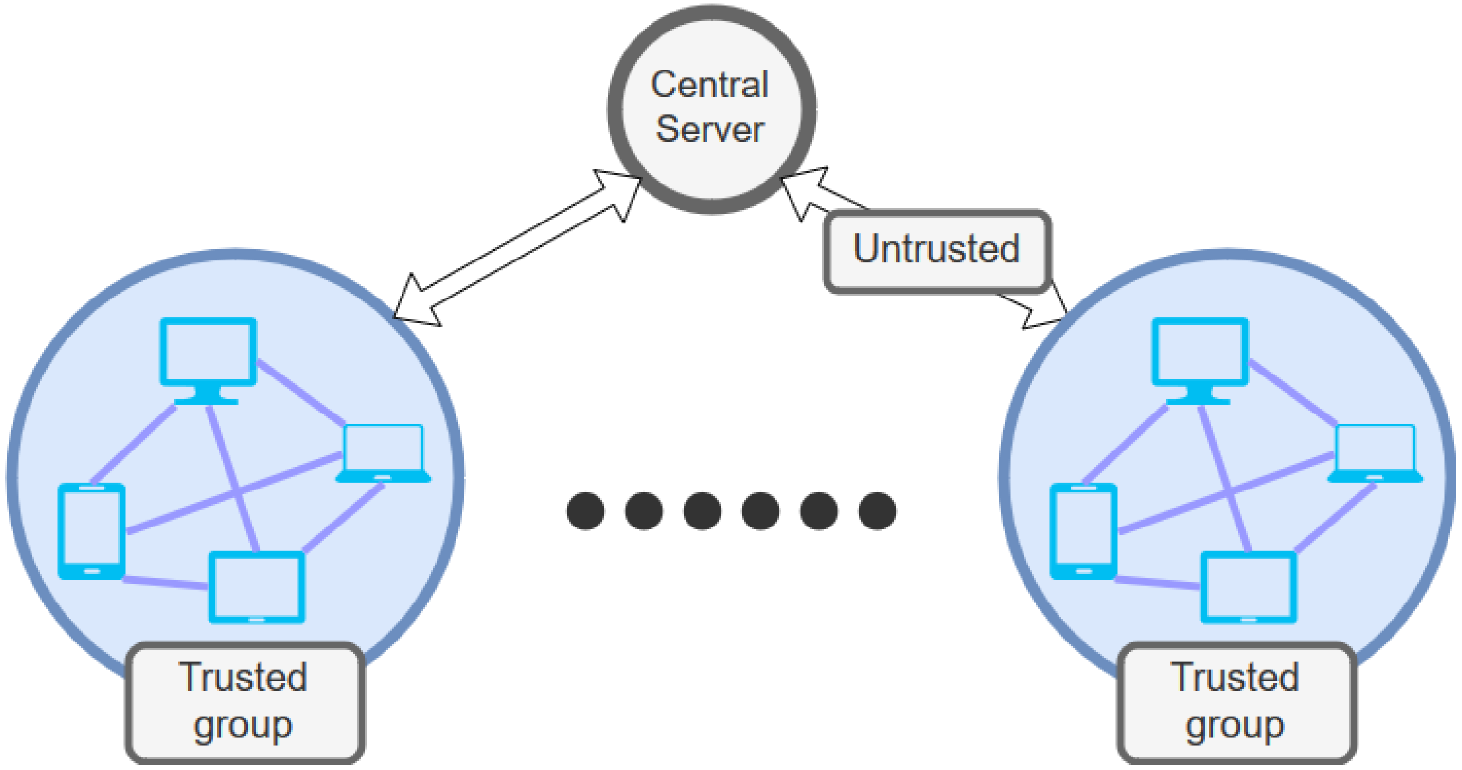
Fault-tolerant federated and distributed learning

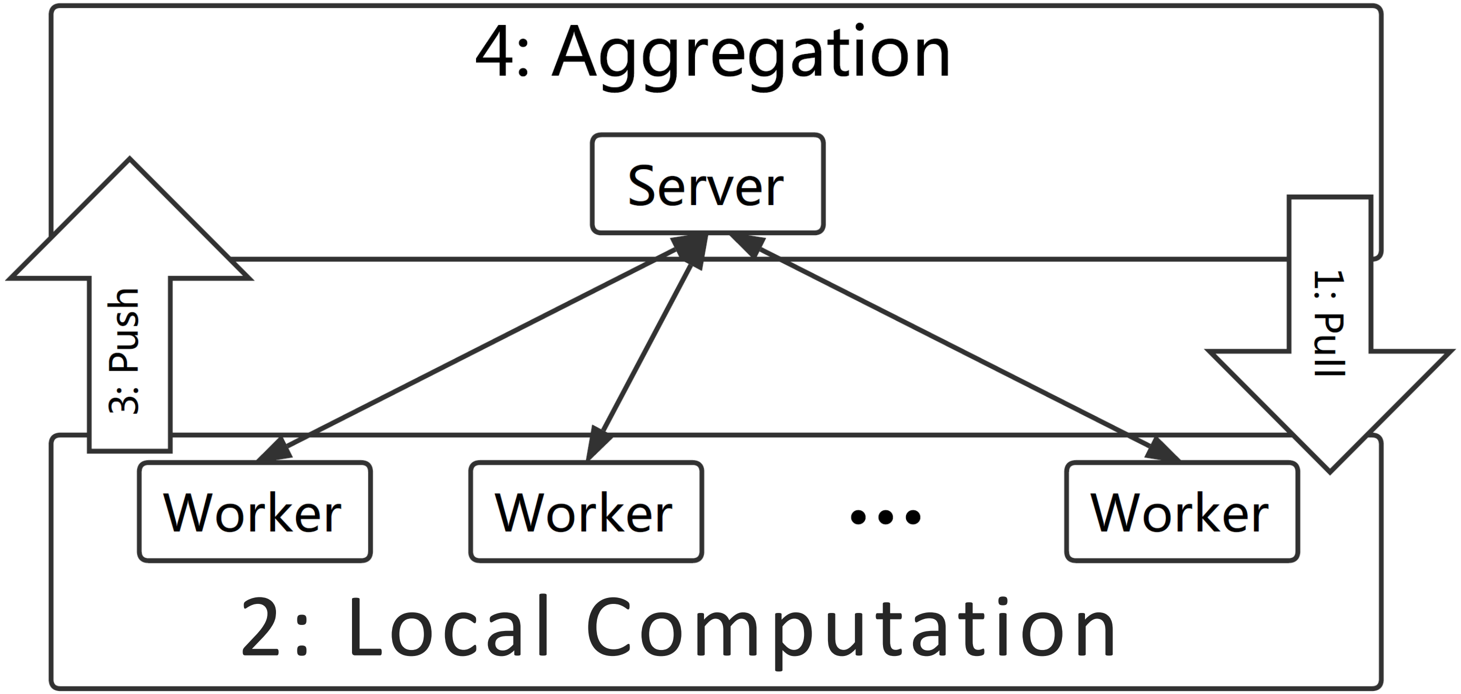
Sanmi Koyejo

Cong Xie

Indranil Gupta



- ML models routinely trained/deployed in distributed settings
- Distributed learning useful for amortizing training costs, learning with physically distributed data.
- Distributed learning has implications for privacy



Centralized Distributed Learning

Common strategies for distributed ML

1

Distributed Training
distributed gradient computation
server aggregates gradient updates

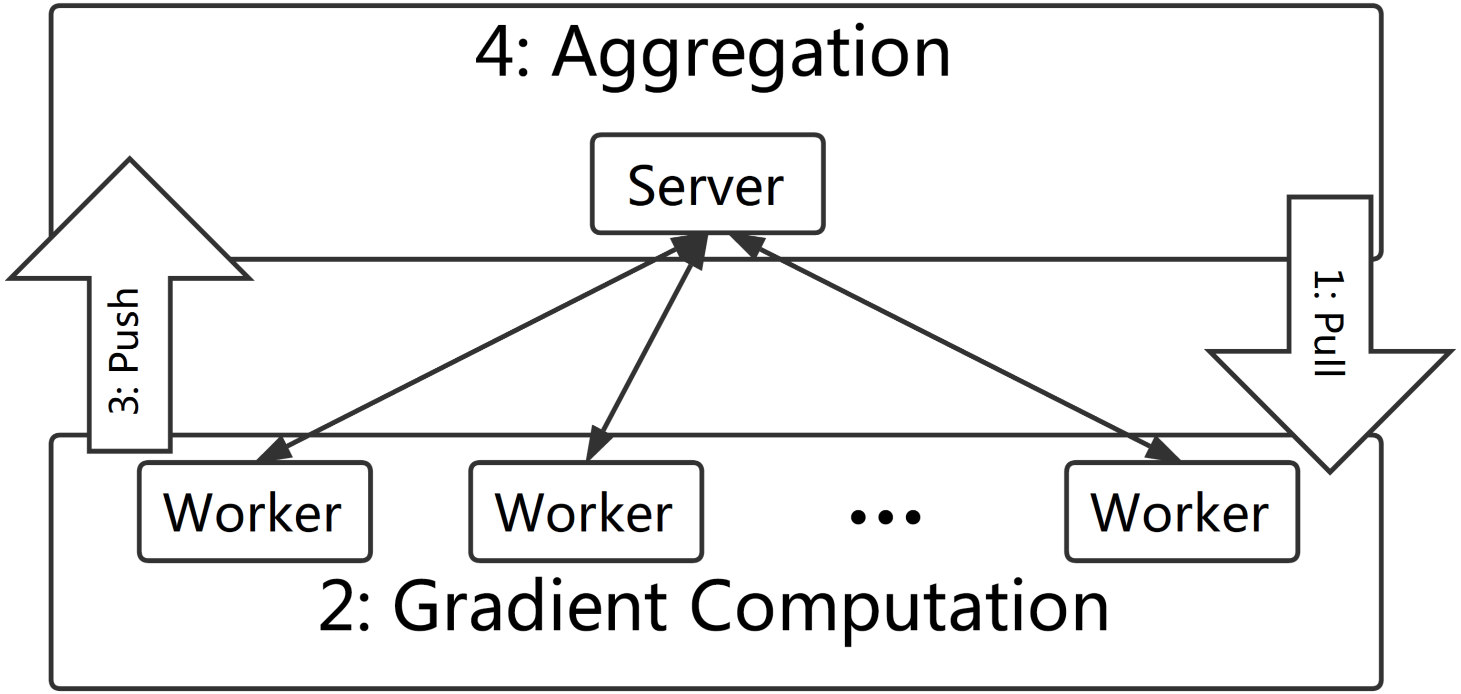
2

Federated Learning
distributed training on local data
server aggregates model parameters

Distributed ML is susceptible to failures

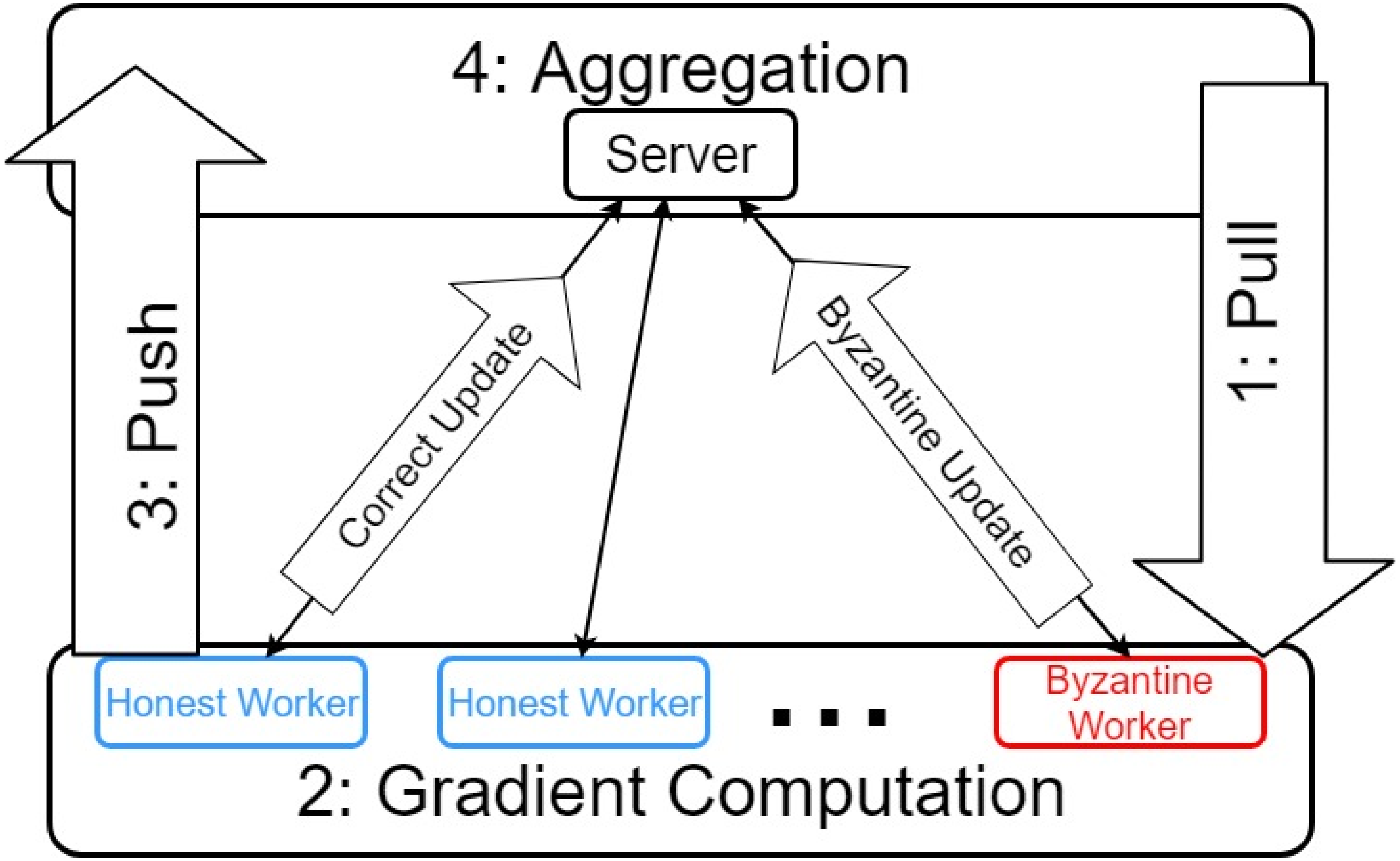
- Hardware failures e.g. bit-flip computation errors
- Software failures e.g. label-flip errors
- Communication failures e.g. dropped updates
- Adversarial attacks (worst case): possibly targeted, coordinated training attacks

Robust Distributed SGD



Workers compute gradients on local data

Threat Model



Distributed SGD

$\min_x F(x)$ where $F(x) = E_{z \sim \mathcal{D}}[f(x; z)]$

m workers, n samples per worker (wlog.) $F_i(x) = \frac{1}{n} \sum_{j=1}^{n} f(x; z^{i,j}), \forall i \in [m]$

Server update rule

$$x^{t+1} = x^t - \gamma^t \texttt{Aggr}(\{g_i(x^t) : i \in [m]\})$$

$$g_i(x^t) = \begin{cases} * & i\text{th worker is faulty,} \\ \nabla F_i(x^t) & \text{otherwise,} \end{cases}$$

Compared to prior work

Algorithm	Byzantine tolerance		Near-linear complexity $O(dm)$	Scalability
	$2q < m$	$m \leq 2q < 2m$		
Krum[1]	✓			
Trimmed mean[2] (median)	✓		✓	✓
Zeno (our work)	✓	✓	✓	✓

- m workers
- q malicious workers
- d dimensional feature

1. Blanchard et al. Machine learning with adversaries: Byzantine tolerant gradient descent. NIPS (2017).
2. Yin et al. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. ICML (2018).

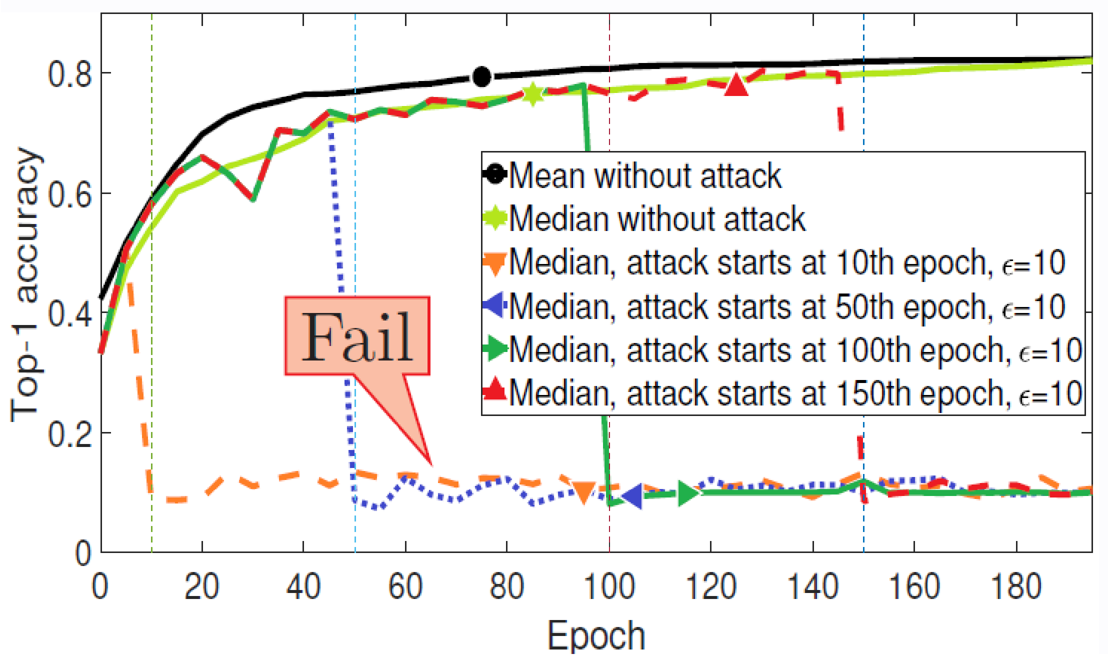
Important to focus on learning convergence, not generic robustness

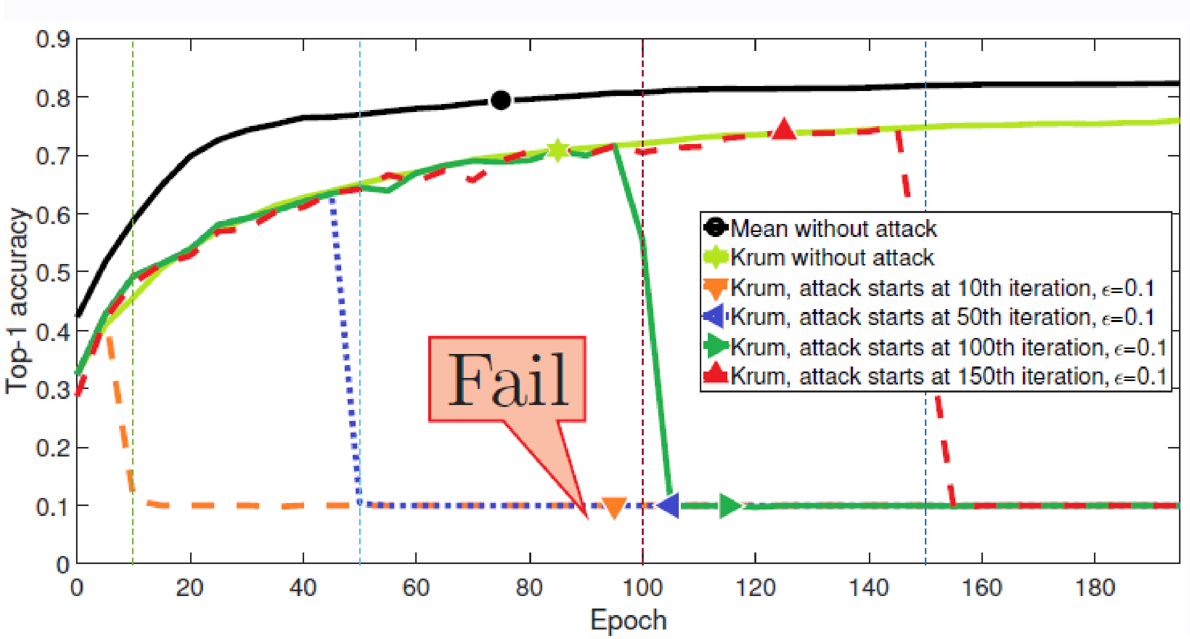
- Previous work on robust distributed learning (Median, Krum) has focused on Euclidean norm guarantees, roughly:

$$\|g_t - E[\nabla F_t(x)]\| < \epsilon$$

- Note that norm robustness is less important than robustly estimating the descent direction
- Example: construct an attacker that satisfies norm guarantees, but is pointed in the wrong direction

Xie, **K.**, Gupta "Fall of Empires: Breaking Byzantine-tolerant SGD by Inner Product Manipulation" (UAI 2019)





Breaking Robust Distributed Learning

Aggregation using Zeno

Key idea: Average the top-k gradients as sorted by ***stochastic descendant score***

$$Score_{\gamma,\rho}(u,x) = f_r(x) - f_r(x - \gamma u) - \rho\|u\|^2$$

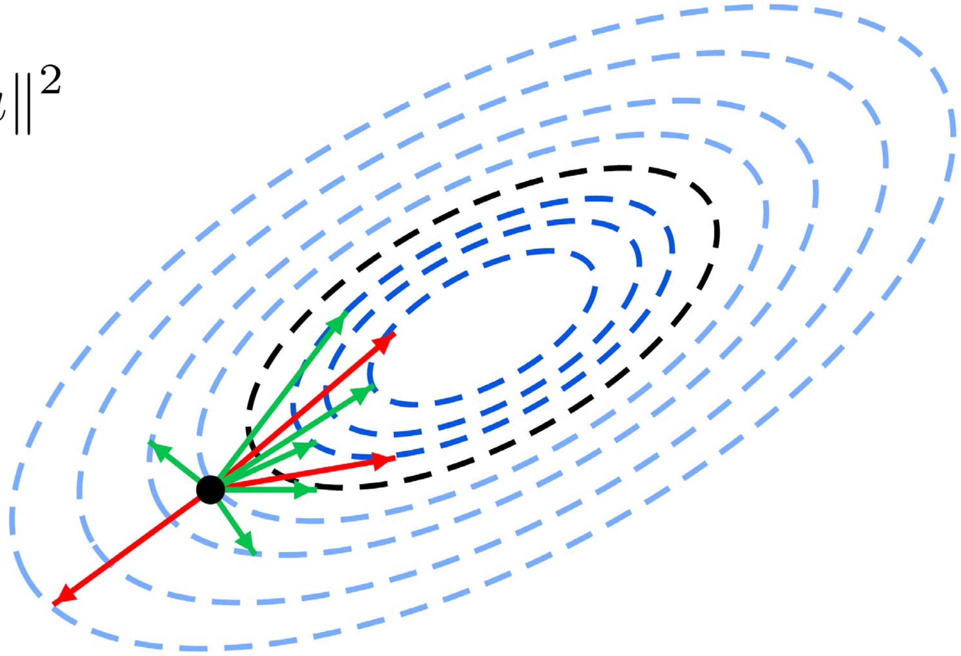
$$\text{where } f_r(x) = \frac{1}{n_r}\sum_{i=1}^{n_r} f(x; z_i)$$

●: current model

→:correct updates

→:incorrect updates

Intuition: Correct updates establish a boundary (black dashed circle); Zeno lies inside the boundary



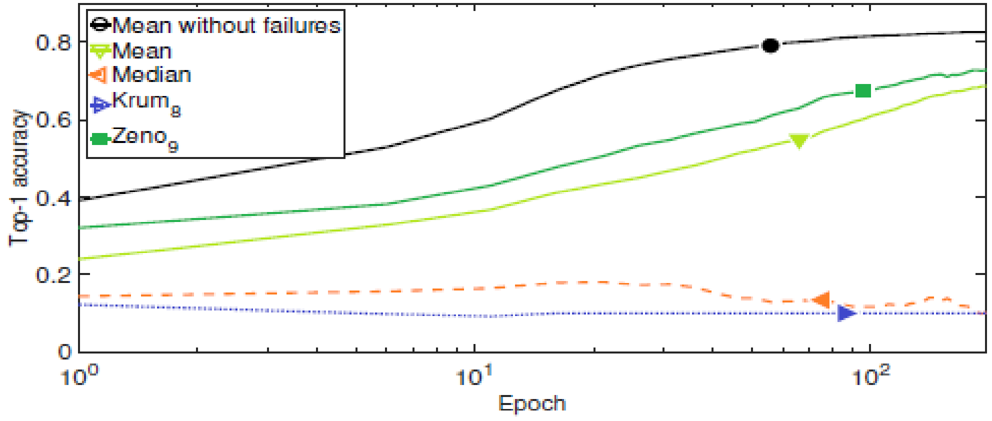
Zeno aggregation rule is robust

- Assumptions:
  - Stochastic descendant score estimate is unbiased
  - Loss function $f(x; z)$ is L-smooth and μ-weakly convex
  - Variance of population gradient is bounded

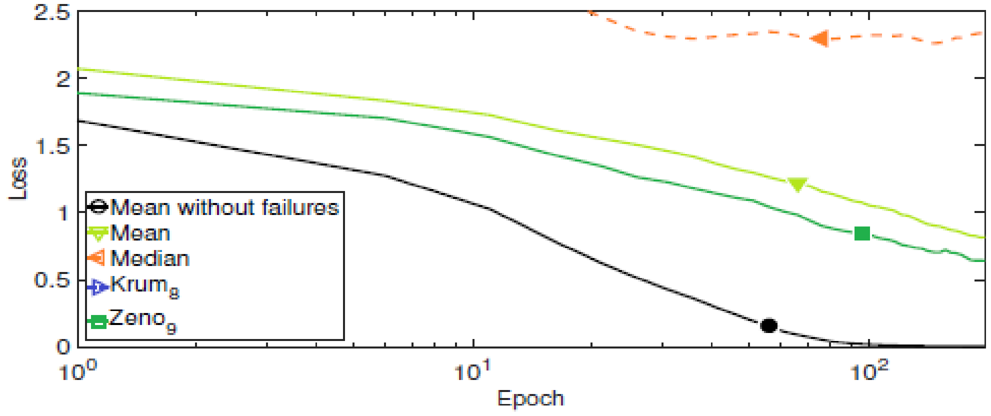
- Sketch of main result (with up to q failed / malicious workers)

$$\frac{\sum_{t=0}^{T-1} E\|\nabla F(x^t)\|^2}{T} \leq \mathcal{O}\left(\frac{1}{\sqrt{T}}\right) + \mathcal{O}\left(\frac{(k-q+1)(m-q)}{(m-k)^2}\right)$$

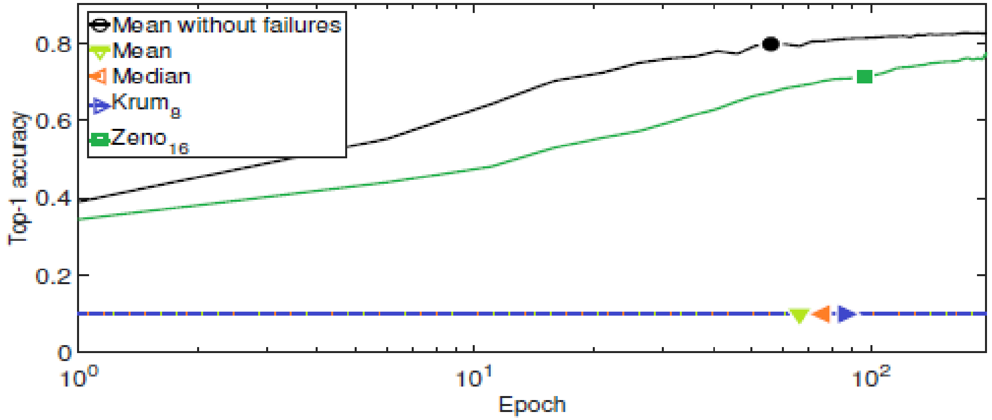
5-layer CNN, CIFAR-10, bit-flipping attack, m=20



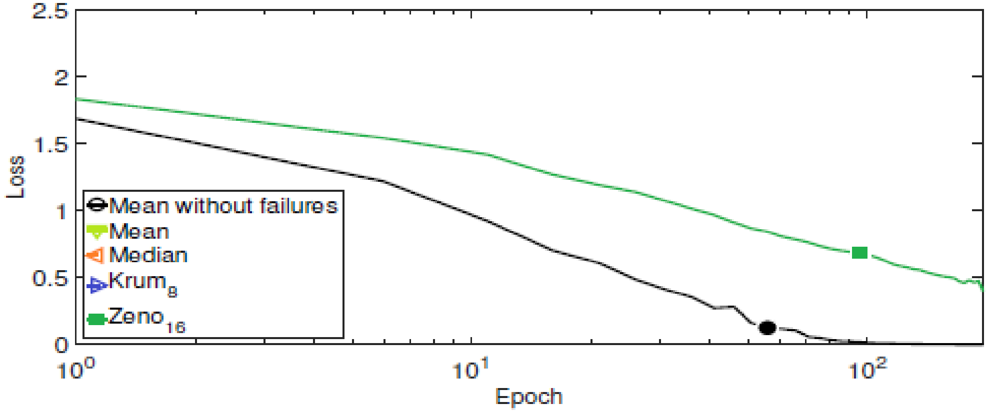
(a) Top-1 accuracy on testing set, with $q = 8$



(b) Cross entropy on training set, with $q = 8$

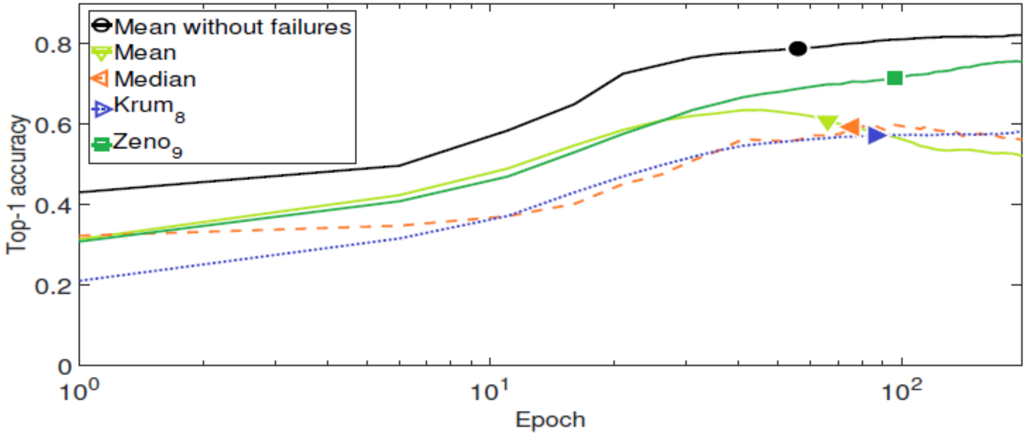


(c) Top-1 accuracy on testing set, with $q = 12$

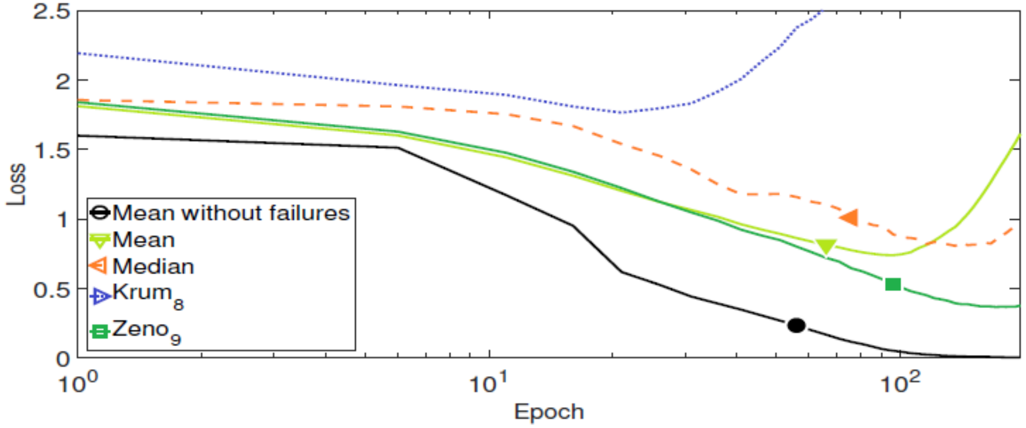


(d) Cross entropy on training set, with $q = 12$

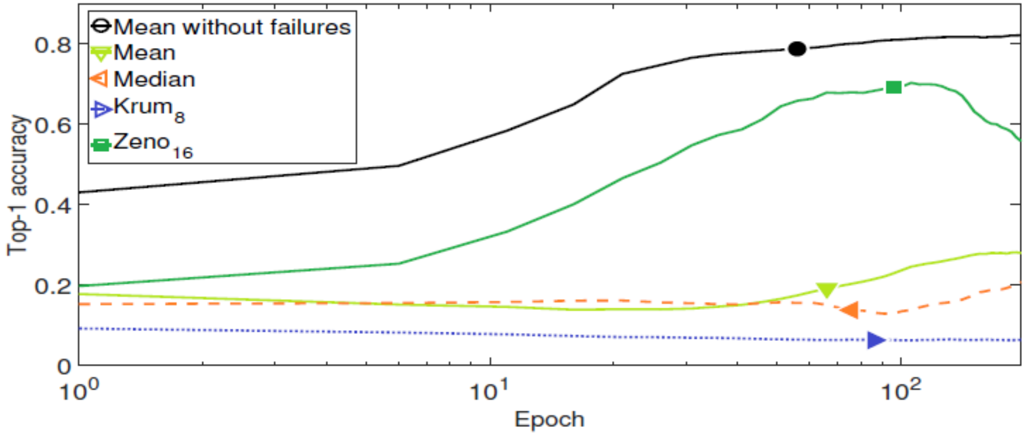
5-layer CNN, CIFAR-10, label-flipping attack, m=20



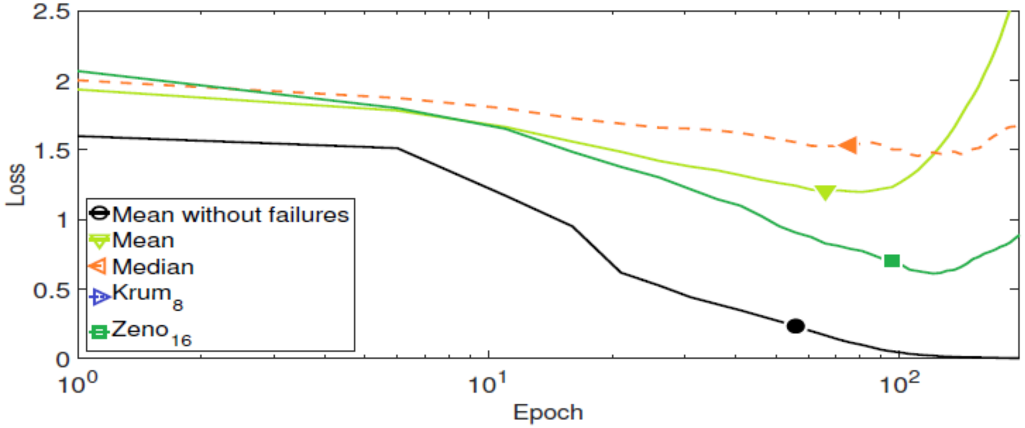
(a) Top-1 accuracy on testing set, with $q = 8$



(b) Cross entropy on training set, with $q = 8$



(c) Top-1 accuracy on testing set, with $q = 12$

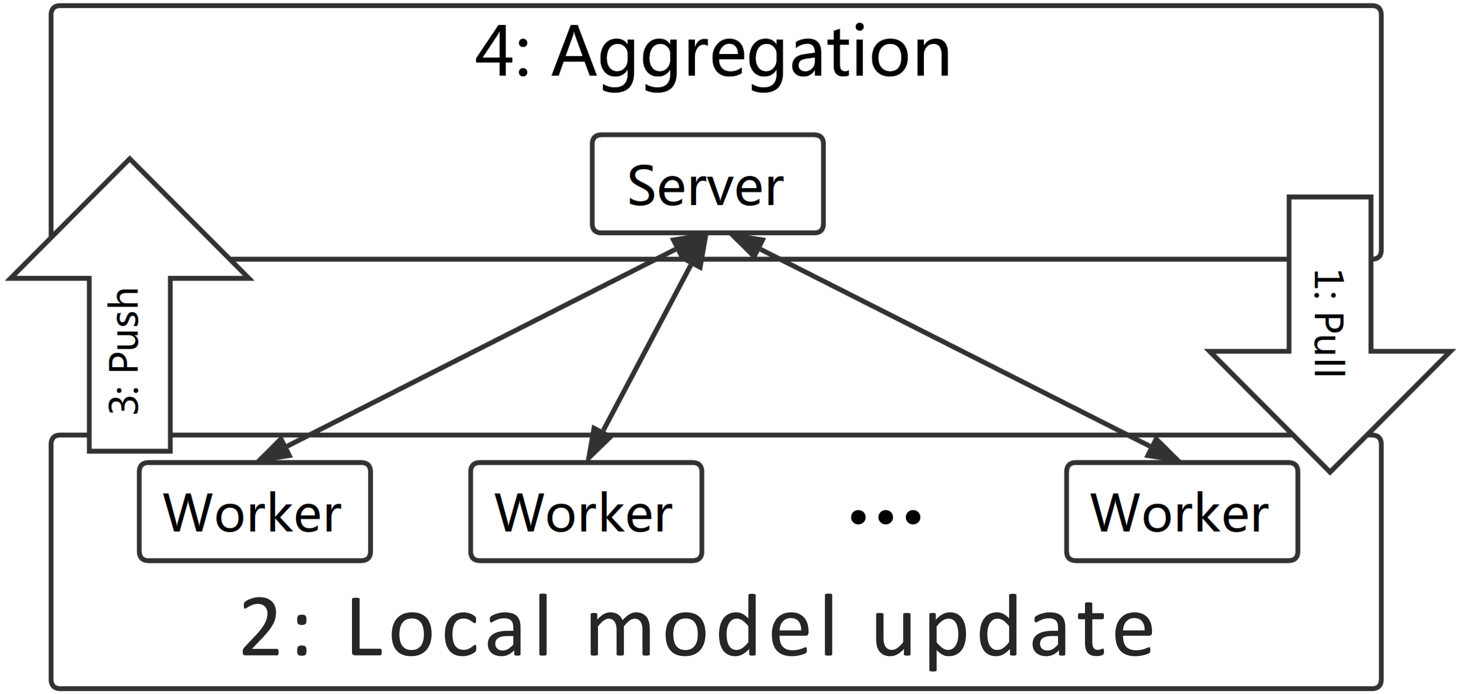


(d) Cross entropy on training set, with $q = 12$

Robust Federated Learning

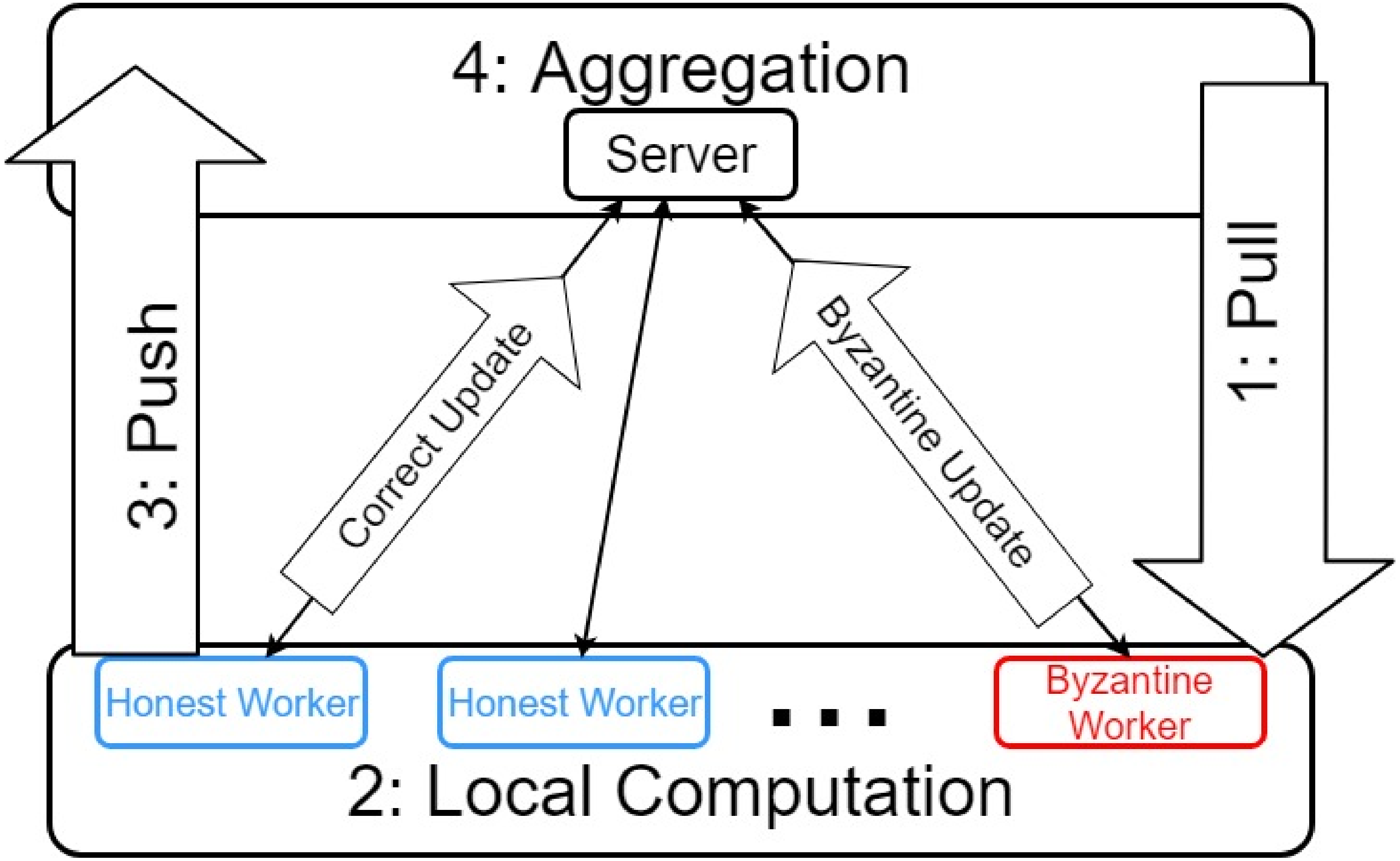
Is Federated Learning Simply Re-branded Distributed Learning?

- unbalanced, non-IID device data
- limited, heterogeneous device computation
- infrequent task scheduling
- limited, infrequent communication, congestion
- **untrusted devices and data poisoning**



Workers compute updated local model parameters

Threat Model



Compared to prior work

Key property	Solution	By
Limited computation	SGD	Previous work[1]
Limited communication	Dropped updates	
Private local data	Distributed (decentralized) training	
Hardware, Software, Communication failures, Poisoned workers	Robust estimator	Our work

1. McMahan, H. Brendan et al. Communication-Efficient Learning of Deep Networks from Decentralized Data. AISTATS (2017).

Federated Learning using Secure Local SGD

$\min_x F(x)$ where $F(x) = \frac{1}{n}\sum_{i\in[n]} E_{z^i\sim\mathcal{D}^i}[f(x;z^i)]$

Device update: $x^i_{t,h} \leftarrow x^i_{t,h-1} - \gamma\nabla f(x^i_{t,h-1}; z^i_{t,h})$ [for H steps]

Server update: $x'_t = \texttt{Trmean}_b\left(\left\{x^i_{t,H} : i \in S_t\right\}\right);$
$x_t \leftarrow (1-\alpha)x_{t-1} + \alpha x'_t$

$\texttt{Trmean}_b(\{u_i : i\in[l]\}) = \frac{1}{l-2b}\sum_{i=b+1}^{l-b} u_{\pi(i):\pi(l)}$

$\pi(\cdot) = \mathrm{argsort}(\cdot)$ $S_t =$ random subset of devices, $|S_t| = k$

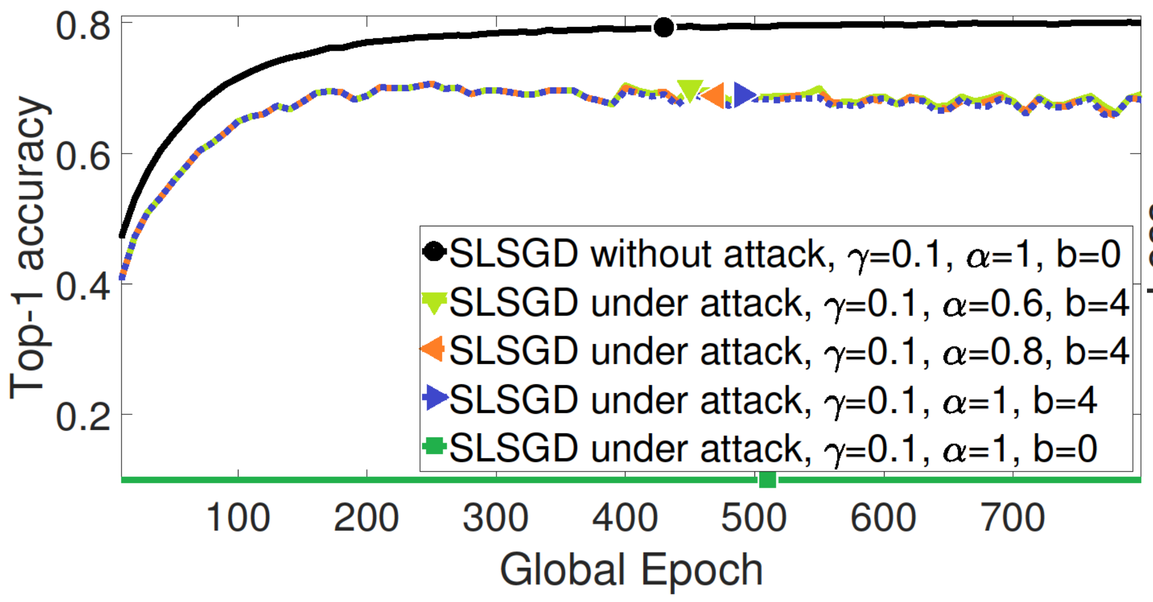
Proposed aggregation rule is robust

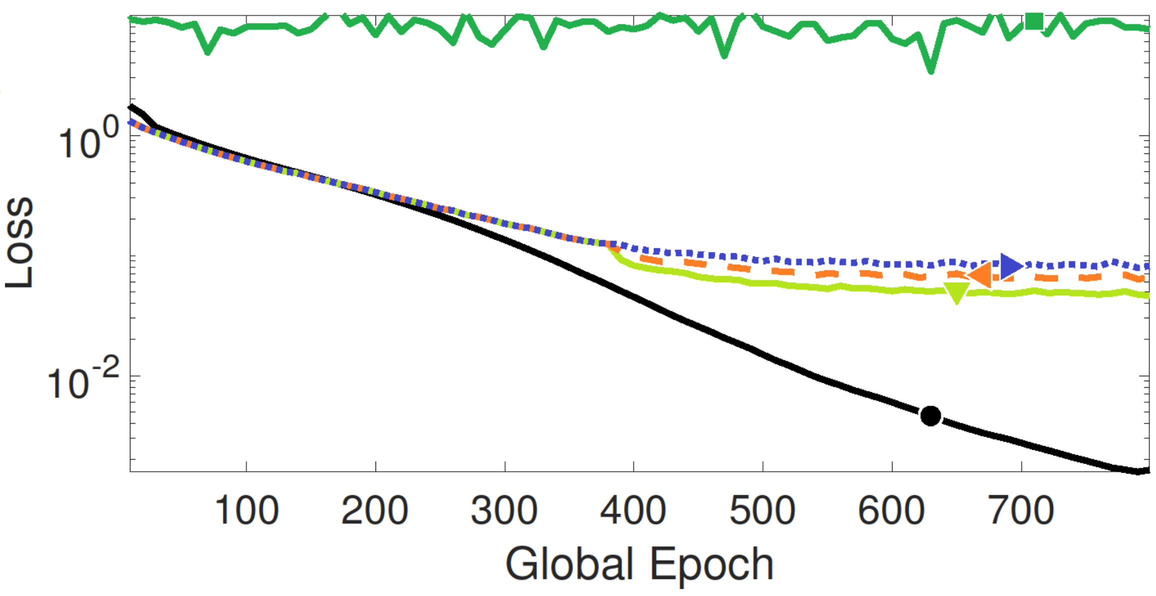
- Assumptions:
  - Existence of at least one global optimum (not necessarily unique)
  - Loss function $f(x;z)$ is L-smooth and μ-weakly convex
  - Variance of population gradient is bounded by V_1

- Sketch of main result: With up to q failed/malicious devices, Federated learning convergence rate

$$\frac{\sum_{t=0}^{T-1} E\|\nabla F(x^t)\|^2}{T} \leq \mathcal{O}\left(\frac{\frac{k(k+b)}{(k-b-q)^2}+\frac{1}{k-q}-\frac{1}{n}}{T}\right)+\mathcal{O}\left(V_1\right)$$

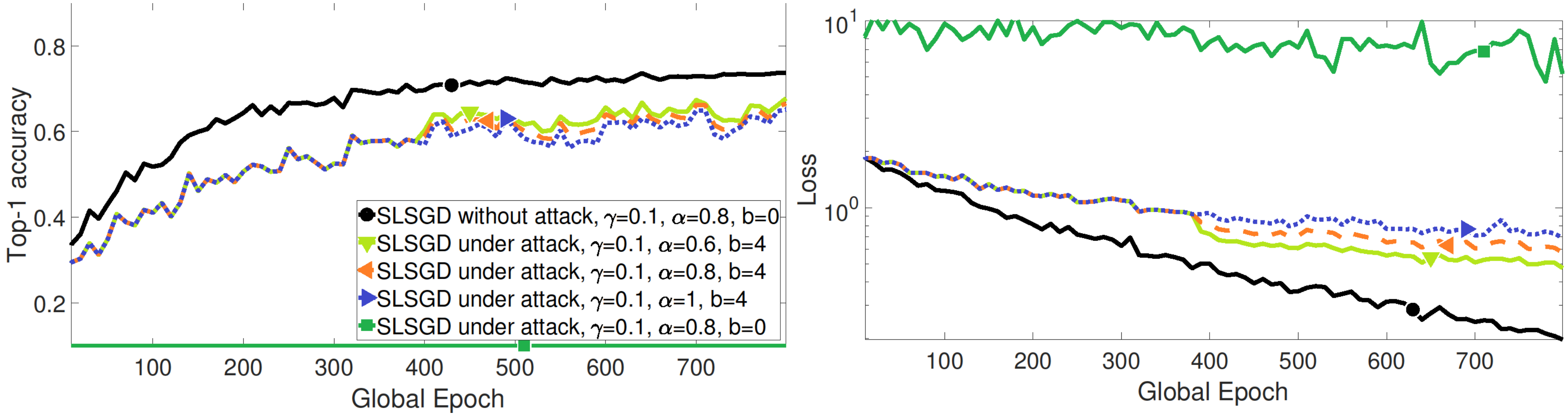
5-layer CNN, CIFAR-10; Balanced data
100 workers; k=10; label-flipping attack; q=4 (per)





NOTE: SLSGD is equiv. to *FedAvg* when α = 1; b=0.

5-layer CNN, CIFAR-10; Unbalanced data
100 workers; k=10; label-flipping attack; q=4 (per)



NOTE: SLSGD is equiv. to *FedAvg* when α = 1; b=0.

Careful aggregation is robust to worst-case failures

1

Suspicion-based aggregation for **distributed SGD**; robust to more than half adversarial workers

2

Regularized trimmed mean aggregation for **federated learning**; robust to non-IID data, communication failures, adversarial devices

Papers presented today

Xie, C., Koyejo, O., & Gupta, I.
Zeno: Byzantine-suspicious stochastic gradient descent. ICML 2019
arXiv:1805.10032

Xie, C., Koyejo, O., & Gupta, I.
SLSGD: Secure and Efficient Distributed On-device Machine Learning.
In ECML PKDD 2019. arXiv: 1903.06996

Some more light reading…

Xie, C., Koyejo, O., & Gupta, I. Zeno++: Robust Asynchronous SGD with an Arbitrary Number of Byzantine Workers (2019). arXiv:1903.07020

Xie, C., Koyejo, S., & Gupta, I. Fall of Empires: Breaking Byzantine-tolerant SGD by Inner Product Manipulation. In UAI 2019. arXiv:1903.03936

Xie, C., Koyejo, S., & Gupta, I. Generalized Byzantine-tolerant SGD (2018). arXiv:1802.10116

Asynchronous Federated ML

Worker Side

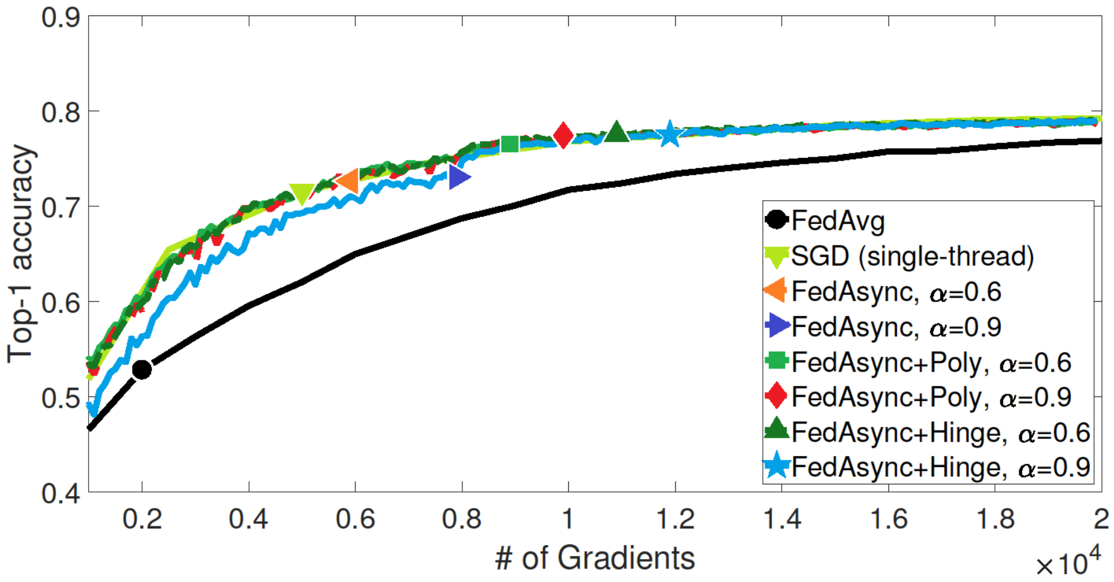
- Update local model using SGD on local loss regularized by global model

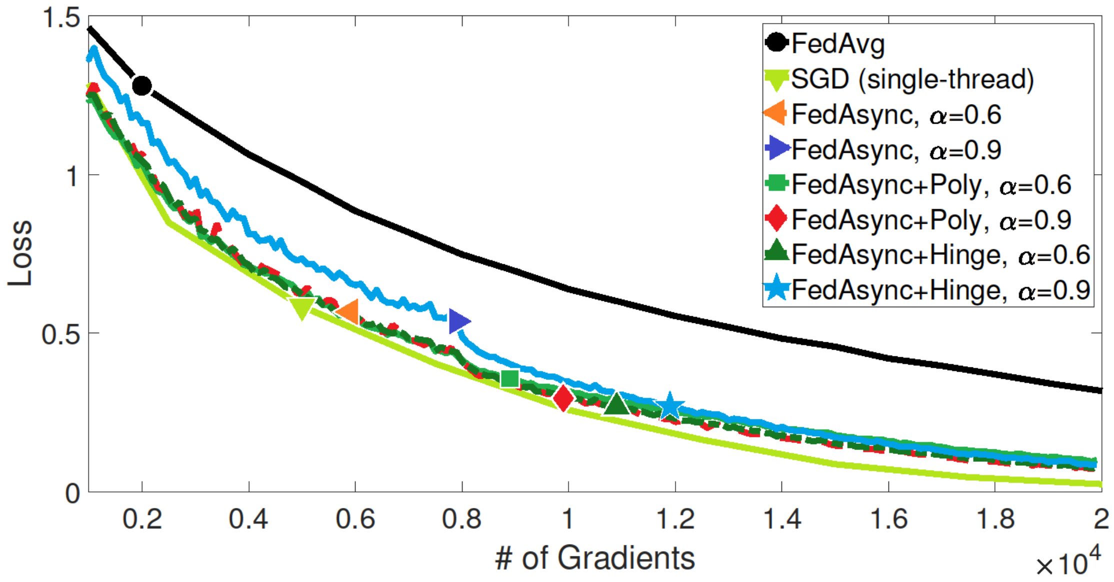
Server Side

- Scheduler thread to periodically trigger workers
- Update global model when updates received, with a discount factor proportional to staleness

Taken together, optimizes federated objective yet remains robust to delays, non-IID data, …

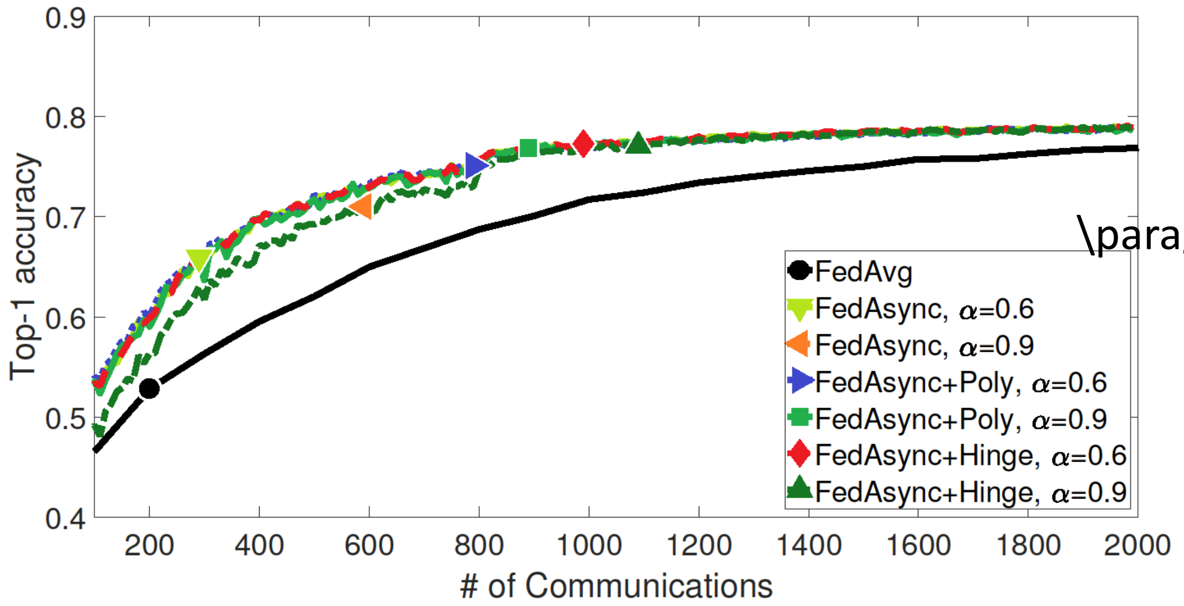
5-layer CNN, CIFAR-10; 100 workers



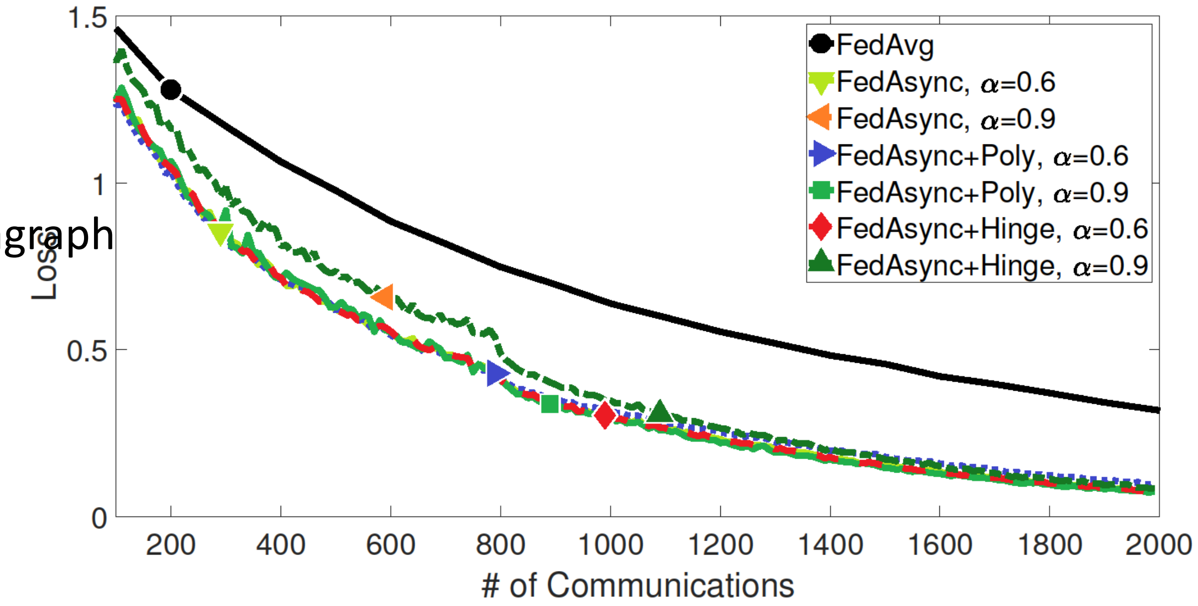


Performance vs # Gradients Max staleness of 4,with *Poly* and *Hinge* temporal smoothing

5-layer CNN, CIFAR-10; Unbalanced data
100 workers; k=10; label-flipping attack; q=4 (per)



\paragraph



Performance vs # communication Max staleness of 4,with *Poly* and *Hinge* temporal smoothing

Thank you

sanmi@Illinois.edu

@sanmikoyejo