Collaborative Learning as an Agreement Problem

Sadegh Farhadkhani

Principles of Distributed Learning @ PODC 2022

EPFL

Based on:

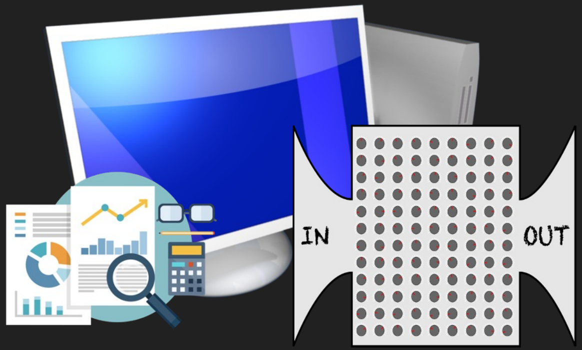
Collaborative Learning in the Jungle (fully decentralized, heterogeneous, Byzantine, asynchronous and nonconvex), NeurIPS 2021

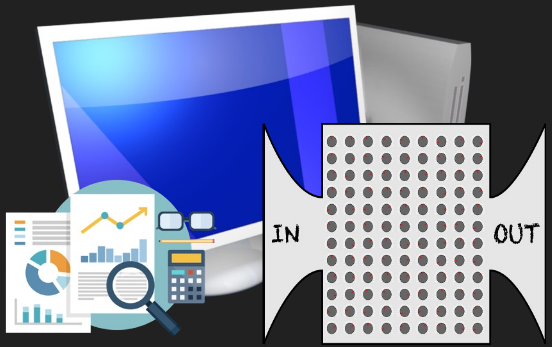
Joint work with:

El-Mahdi El-Mhamdi, Rachid Guerraoui, Arsany Guirguis,
Lê-Nguyên Hoang, Sébastien Rouault

ÉCOLE POLYTECHNIQUE
IP PARIS

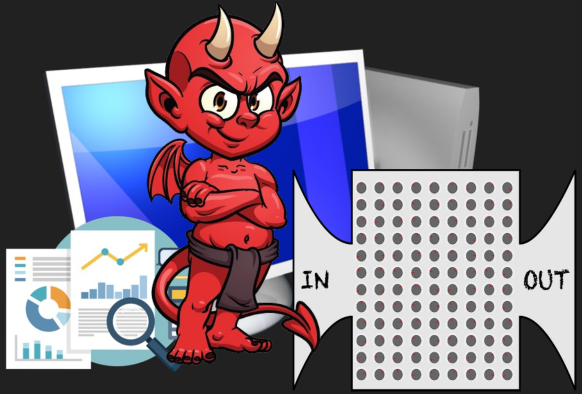
EPFL

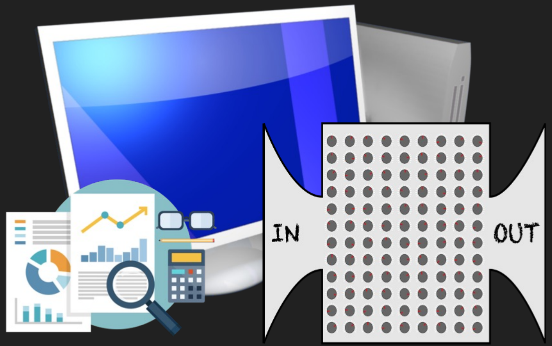




Nodes

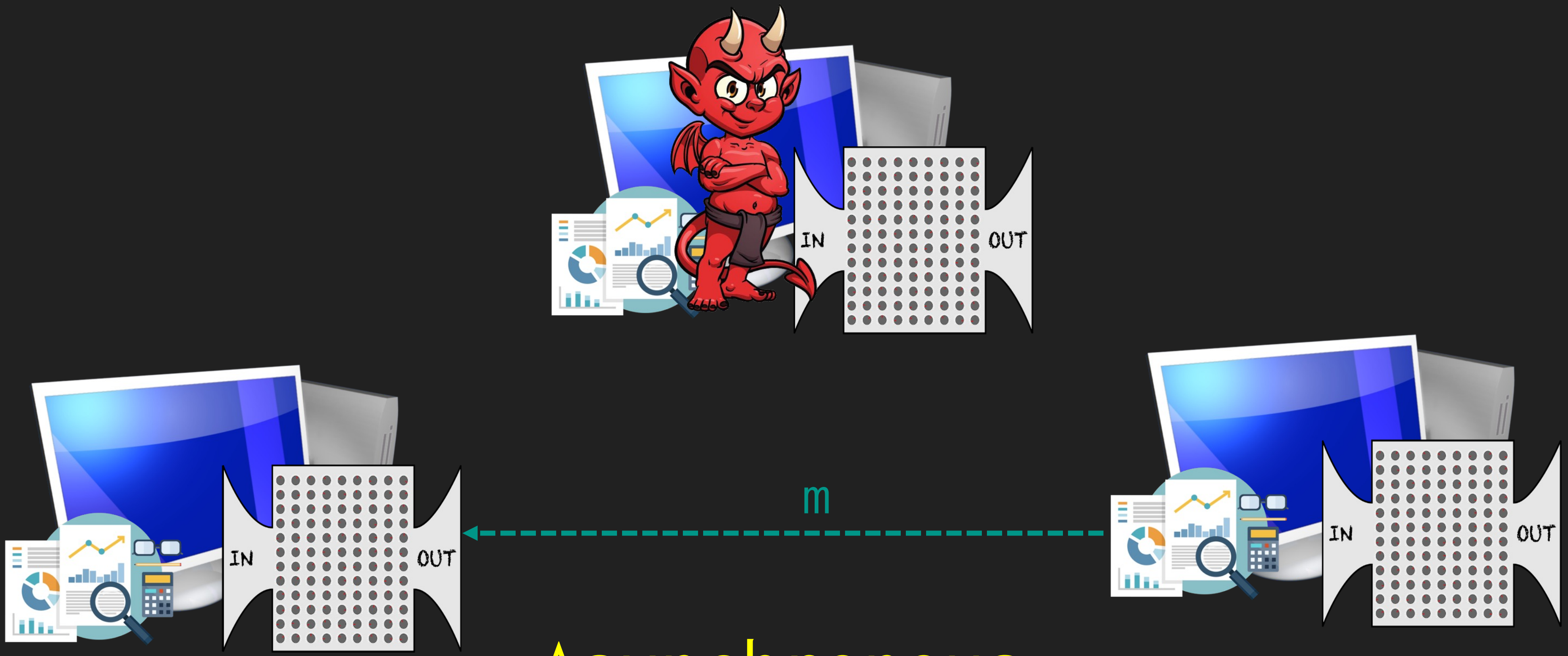
(compute and send gradient estimates, update and send models)





Byzantine

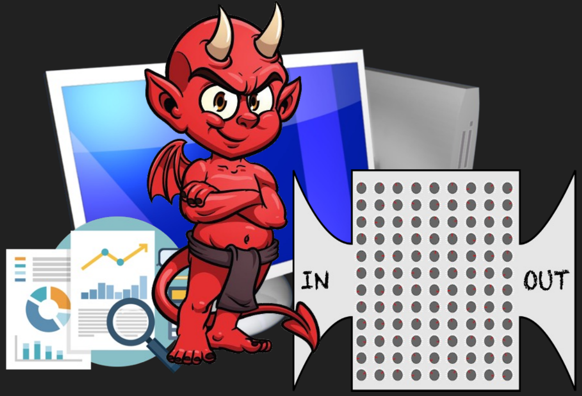
(at most f Byzantines out of n nodes)

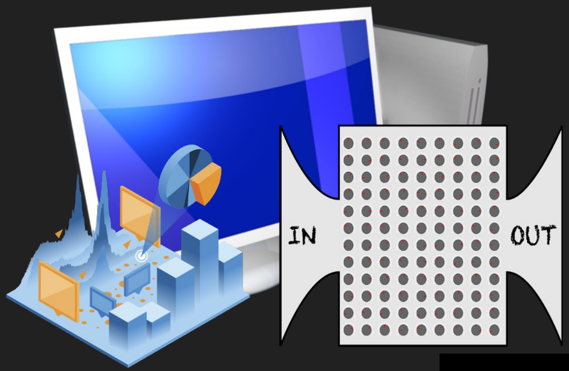


Asynchronous

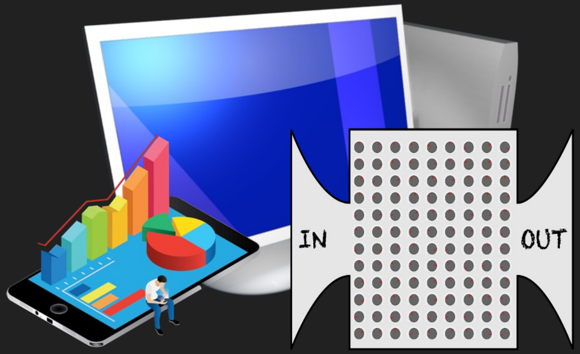
No bound on communication delays

No timing assumption



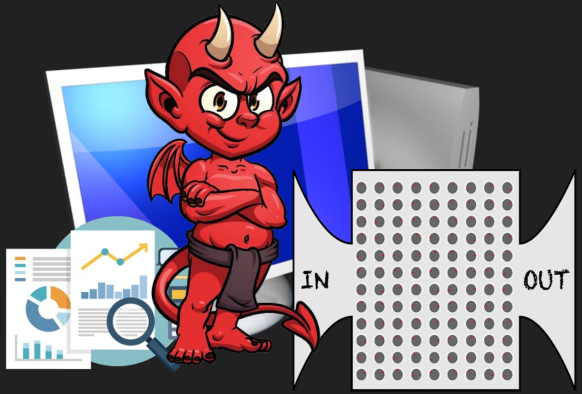


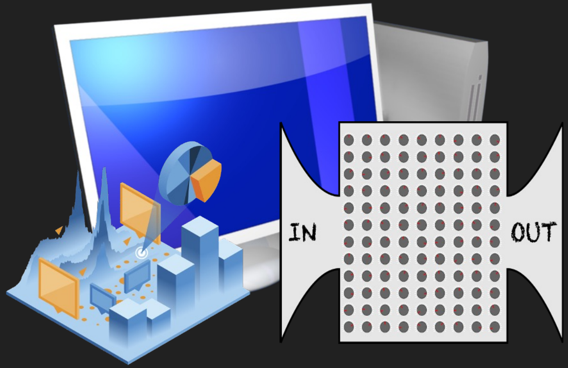
Heterogeneous data

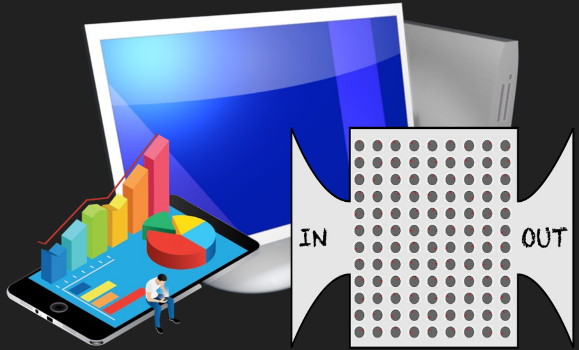


$$\nabla \mathcal{L}^{(j)}(\theta) \neq \nabla \mathcal{L}^{(k)}(\theta)$$

$$K := \sup_{j,k \in [n-f],\ \theta \in \mathbb{R}^d} \sup \left\| \nabla \mathcal{L}^{(j)}(\theta) - \nabla \mathcal{L}^{(k)}(\theta) \right\|_2$$

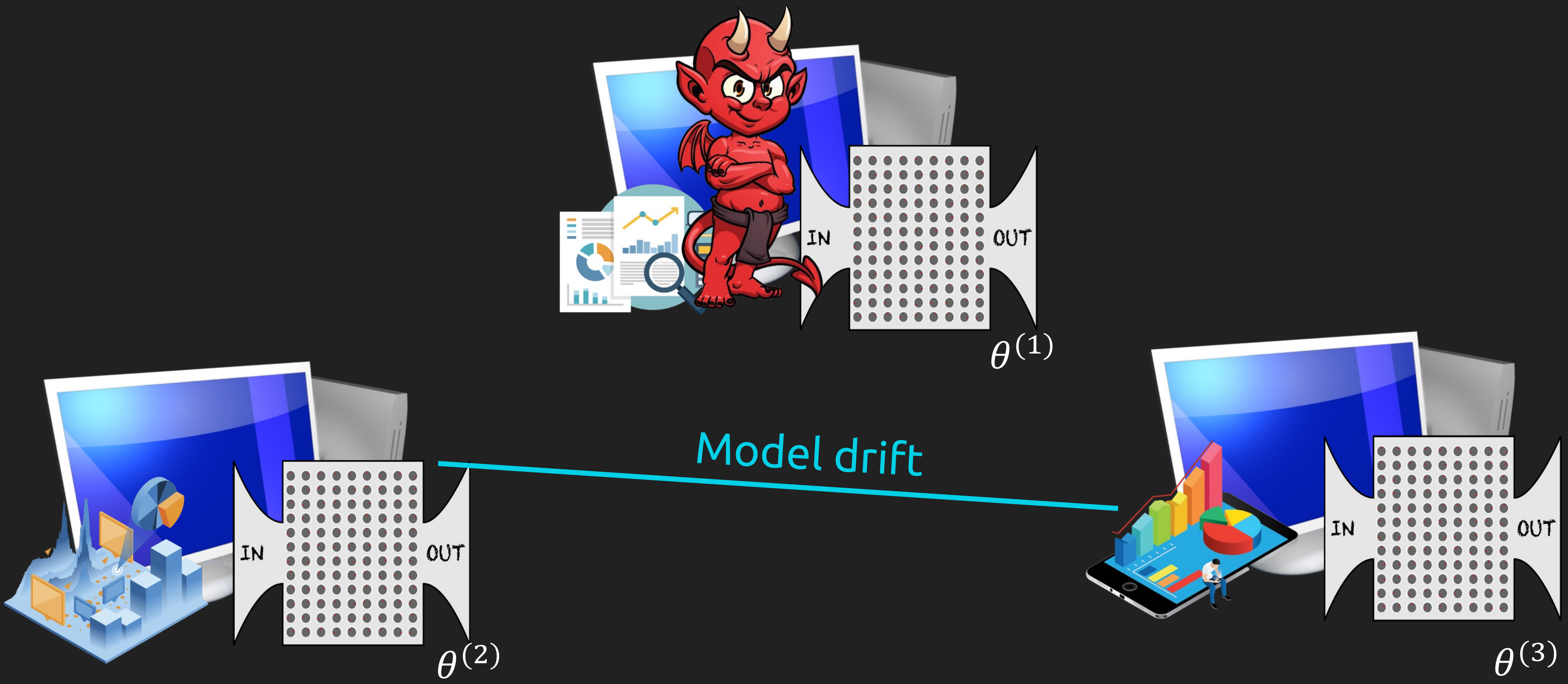






Byzantine, Asynchrony, nonconvex

Heterogeneous data



Byzantine, Asynchrony, nonconvex

Heterogeneous data

Definition

C-Collaborative learning is achieved if all honest nodes achieve approximate agreement and small enough gradient.

$$\Delta_2(\vec{\theta}) \leq \delta$$

$$||\nabla \bar{\mathcal{L}}(\bar{\theta})||_2 \leq (1+\delta)CK$$

{ $\Delta_2(\cdot)$ = diameter = maximum distance}

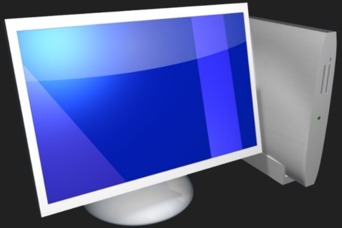
Theorem

Byzantine asynchronous nonconvex heterogeneous

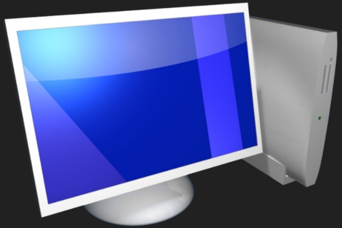
C-collaborative learning

is equivalent to

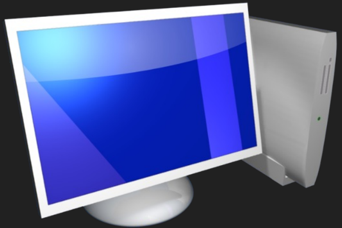
C-averaging agreement.



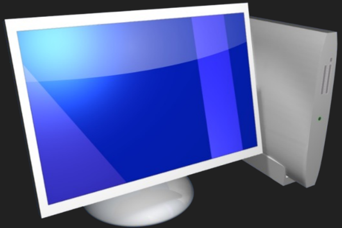
(0,1,3)



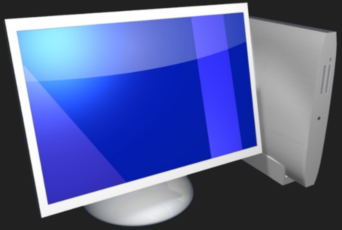
(2,3,3)



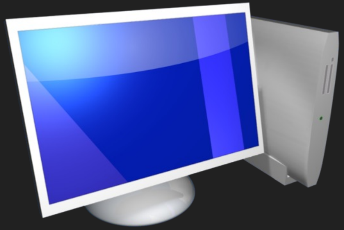
(7,2,6)



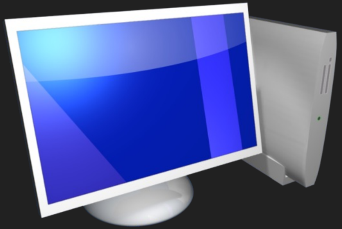
(1,6,5)



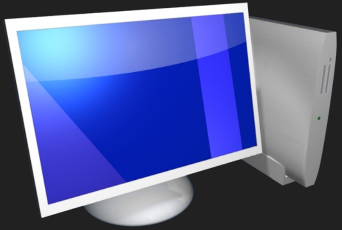
(0,1,3)



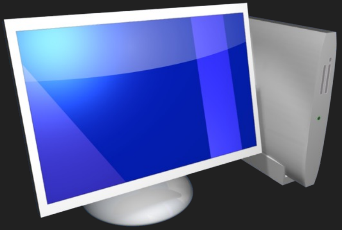
(2,3,3)



(7,2,6)



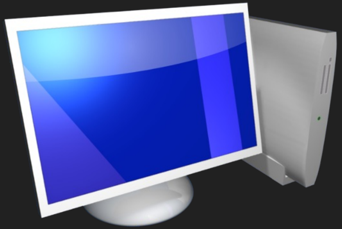
(0,1,3)



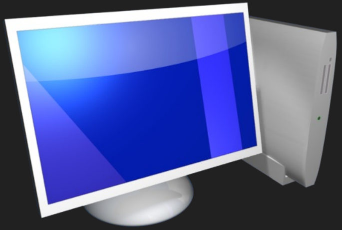
(2,3,3)

True average

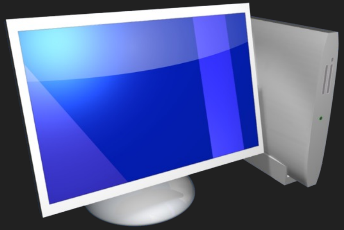
(3,2,4)



(7,2,6)



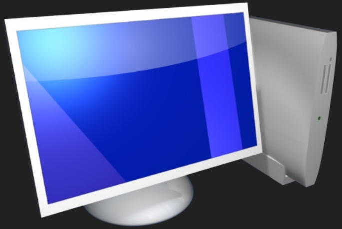
(0,1,3)
(2,2,3)



(2,3,3)
(2,3,3)

True average

(3,2,4)



(7,2,6)
(2,3,4)

Definition

C-Averaging agreement is achieved if
all honest nodes achieve approximate agreement
and estimate well the average.

$$\Delta_2(\vec{y}) \leq \delta$$

$$||\bar{x} - \bar{y}|| \leq C\Delta_2(\vec{x})$$

{ $\Delta_2(\cdot)$ = diameter = maximum distance}

Averaging-agreement

Similar to the classical approximate-agreement.

Without requiring the outputs to be in the convex hull.

Therefore, we do not need $n > (d+2)f$.

Theorem

Byzantine asynchronous nonconvex heterogeneous

C-collaborative learning

is (essentially) equivalent to

C-averaging agreement.

Theorem

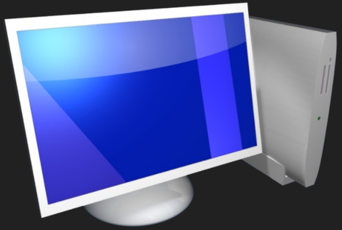
There is no solution to Byzantine asynchronous C-averaging agreement for $n \le 3f$, nor for $C < 2f/(n-f)$.

(n = #nodes, f = #Byzantines)

Corollary

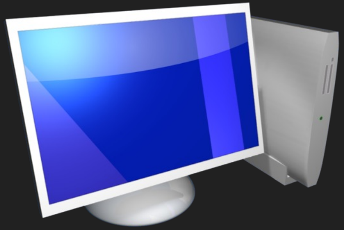
There is no solution to Byzantine asynchronous C-collaborative learning for $n \leq 3f$, nor for $C < 2f/(n-f)$.

(n = #nodes, f = #Byzantines)



(0,1,3)

(2,2,3)

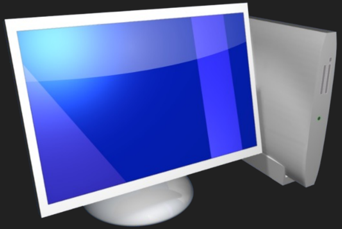


(2,3,3)

(2,3,3)

True average

(3,2,4)



(7,2,6)

(2,3,4)

Can we solve

Byzantine asynchronous

C-averaging?

Theorem

Coordinate-wise trimmed mean with reliable broadcasts solves averaging agreement for $n > 3f$ (optimal Byzantine resilience!!), with averaging constant $C = 4f/\sqrt{(n-f)}$.

Theorem

Minimum Diameter Averaging solves **averaging agreement** for $n \geq 6f+1$.
For $n >> f$, it achieves $C \sim 3f/(n-f)$
(quasi-optimal up to a factor 3/2 !!).

Corollary

SGD-modified + RB + ICwTM solves C-collaborative learning for $n > 3f$.

In the limit $n \gg f$, SGD-modified + MDA solves C-collaborative learning for $C \sim 3f/(n-f)$.

Conclusion

We solve decentralized, heterogeneous, Byzantine, asynchronous and nonconvex collaborative learning.

We show the equivalence between collaborative learning and averaging agreement.

We provide 2 algorithms, each is optimal in a different aspect.

Thank you!

Algorithm

Fix learning rate $\eta \triangleq \delta/12L$;

Fix number of iterations $T \triangleq T_{\textsc{Learn}}(\delta)$;

for $t \leftarrow 1, \ldots, T$ **do**

  $g_t \leftarrow \texttt{GradientOracle}(\theta_t, b_t)$;

  $\gamma_t \leftarrow \textsc{Avg}_{N(t)}(\vec{g}_t, \textsc{Byz})$ `// Vulnerable to Byzantine attacks`

  $\theta_{t+1/2} \leftarrow \theta_t - \eta\gamma_t$;

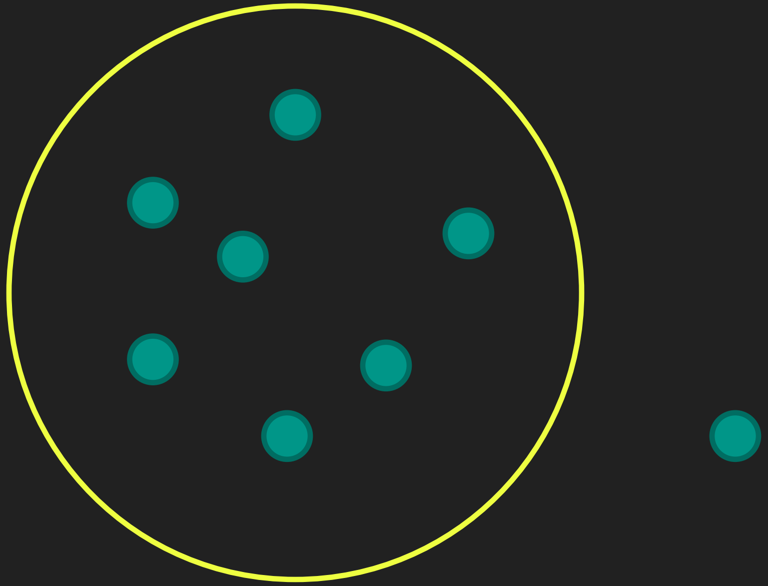
  $\theta_{t+1} \leftarrow \textsc{Avg}_1\left(\vec{\theta}_{t+1/2}, \textsc{Byz}\right)$ `// Vulnerable to Byzantine attacks`

end

Draw $* \sim \mathcal{U}([T])$ using the fixed common seed;

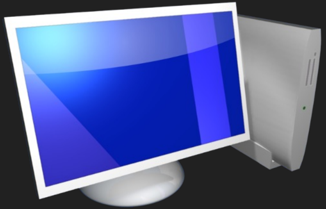
Return θ_*;

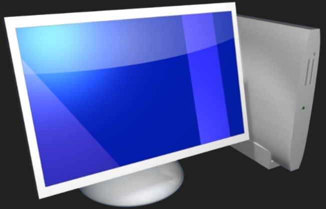
MDA



$f = 1$

Evaluation Setup

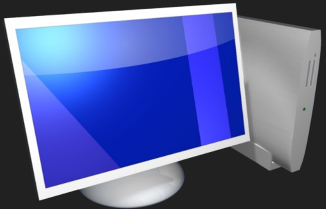




Our 4 algorithms vs. vanilla baseline

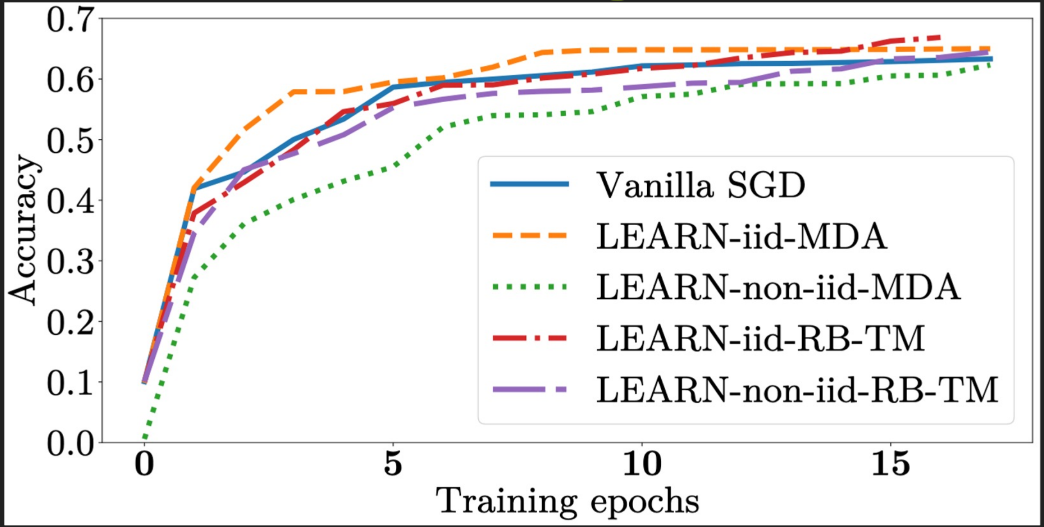
Garfield* - PyTorch

Image classification with f=1



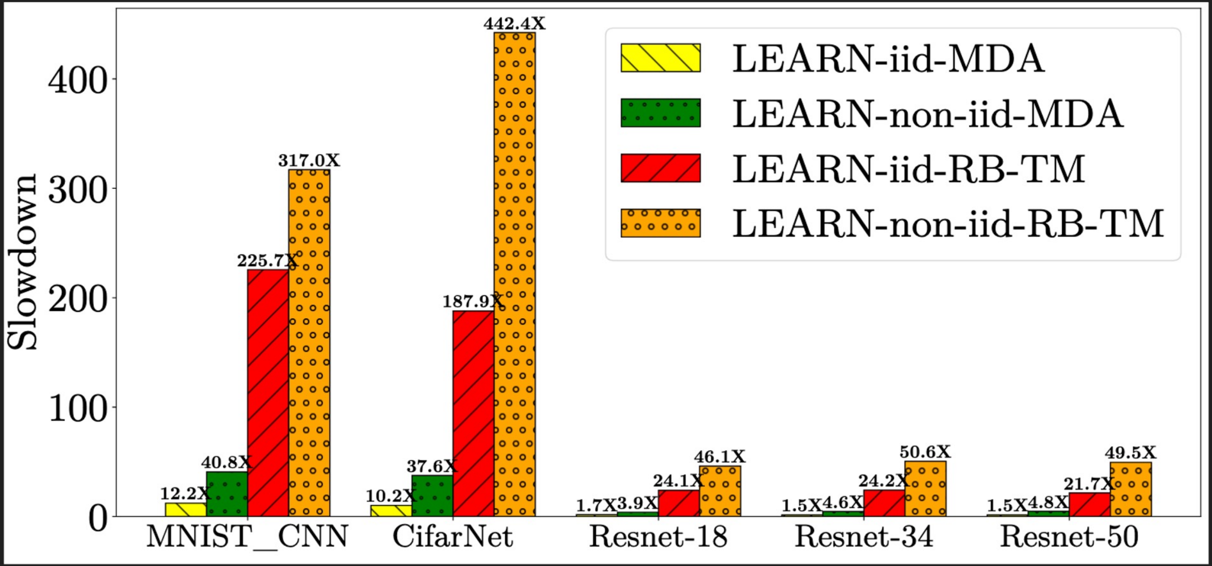
*https://github.com/LPD-EPFL/Garfield

Convergence



Our algorithms have similar behavior to our vanilla baseline.

Overhead



The more communication you require,
the higher price you pay.