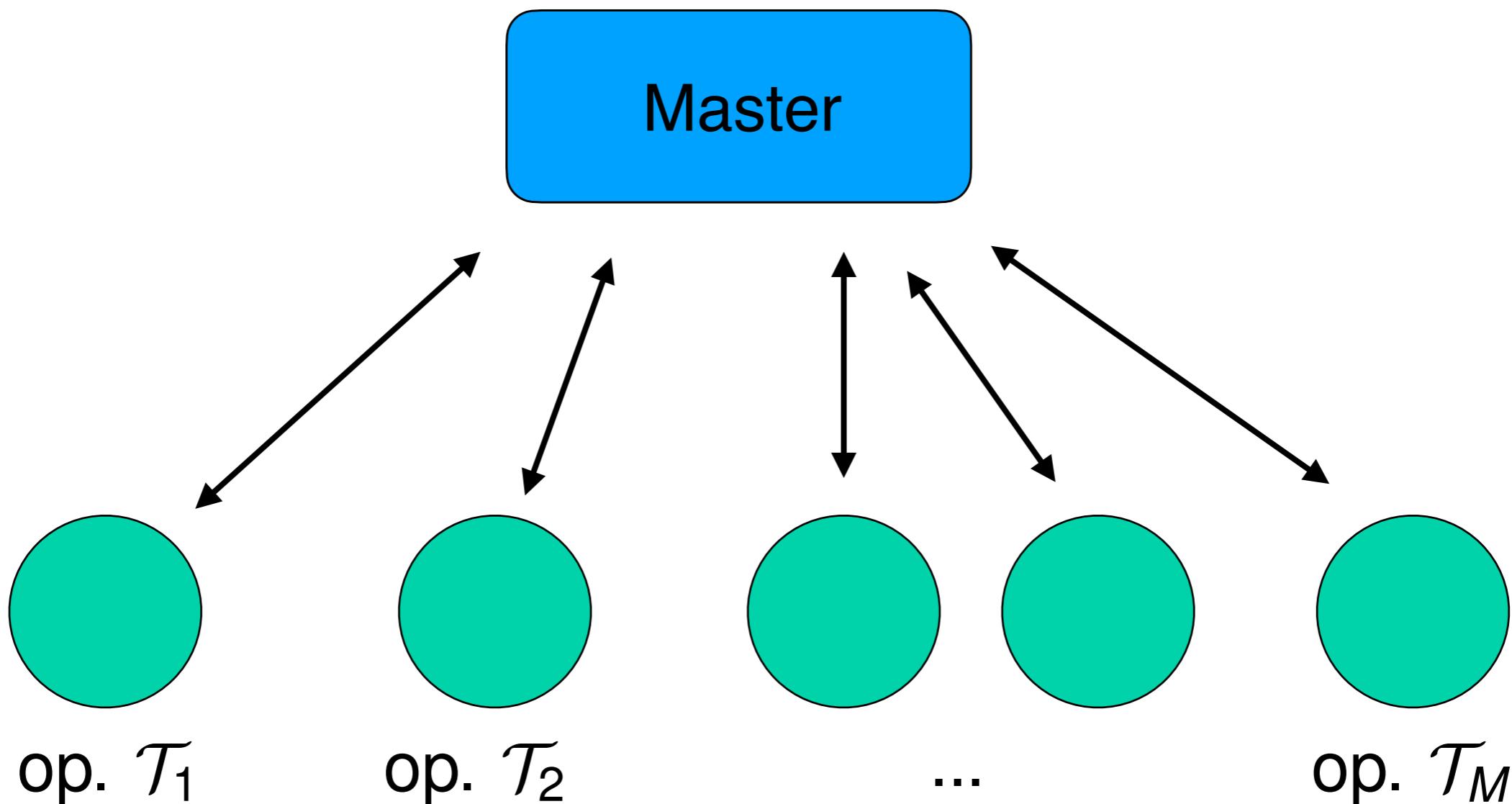
Federated learning



Federated learning



Distributed algorithms



Distributed fixed-point problem

We define the average operator

$$\mathcal{T} : x \in \mathbb{R}^d \mapsto \frac{1}{M} \sum_{i=1}^M \mathcal{T}_i(x).$$

Distributed fixed-point problem

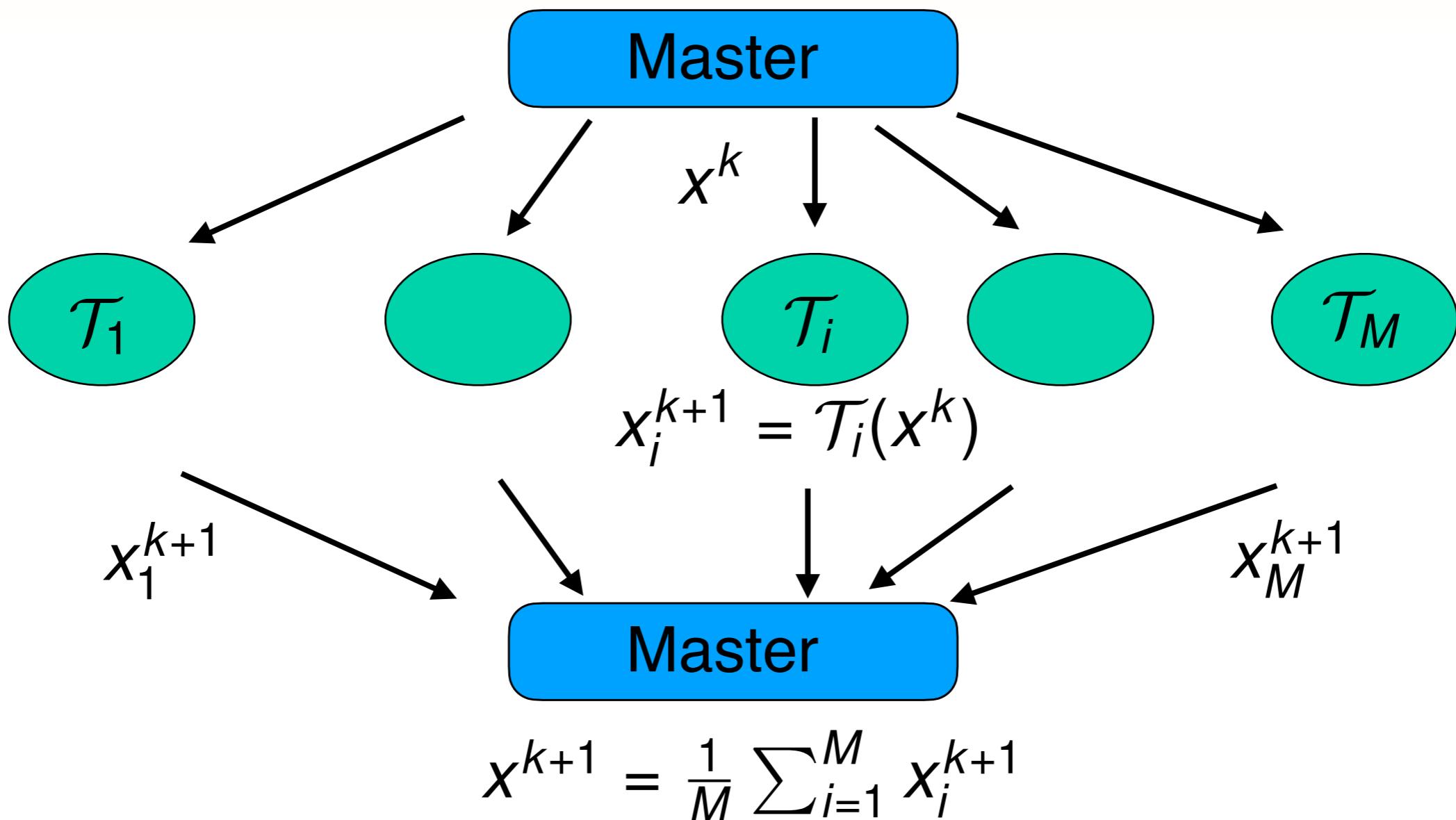
We define the average operator

$$\mathcal{T} : x \in \mathbb{R}^d \mapsto \frac{1}{M} \sum_{i=1}^M \mathcal{T}_i(x).$$

Goal: find $x^* \in \mathbb{R}^d$ such that

$$\mathcal{T}(x^*) = x^*.$$

Distributed fixed-point algorithm



$$x^{k+1} = \mathcal{T}(x^k) = \frac{1}{M} \sum_{i=1}^M \mathcal{T}_i(x^k).$$

Optimization algorithms

- * Find a minimizer of a function:

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad F(x) = \frac{1}{M} \sum_{i=1}^M F_i(x)$$

Distributed gradient descent:

$$\mathcal{T}_i : x \mapsto x - \gamma \nabla F_i(x)$$

Optimization algorithms

- * Find a minimizer of a function:

$$\underset{x \in \mathbb{R}^d}{\text{minimize}} \quad F(x) = \frac{1}{M} \sum_{i=1}^M F_i(x)$$

Distributed gradient descent:

$$\mathcal{T}_i : x \mapsto x - \gamma \nabla F_i(x)$$



$$x^{k+1} = \mathcal{T}(x^k) = \frac{1}{M} \sum_{i=1}^M \mathcal{T}_i(x^k).$$

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

$$= x^k - \gamma \nabla F(x^k)$$

Optimization algorithms

Fixed-point algorithms:

- * Find a minimizer of a function

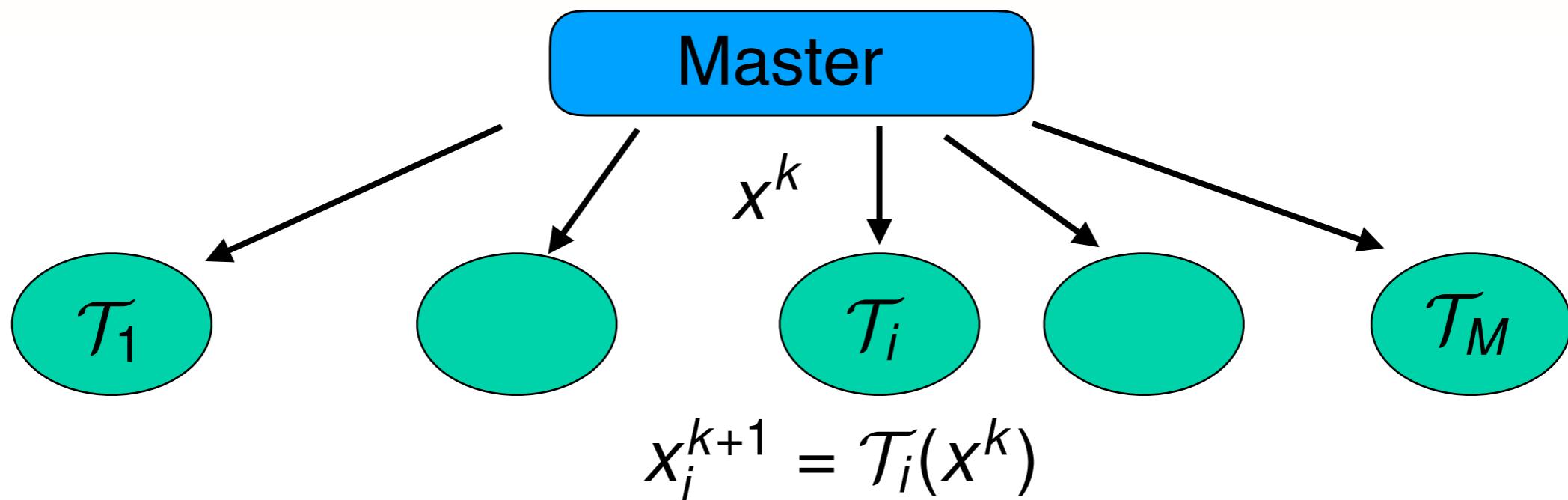
- * Proximal splitting algorithms
- * Primal-dual algorithms
- * Cyclic or shuffled GD
- * (Block-)coordinate methods
- * Methods with inertia, momentum...
- * Conjugate gradient methods
- * Higher-order methods
- * ...

Fixed-point methods

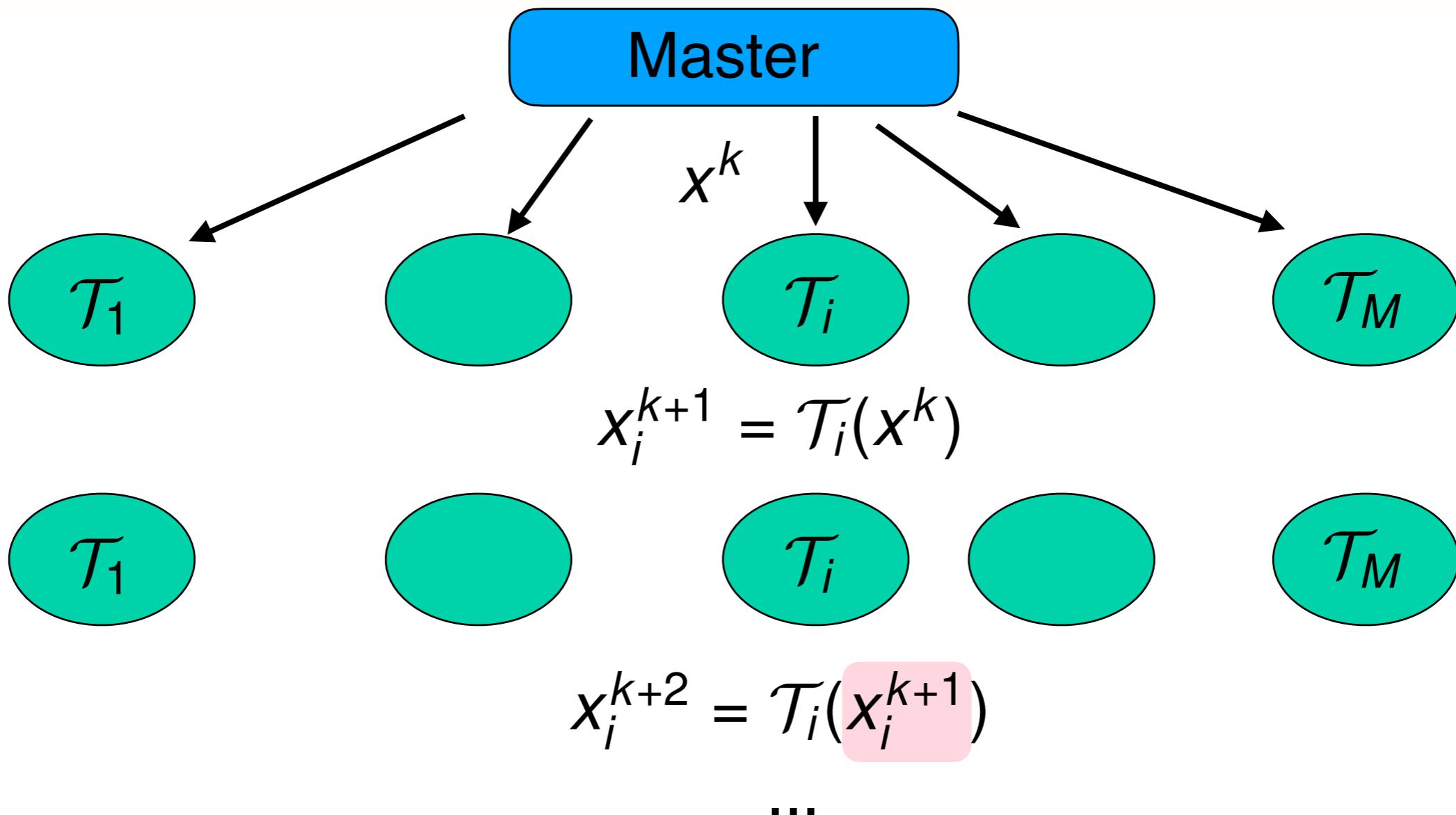
Fixed-point algorithms:

- * Find a minimizer of a function
- * Find a saddle point of a convex-concave function
- * Find a solution of a PDE
- * Find an eigenvector
- * Solve a monotone inclusion or variational inequality
- * ...

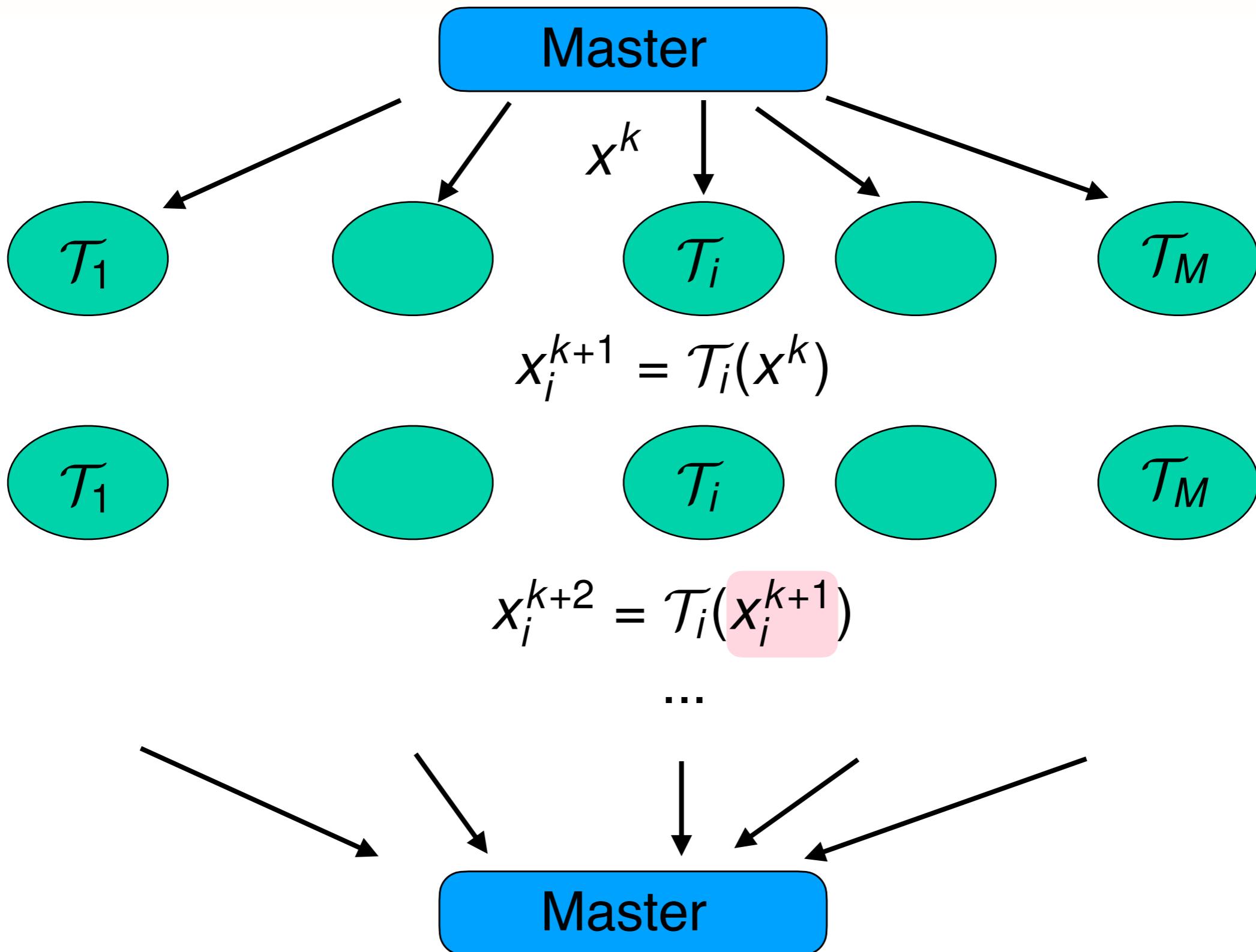
Local fixed-point method



Local fixed-point method



Local fixed-point method



Prior work: local gradient descent

- * Stich, S. U. Local SGD Converges Fast and Communicates Little. In International Conference on Learning Representations, 2019.
- * 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*, 2019.
- * 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.
- * Ma, C., Konecny, J., Jaggi, M., Smith, V., Jordan, M. I., Richtárik, P., and Takáć, M. Distributed optimization with arbitrary local solvers. *Optimization Methods and Software*, 32(4):813–848, 2017.
- * Haddadpour, F. and Mahdavi, M. On the convergence of local descent methods in federated learning. *arXiv preprint arXiv:1910.14425*, 2019.

Algorithm 1

Algorithm 1 Local distributed fixed-point method

Input: Initial estimate $\hat{x}^0 \in \mathbb{R}^d$, stepsize $\lambda > 0$, sequence of synchronization times $0 = t_0 < t_1 < \dots$

Initialize: $x_i^0 := \hat{x}^0$, for $i = 1, \dots, M$

for $k = 0, 1, \dots$ **do**

for $i = 1, 2, \dots, M$ in parallel **do**

$$h_i^{k+1} := (1 - \lambda)x_i^k + \lambda\mathcal{T}_i(x_i^k)$$

if $k + 1 = t_n$, for some n , **then**

Communicate h_i^{k+1} to master node

else

$$x_j^{k+1} := h_j^{k+1}$$

end if

end for

if $k + 1 = t_n$, for some n , **then**

At master node: $\hat{x}^{k+1} := \frac{1}{M} \sum_{i=1}^M h_i^{k+1}$

Broadcast: $x_i^{k+1} := \hat{x}^{k+1}$ for all $i = 1, \dots, M$

end if

end for

Algorithm 1

Algorithm 1 Local distributed fixed-point method

Input: Initial estimate $\hat{x}^0 \in \mathbb{R}^d$, stepsize $\lambda > 0$, sequence of synchronization times $0 = t_0 < t_1 < \dots$

Initialize: $x_i^0 := \hat{x}^0$, for $i = 1, \dots, M$

for $k = 0, 1, \dots$ **do**

for $i = 1, 2, \dots, M$ in parallel **do**

$$h_i^{k+1} := (1 - \lambda)x_i^k + \lambda\mathcal{T}_i(x_i^k)$$

if $k + 1 = t_n$, for some n , **then**

 Communicate h_i^{k+1} to master node

else

$$x_j^{k+1} := h_j^{k+1}$$

end if

end for

if $k + 1 = t_n$, for some n , **then**

 At master node: $\hat{x}^{k+1} := \frac{1}{M} \sum_{i=1}^M h_i^{k+1}$

 Broadcast: $x_i^{k+1} := \hat{x}^{k+1}$ for all $i = 1, \dots, M$

end if

end for

n-th epoch:
sequence
of iterations
 $k + 1 = t_{n-1} + 1, \dots, t_n$

Communication times

Nb of iterations in each epoch supposed bounded:

Assumption: $1 \leq t_n - t_{n-1} \leq H$, for every $n \geq 1$.

Communication times

Nb of iterations in each epoch supposed bounded:

Assumption: $1 \leq t_n - t_{n-1} \leq H$, for every $n \geq 1$.

Example:

$$t_n = nH$$

Analysis in the contractive case

- $t_n = nH$
- All \mathcal{T}_i are χ -contractive, for $\chi \in [0, 1)$
i.e. $\|\mathcal{T}_i(x) - \mathcal{T}_i(y)\| \leq \chi\|x - y\|, \quad \forall x, y$

Analysis in the contractive case

- $t_n = nH$
- All \mathcal{T}_i are χ -contractive, for $\chi \in [0, 1)$
- $\lambda = 1$

We define the operator $\tilde{\mathcal{T}} = \frac{1}{M} \sum_{i=1}^M \mathcal{T}_i^H$

Then $\hat{x}^{(n+1)H} = \tilde{\mathcal{T}}(\hat{x}^{nH})$

Analysis in the contractive case

Theorem 2.11 (linear convergence) The fixed point x^\dagger of $\tilde{\mathcal{T}}$ exists and is unique, and \hat{x}^{nH} converges linearly to x^\dagger . More precisely,

- (i) $\tilde{\mathcal{T}}$ is χ^H -contractive.
- (ii) $\forall n \in \mathbb{N}, \quad \|\hat{x}^{nH} - x^\dagger\| \leq \chi^{nH} \|\hat{x}^0 - x^\dagger\|.$

Analysis in the contractive case

Theorem 2.14 (size of the neighborhood)

Suppose that $\lambda = 1$. So, $\xi = \chi$. Then

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

where

$$S = \frac{\xi}{1 - \xi} \frac{1 - \xi^{H-1}}{1 - \xi^H} \frac{1}{M} \sum_{i=1}^M \|\mathcal{T}_i(x^*) - x^*\|.$$

Analysis in the contractive case

Theorem 2.14 (size of the neighborhood)

Suppose that $\lambda = 1$. So, $\xi = \chi$. Then

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

where

$$S = \frac{\xi}{1 - \xi} \frac{1 - \xi^{H-1}}{1 - \xi^H} \frac{1}{M} \sum_{i=1}^M \|\mathcal{T}_i(x^*) - x^*\|.$$

Note 1: $S = 0$ if $H = 1$, or $M = 1$, or $\mathcal{T}_i = \mathcal{T}$, or $\xi = 0$.

Analysis in the contractive case

Theorem 2.14 (size of the neighborhood)

Suppose that $\lambda = 1$. So, $\xi = \chi$. Then

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

where

$$S = \frac{\xi}{1 - \xi} \frac{1 - \xi^{H-1}}{1 - \xi^H} \frac{1}{M} \sum_{i=1}^M \|\mathcal{T}_i(x^*) - x^*\|.$$

Note 1: $S = 0$ if $H = 1$, or $M = 1$, or $\mathcal{T}_i = \mathcal{T}$, or $\xi = 0$.

Note 2: If $H : 1 \nearrow +\infty$, $S : 0 \nearrow S^\infty$ with

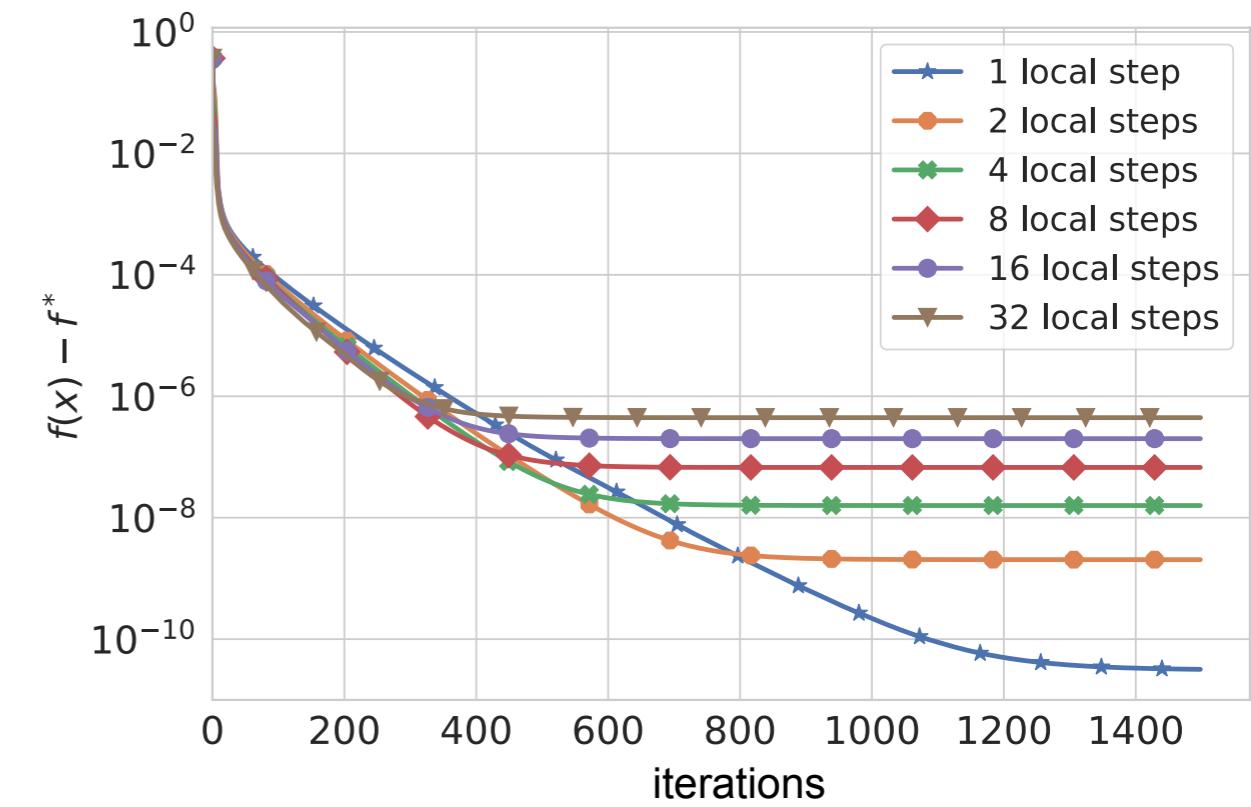
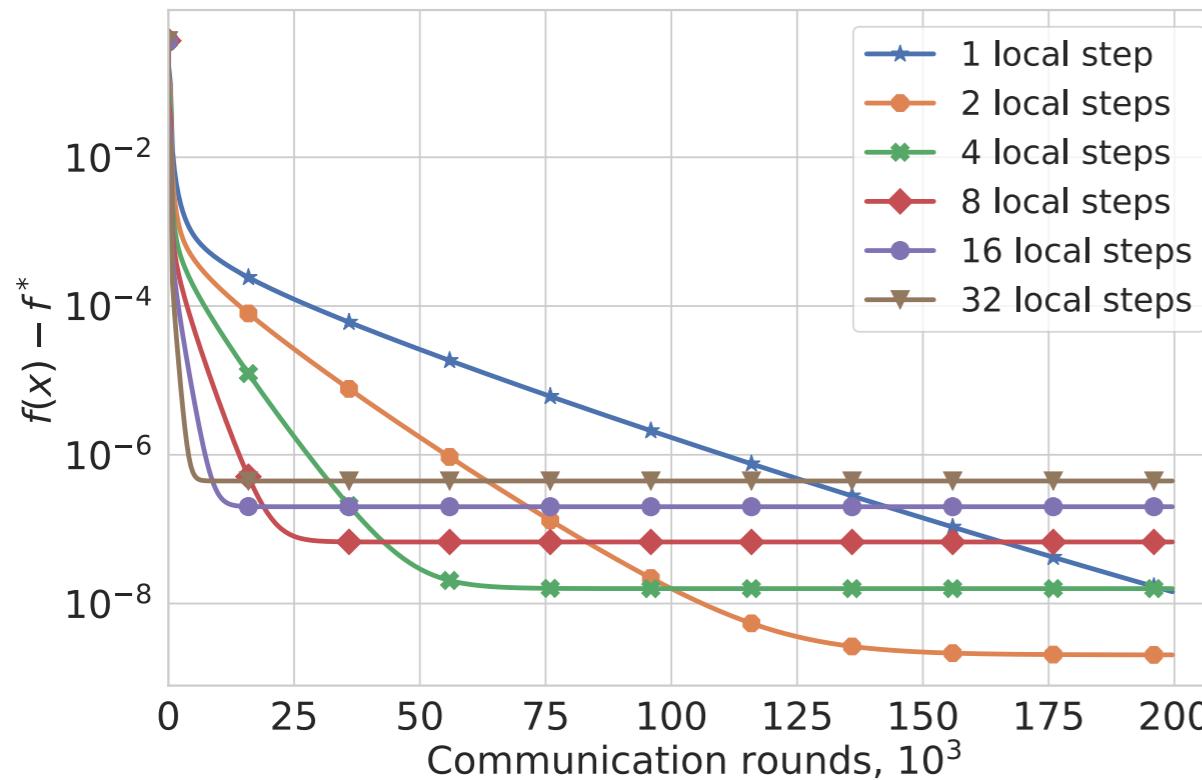
$$S^\infty = \frac{\xi}{1 - \xi} \frac{1}{M} \sum_{i=1}^M \|\mathcal{T}_i(x^*) - x^*\|.$$

Analysis in the contractive case

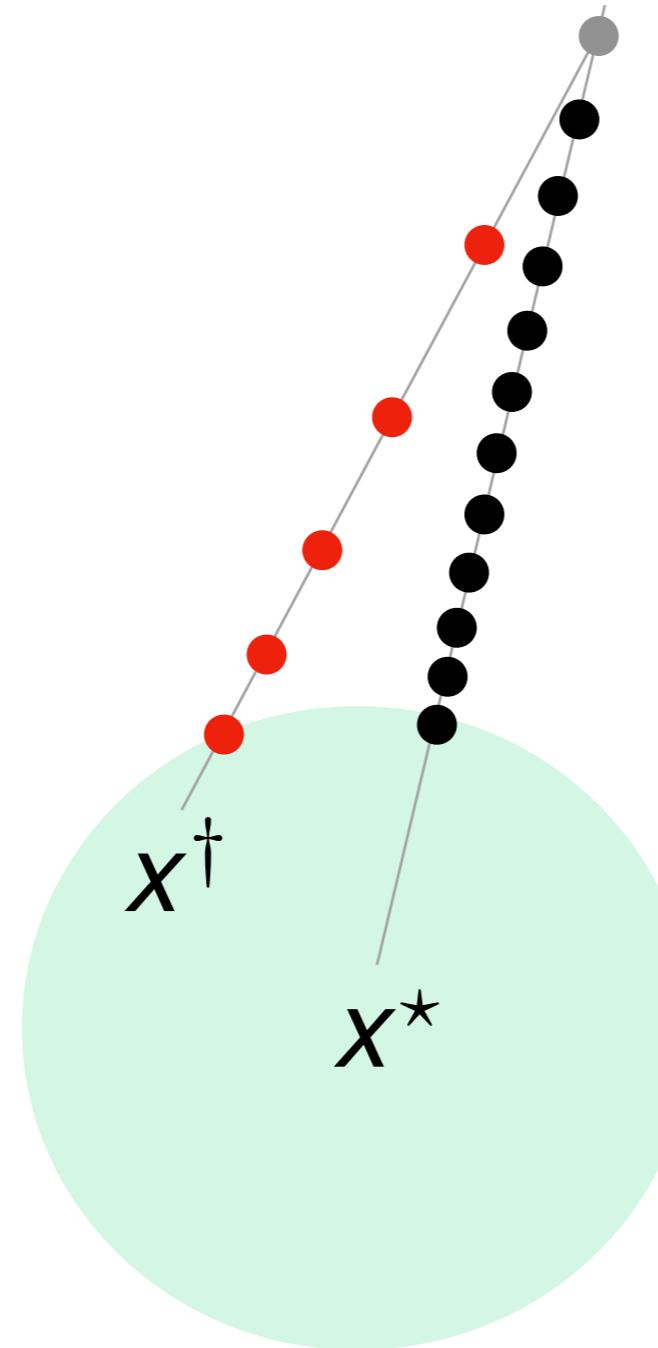
Corollary: For every $n \in \mathbb{N}$,

$$\begin{aligned}\|\hat{x}^{nH} - x^*\| &\leq \xi^{nH} \|\hat{x}^0 - x^\dagger\| + S \\ &\leq \xi^{nH} (\|\hat{x}^0 - x^*\| + S) + S.\end{aligned}$$

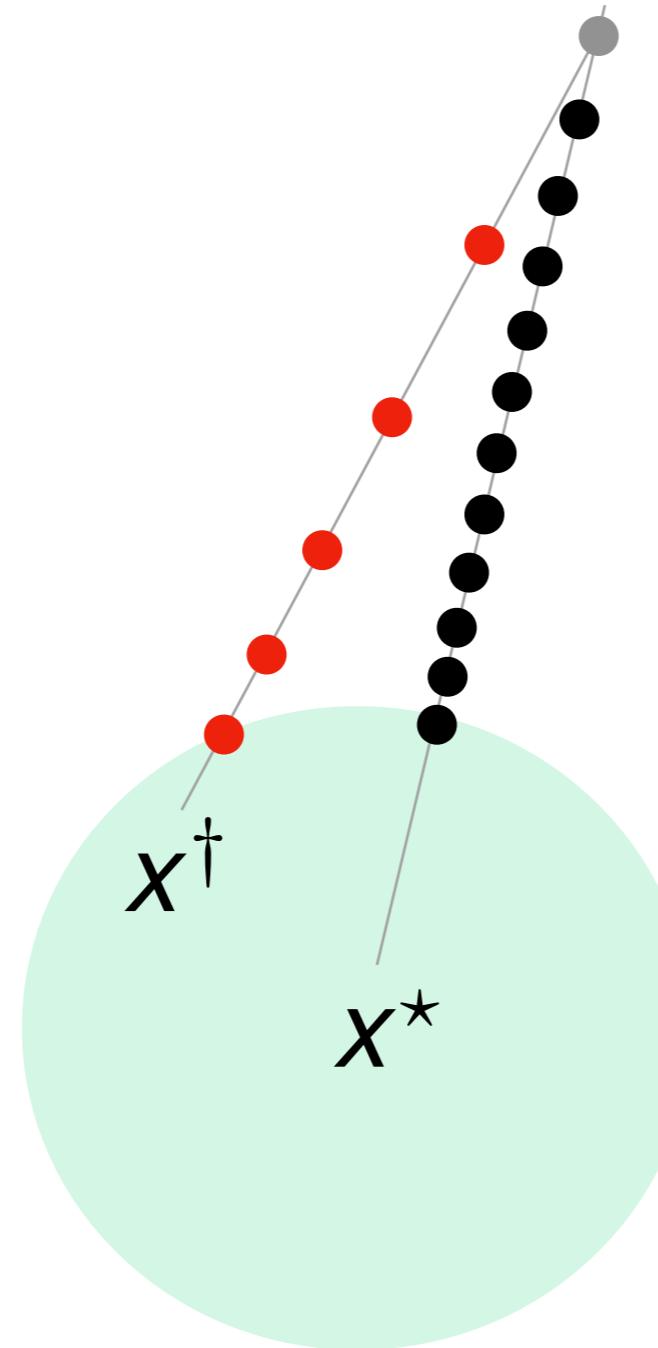
Results: logistic regression



Reaching epsilon-accuracy



Reaching epsilon-accuracy



Note:

Local GD:

$$O\left(\frac{L}{\mu} \frac{1}{H} \log\left(\frac{1}{\epsilon}\right)\right)$$

but

$$H = O(1 + \epsilon)$$



$$O\left(\frac{L}{\mu} \log\left(\frac{1}{\epsilon}\right)\right)$$

Analysis in the non-contractive case

- $t_n = nH$  convergence to x^\dagger , a fixed point of
$$\tilde{\mathcal{T}} = \frac{1}{M} \sum_{i=1}^M (\lambda \mathcal{T}_i + (1 - \lambda) \text{Id})^H$$
- sublinear rates on $\|\hat{x}^{(n+1)H} - \hat{x}^{nH}\|^2$ or $\|\hat{x}^k - \mathcal{T}(\hat{x}^k)\|^2$
- $t_n = nH$  convergence w.r.t. nb. epochs
1 to H times faster

Algorithm 2

Algorithm 2 Randomized distributed fixed-point method

Input: Initial estimate $\hat{x}^0 \in \mathbb{R}^d$, stepsize $\lambda > 0$, communication probability $0 < p \leq 1$

Initialize: $x_i^0 = \hat{x}^0$, for all $i = 1, \dots, M$

for $k = 1, 2, \dots$ **do**

for $i = 1, 2, \dots, M$ in parallel **do**

$$h_i^{k+1} := (1 - \lambda)x_i^k + \lambda\mathcal{T}_i(x_i^k)$$

end for

 Flip a coin and

with probability p **do**

 Communicate h_i^{k+1} to master, for $i = 1, \dots, M$

 At master node: $\hat{x}^{k+1} := \frac{1}{M} \sum_{i=1}^M h_i^{k+1}$

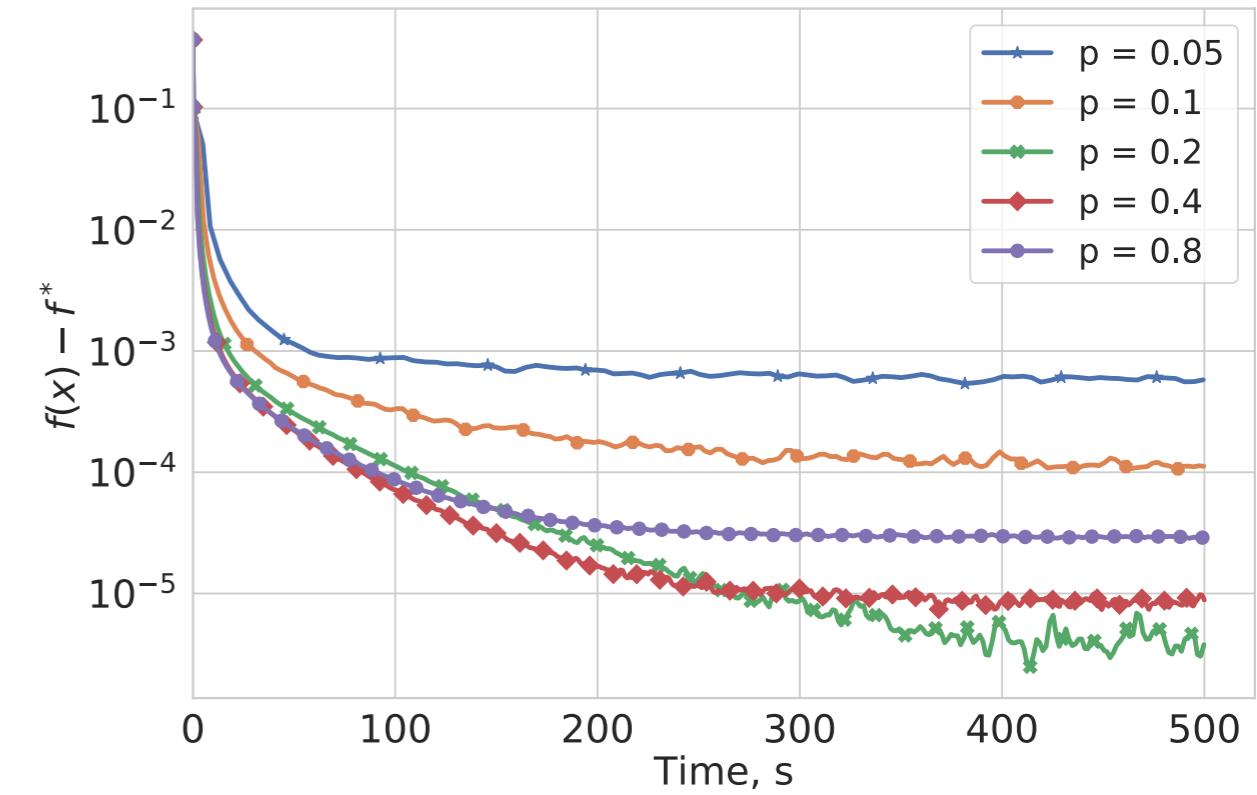
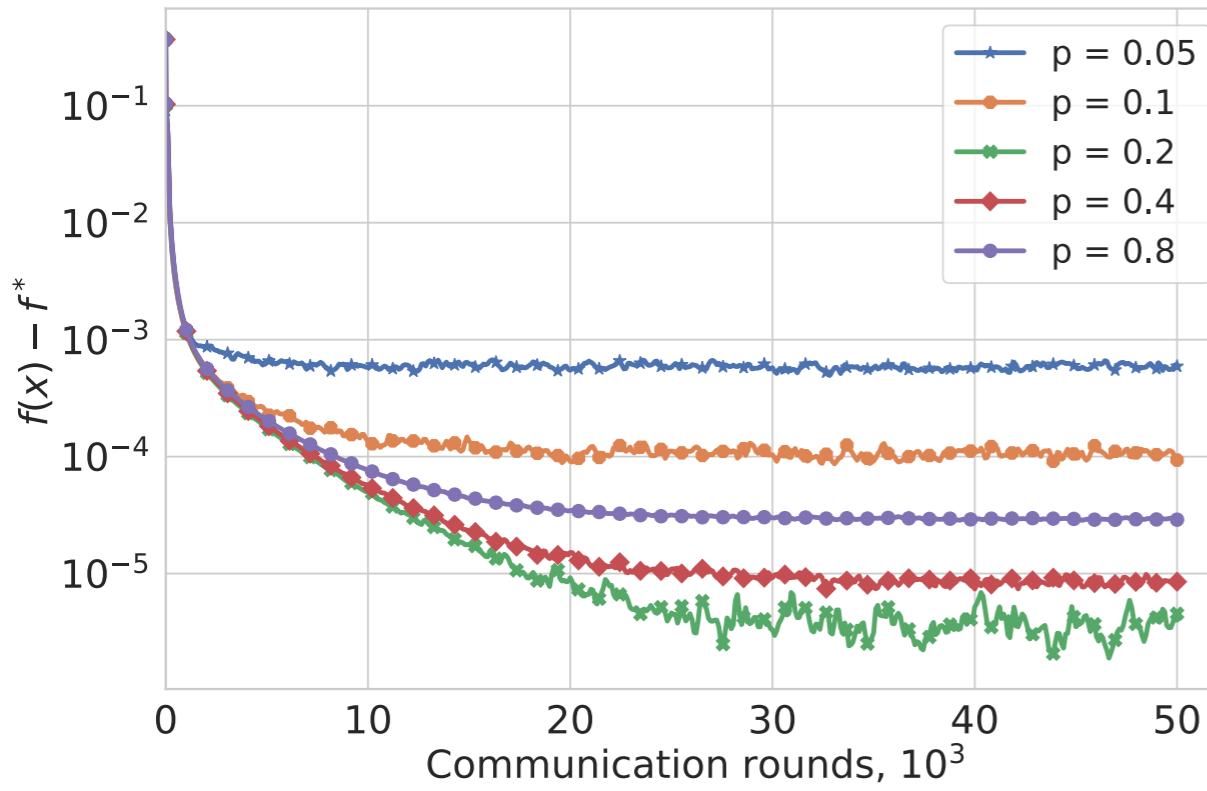
 Broadcast: $x_i^{k+1} := \hat{x}^{k+1}$, for all $i = 1, \dots, M$

else, with probability $1 - p$, **do**

$x_i^{k+1} := h_i^{k+1}$, for all $i = 1, \dots, M$

end for

Results: logistic regression



Analysis of Algorithm 2

Lyapunov function:

$$\psi^k := \|\hat{x}^k - x^*\|^2 + \frac{5\lambda}{p} \frac{1}{M} \sum_{i=1}^M \|x_i^k - \hat{x}^k\|^2$$

For λ small enough:

Theorem 3.2

$$\mathbb{E}[\psi^k] \leq \left(1 - \min\left(\frac{\lambda\rho}{1+\rho}, \frac{p}{5}\right)\right)^k \psi^0 + \frac{150}{\min\left(\frac{\lambda\rho}{1+\rho}, \frac{p}{5}\right) p^2} \frac{\lambda^3}{M} \sum_{i=1}^M \|x^* - \mathcal{T}_i(x^*)\|^2$$

Conclusion

Using local steps: provably good strategy
to achieve a medium-accuracy solution faster,
whenever communication is the bottleneck

Conclusion

Using local steps: provably good strategy
to achieve a medium-accuracy solution faster,
whenever communication is the bottleneck

+ one can decrease H along the iterations

Conclusion

Using local steps: provably good strategy to achieve a medium-accuracy solution faster, whenever communication is the bottleneck

Future work:

- * Stochastic fixed point operators...
- * Variance reduction??

Conclusion

Using local steps: provably good strategy
to achieve a medium-accuracy solution faster,
whenever communication is the bottleneck

