Navigating Byzantine Attacks in Distributed Learning: Trial Function Methodology

ICOMP 2024

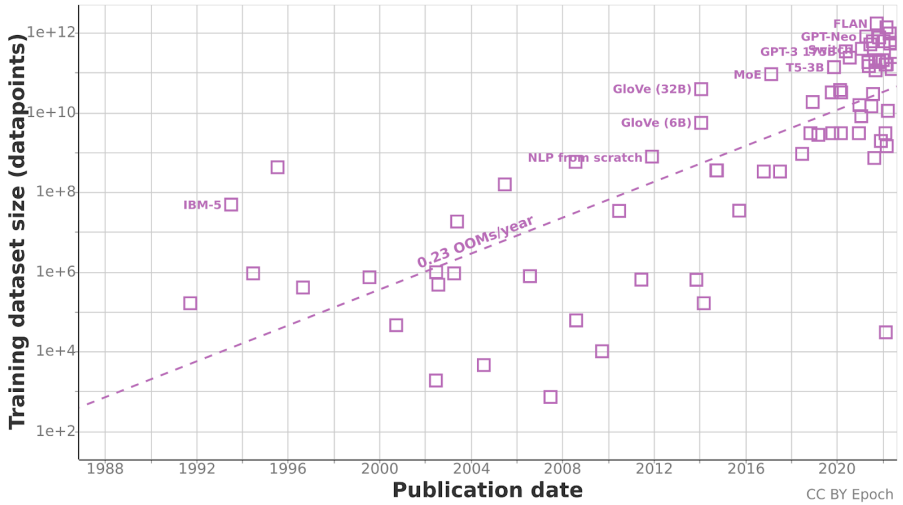
Gleb Molodtsov, Daniil Medyakov, Nikolas Khachaturov, Sergey Skorik, Shahane Tigranyan, Aram Avetisyan, Martin Takac, Aleksandr Beznosikov

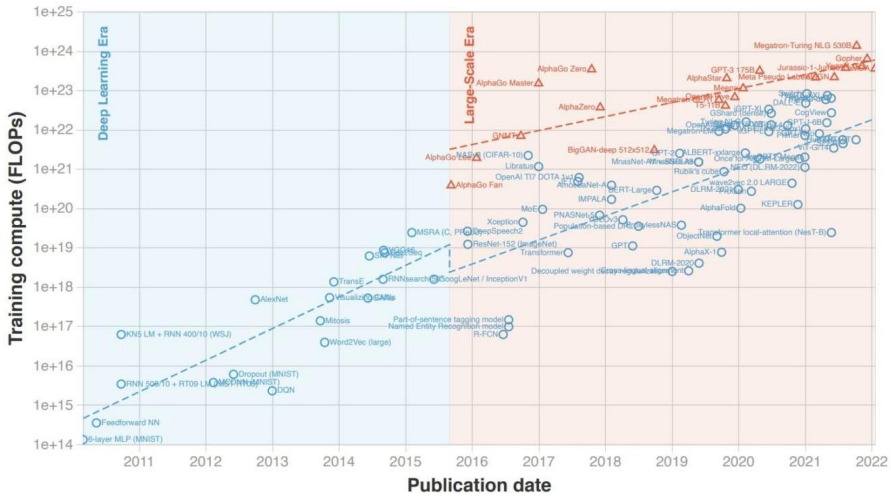
MIPT, ISP RAS, Innopolis University, MBZUAI

October 10, 2024

Modern trends in machine learning

- Exponential growth in model sizes and data volumes.





Varieties of distributed learning

- Cluster learning (big players): training within one large and powerful computing cluster
- Collaborative learning (all players): pooling computing resources over the Internet

Varieties of distributed learning

- Cluster learning (big players): training within one large and powerful computing cluster
- Collaborative learning (all players): pooling computing resources over the Internet
- Formally (horizontal, offline distributed learning):

$$\min_{x \in \mathbb{R}^d} \left[f(x) := \frac{1}{M} \sum_{m=1}^{M} f_m(x) := \frac{1}{M} \sum_{m=1}^{M} \frac{1}{n_m} \sum_{i=1}^{n_m} l(g(x, a_i^m), b_i^m) \right],$$

  where x – model weights, g – model, l – loss function.
- The data is shared among M computational devices, each device m has its own local subsample $\{a_i^m, b_i^m\}_{i=1}^{n_m}$ of size n_m.

Communicate centralized via server

Let us look at an example of how classical SGD becomes distributed.

Алгоритм 1 Centralized Distributed GD

Вход: Stepsize $\gamma > 0$, starting point $x_0 \in \mathbb{R}^d$, number of iterations K

for $k = 0, 1, \ldots, K-1$ **do**
  Send x_k to all workers ▷ server
  for $m = 1, \ldots, M$ in parallel **do**
    Receive x_k from server ▷ workers
    Compute gradient $g_{m,k} = \nabla f_m(x_k, \xi_{m,k})$ ▷ workers
    Send $g_{m,k}$ to server ▷ workers
  end for
  Receive $g_{m,k}$ from all workers ▷ server
  Compute $g_k = \frac{1}{M} \sum_{m=1}^{M} g_{m,k}$ ▷ server
  $x_{k+1} = x_k - \gamma g_k$ ▷ server
end for

Problem

- We can't trust all the workers. They can be lazy or even adversarial.

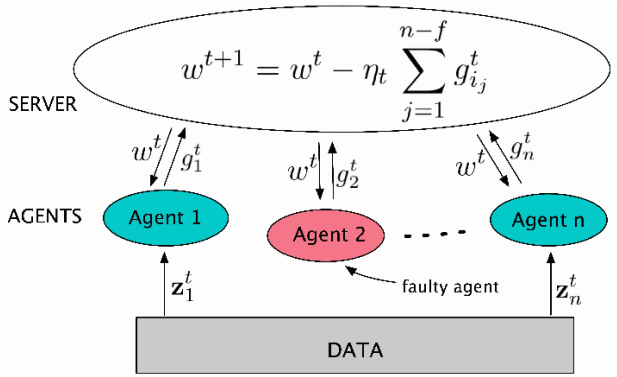
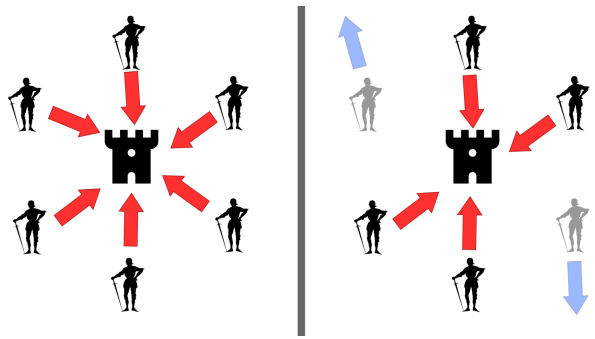


Figure: Distributed computing with adversarial worker

- Classical problem of Byzantine generals – find applications even in modern aviation.



Related work

- Classical methods like SGD, ADAM, and SCAFFOLD fail with even one Byzantine worker.

Related work

- Classical methods like SGD, ADAM, and SCAFFOLD fail with even one Byzantine worker. **Question:** why?
- Mentioned Distributed SGD:

$$g = \frac{1}{M}\sum_{m=1}^{M} g_m.$$

If Byzantine worker j send "gradient" $g_j = -\sum_{m\in[M]/j} g_m$, learning process stops.

Related Work

Coordinate-wise median:

$$[\mathrm{CM}(g_1, \ldots, g_M)]_j = \mathrm{median}([g_1]_j, \ldots, [g_n]_j).$$

Intuition: the median value of the gradients for each coordinate.

Krum: Suppose subset $S \subset [M]$ with size at least $(M - \delta M - 2)$, where among M workers, at least $(1 - \delta)n$ is the honest ones.

$$\mathrm{Krum}(g_1, \ldots, g_M) = \arg \min_{g_i} \min_{S} \sum_{j \in S} \|g_i - g_j\|_2^2.$$

Intuition: the closest to the average gradient after excluding $\delta n + 2$ furthest away gradients.

Trimmed Mean: For each coordinate j compute sotring sort_j.

$$[\mathrm{TM}(g_1, \ldots, g_M)]_j = \frac{1}{M - 2\delta M} \sum_{i=\delta M}^{M - \delta M} [g_{\mathrm{sort}_j(i)}]_j.$$

Intuition: the average after excluding δM smallest and highest gradient values for each coordinate.

Related Work

Client Momentum with Central Clipping Aggregation:

$$\mu_{m,k} = (1-\beta)g_{m,k} + \beta\mu_{m,k-1}$$

$$c_{m,k} = (\mu_{m,k} - \hat{\mu}_{k-1}) \cdot \min\left(1, \tau/\|\mu_{m,k} - \hat{\mu}_{k-1}\|\right)$$

$$\hat{\mu}_k = \hat{\mu}_{k-1} + 1/M \sum_{m=1}^{M} \mu_{m,k}$$

$$x_{k+1} = x_k - \gamma\hat{\mu}_k$$

Intuition: clip/cut large deviations from the past average.

Related Work

Client Momentum with Central Clipping Aggregation:

$$\mu_{m,k} = (1-\beta)g_{m,k} + \beta\mu_{m,k-1}$$

$$c_{m,k} = (\mu_{m,k} - \hat{\mu}_{k-1}) \cdot \min\left(1, \tau/\|\mu_{m,k} - \hat{\mu}_{k-1}\|\right)$$

$$\hat{\mu}_k = \hat{\mu}_{k-1} + 1/M \sum_{m=1}^{M} \mu_{m,k}$$

$$x_{k+1} = x_k - \gamma\hat{\mu}_k$$

Intuition: clip/cut large deviations from the past average.

RECCESS:

$$S(g_m, g_j) = \frac{\langle g_m, g_j \rangle}{\|g_j\| \ cdot \|g_m\|}$$

$$w_{m,k} = \frac{\sum_{j=1}^{M} S(g_m, g_j)}{\sum_{j=1}^{M}\sum_{i=1}^{M} S(g_i, g_j)}$$

$$x_{k+1} = x_k - \gamma \sum_{m=1}^{M} w_{m,k} g_{m,k}$$

Intuition: to see if all possible gradient pairs are same orientated.

Setup

In the distributed setting, we are guaranteed data homogeneity, which means that each distribution $\mathcal{D}_i$ coincides with the distribution of the entire training sample $\mathcal{D}$, where $\mathcal{D}_i$ is an unknown distribution of the training sample data on the i-th device.

Stochastic minimization problem

This allows us to move to the classical formulation of the problem:

$$\min_{x \in \mathbb{R}^d} \left[f(x) = \mathbb{E}_{\xi \sim \mathcal{D}} \left[f_\xi(x) \right] \right]. \tag{1}$$

GofM is the number of good workers.

Setup

To tackle Byzantine attacks, we introduce a trial loss function $\hat{f}$. In setting (1), we sample a subset from $\mathcal{D}$ to calculate $\hat{f}$. The distribution of the trial function matches $\mathcal{D}$, making it an honest estimator of f. The larger the sample, the better $\hat{f}$ approximates f.

Formal definition

Formally, the trial function is as follows:

$$\hat{f}(x) = \frac{1}{N} \sum_{i=1}^{N} f_i(x),$$

where N is the number of samples in $\hat{f}$.

Assumptions

Assumption 1

The function $\hat{f}$ is L-smooth, i.e., it satisfies $\|\nabla \hat{f}(x) - \nabla \hat{f}(y)\| \leq L\|x - y\|$ for any $x, y \in \mathbb{R}^d$.

Assumption 2

- Convexity: the function $\hat{f}$ is convex, i.e., it satisfies $\hat{f}(y) \leqslant \hat{f}(x) + \langle \nabla \hat{f}(x), y - x \rangle$ for any $x, y \in \mathbb{R}^d$.
- Non-convexity: the function $\hat{f}$ has a (may be not unique) minimum, i.e., $\hat{f}^* = \inf\limits_{x \in \mathbb{R}^d} \hat{f}(x) > -\infty$.

Assumptions

Assumption 3

Each worker $i \in G$ has access to an independent and unbiased stochastic gradient with (depending on the either homogeneity or heterogeneity of the data distribution): $\mathbb{E}[g_i(x, \xi_i)] = \nabla f(x)$ and its variance is bounded by σ^2_{hom} : $\mathbb{E}[\|g_i(x, \xi_i) - \nabla f(x)\|^2] \leq \sigma^2_{\text{hom}}$;

Assumption 4

Byzantine workers are assumed to be omniscient, i.e., they have access to the computations made by the rest of the workers.

First idea

- Indication of Trial Function Reduction (ITR-SGD): Weights (trust score) are determined by how much the gradient, relative to those from other devices, reduces the trial function:

$$\omega_{m,k} = (1-\beta)\omega_{m,k-1} + \beta \frac{[\hat{f}(x_k) - \hat{f}(x_k - \gamma g_{m,k-1})]_0}{\sum_{j=1}^{M}[\hat{f}(x_k) - \hat{f}(x_k - \gamma g_{m,k})]_0}$$

$$x_{k+1} = x_k - \gamma \sum_{i=1}^{n} 1_{[\hat{f}(x_k) - \hat{f}(x_k - \gamma g_{m,k}) > 0]} \cdot \omega_{m,k} \cdot g_{m,k}$$

ITR-SGD analysis

Theorem

For solving the problem described in (1)*, after* K *iterations of* ITR-SGD *with* $\gamma \leqslant \frac{1}{4L}$*, the following holds under Assumptions 1, 2.1, 3, 4*

$$\frac{1}{K}\sum_{t=0}^{K-1}\mathbb{E}\|\nabla f(x_k)\|^2 \leqslant \frac{2\mathbb{E}\left[\hat{f}(x_0)-\hat{f}(\hat{x}^*)\right]}{\gamma T}\cdot\frac{M}{\beta|G|}+4L\gamma\sigma_{hom}^2+\frac{1}{L\gamma}\zeta(N).$$

- First term – convergence of GD, but slow down by $fracM\beta|G|$.
- Second term – convergence of SGD with $|G|$ clients.
- Third term – $\hat{f} \approx f$ error.

Second idea

- Trial Function Minimization (TFM-SGD):
  Select optimal weights (trust scores) of the gradients on which the trial function is minimal:

$$\omega_k \approx \arg\min_{\omega \in \Delta} \hat{f}\left(x_k - \gamma \sum_{m=1}^{M} \omega_m g_{m,k}\right)$$

$$x_{k+1} = x_k - \gamma \sum_{m=1}^{M} \omega_{m,k} g_{m,k}$$

TFM-SGD analysis

Theorem

Suppose Assumptions 1, 2.2, 3, 4 hold. Then after T iteration of TFM-SGD *with $\gamma \leqslant \frac{1}{4L}$, it implies that*

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}\left\|\nabla f(x_k)\right\|_2^2 \leqslant \frac{2\mathbb{E}\left[\hat{f}(x_0)-\hat{f}^*\right]}{\gamma K} + \frac{4L\gamma}{|G|}\sigma_{hom}^2 + \frac{\zeta(N)}{\gamma L} + \frac{2\delta}{\gamma}.$$

- First term – convergence of GD, but slow down by $fracM\beta|G|$.
- Second term – convergence of SGD with $|G|$ clients.
- Third term – $\hat{f} \approx f$ error.
- Fourth term – subproblem error.

Third idea

- FineTuned-SGD:
  Look not at the loss function $\hat{f}$, but at the output of the model g on the trial data $\hat{D}$.

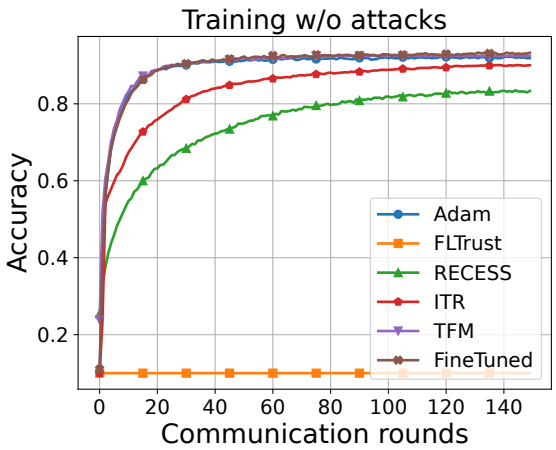
$$\omega_{m,k} = (1-\beta)\omega_{m,k-1} + \beta \frac{sim(m(x_k - \gamma g_{m,k}, \hat{D}), m(x_k - \gamma g_{0,k}, \hat{D}))}{\sum_{j=1}^{n} sim(m(x_k - \gamma g_{m,k}, \hat{D}), m(x_k - \gamma g_{0,k}, \hat{D}))}$$

$$x_{k+1} = x_k - \gamma \sum_{m=1}^{M} \omega_{m,k} g_{m,k}$$

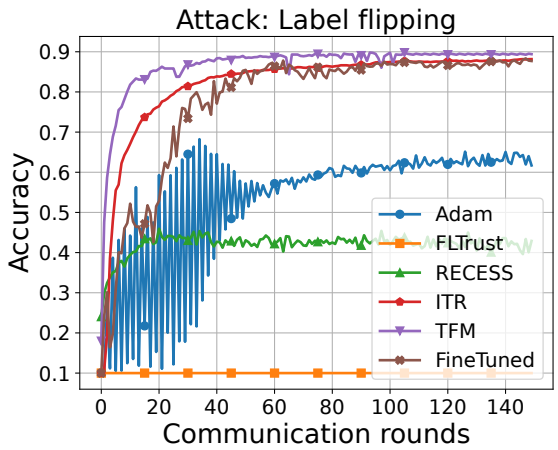
Experiments

- We conduct experiments on public CIFAR-10 image dataset.
- We use ResNet-18 model.

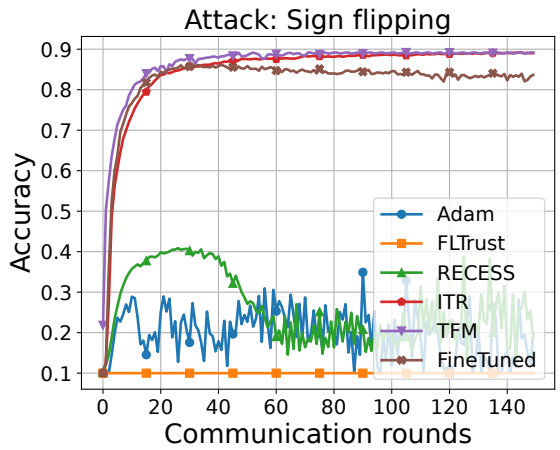
Experiments



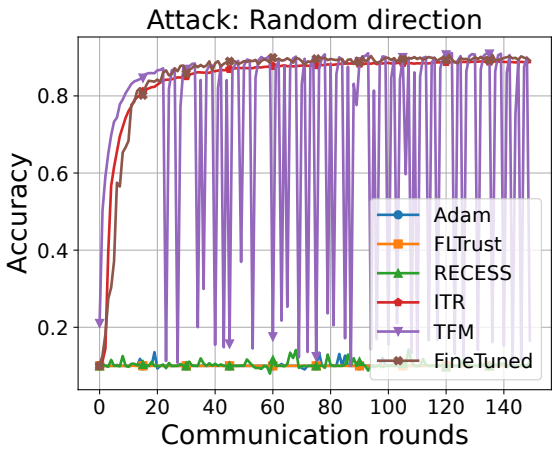
(a) Without Attack



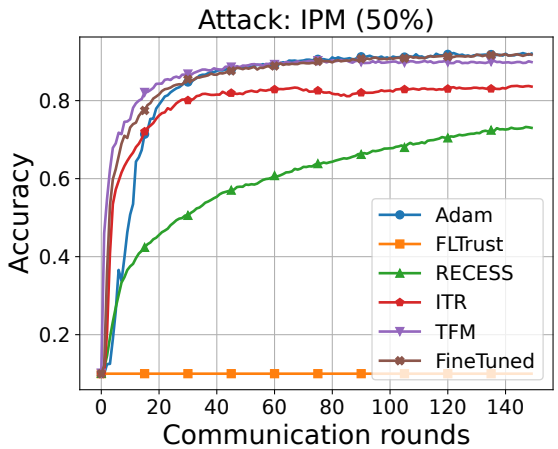
(b) Label Flip



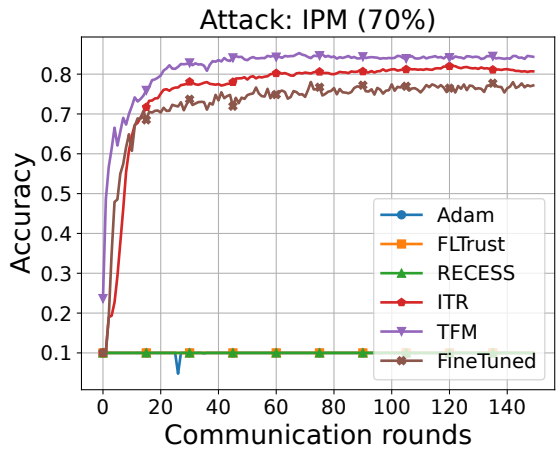
(c) Sign Flip



(d) Random Clients

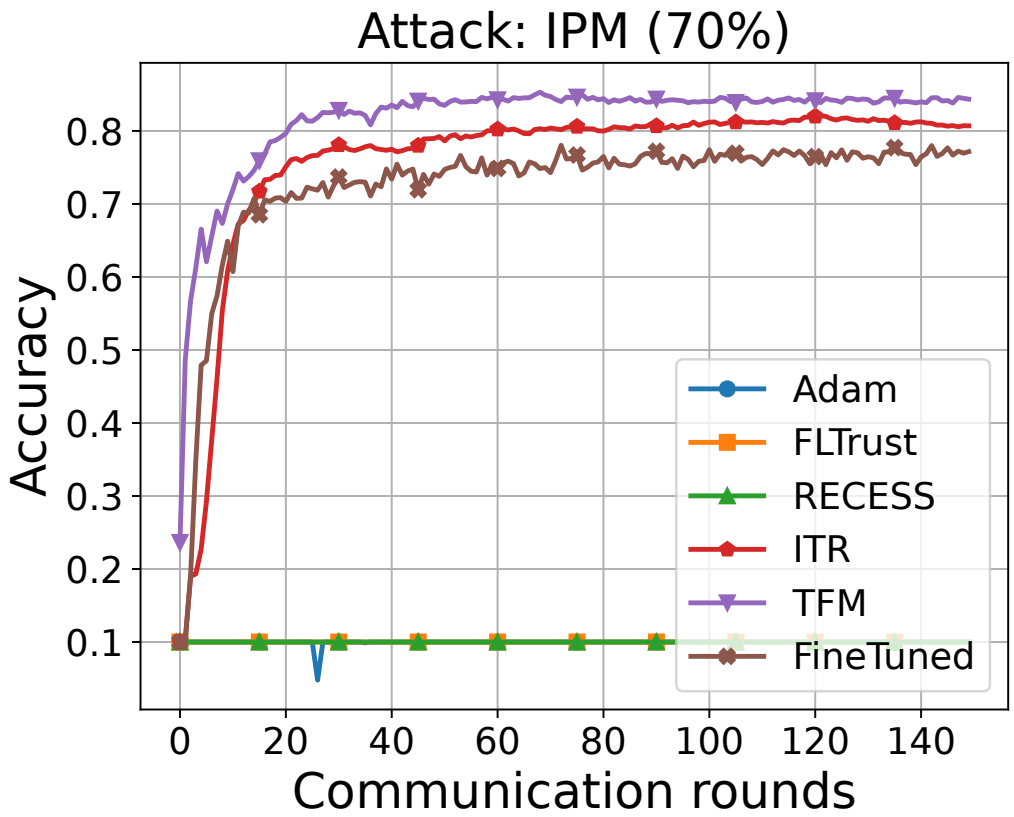


(e) IPM (50 %)



(f) IPM (70 %)

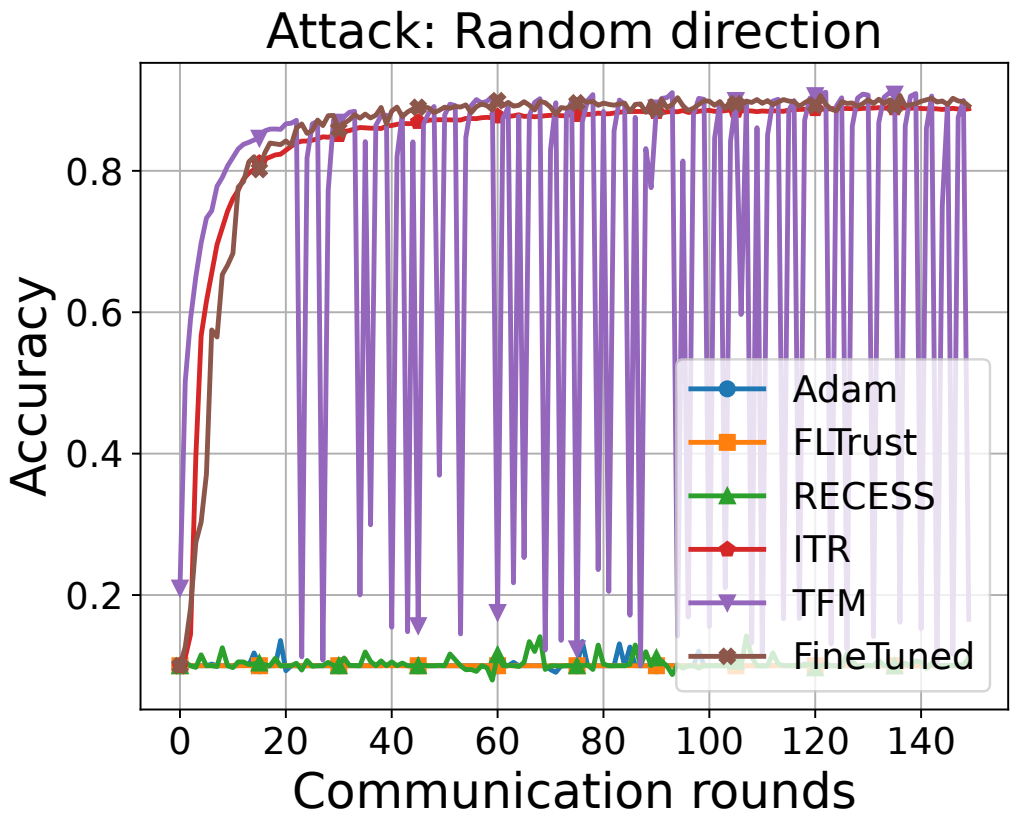
Experiments



(f) IPM (70 %)

Figure: ResNet18 on CIFAR-10 experiments results. Test accuracy in case of various attacks and Byzantine-tolerance techniques.

Experiments



(d) Random Clients

Figure: ResNet18 on CIFAR-10 experiments results. Test accuracy in case of various attacks and Byzantine-tolerance techniques.