Moshpit SGD:

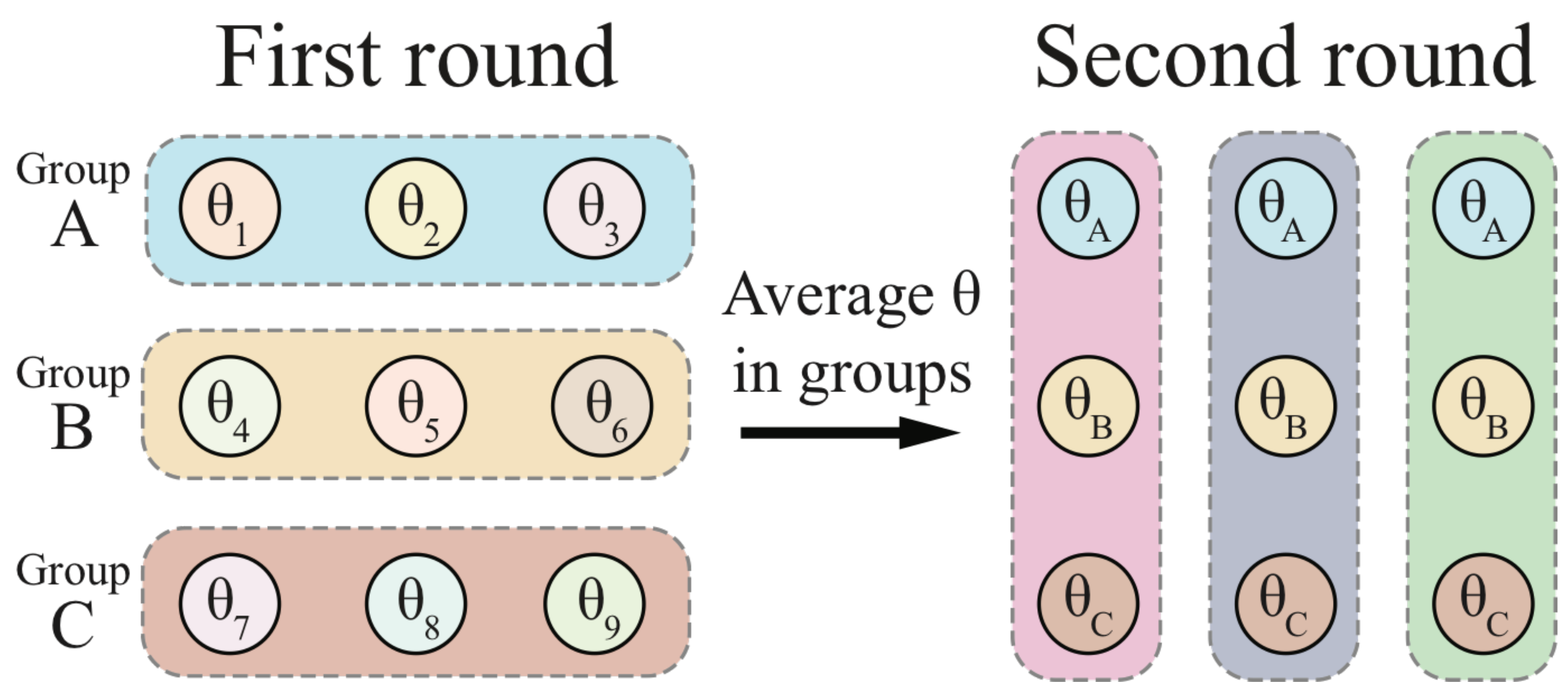
Communication-Efficient Decentralized Training on Heterogeneous Unreliable Devices

Max Ryabinin*, Eduard Gorbunov*,
Vsevolod Plokhotnyuk, Gennady Pekhimenko

Research MIPT MOSCOW INSTITUTE OF PHYSICS AND TECHNOLOGY HSE UNIVERSITY VECTOR INSTITUTE

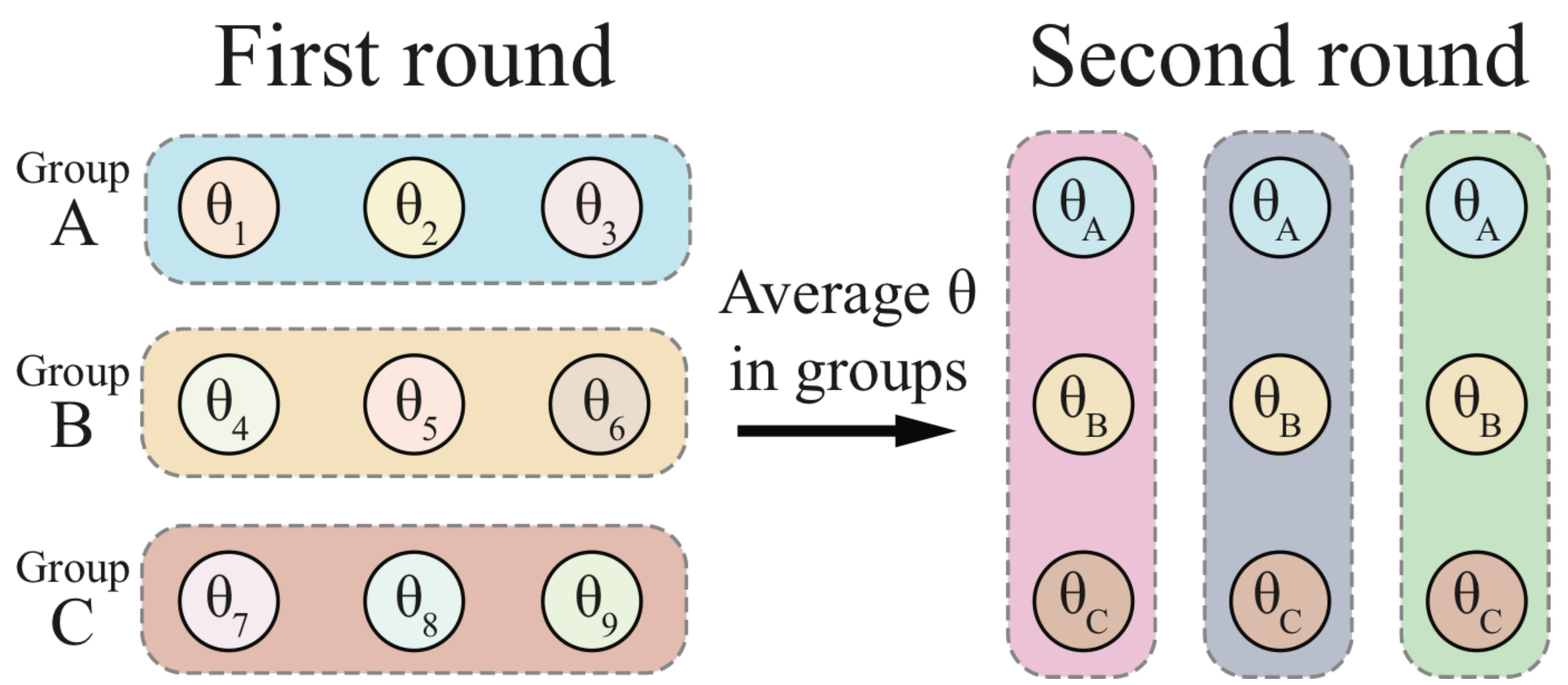
Summary

- We propose a scalable method for data-parallel training on unreliable devices



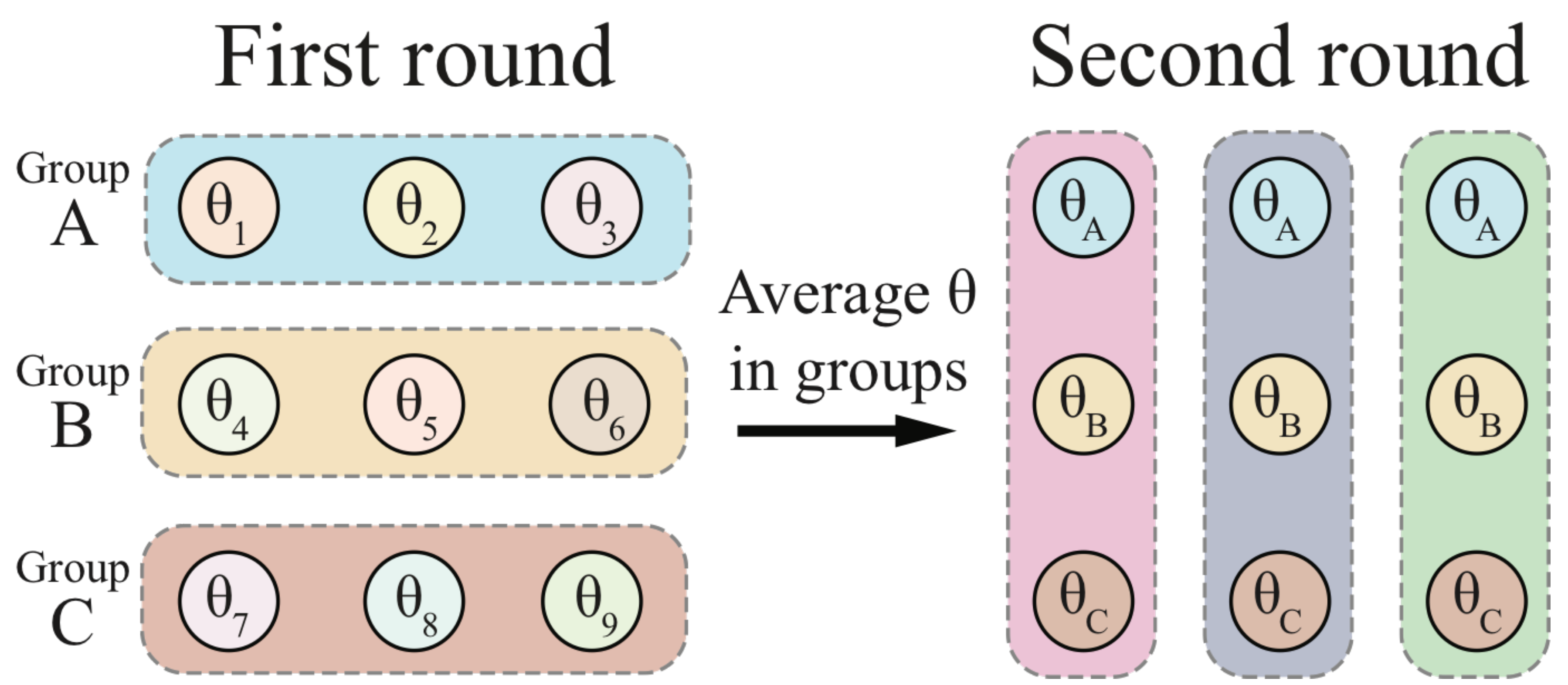
Summary

- We propose a scalable method for data-parallel training on unreliable devices
- It iteratively performs All-Reduce in non-overlapping groups to average the gradients



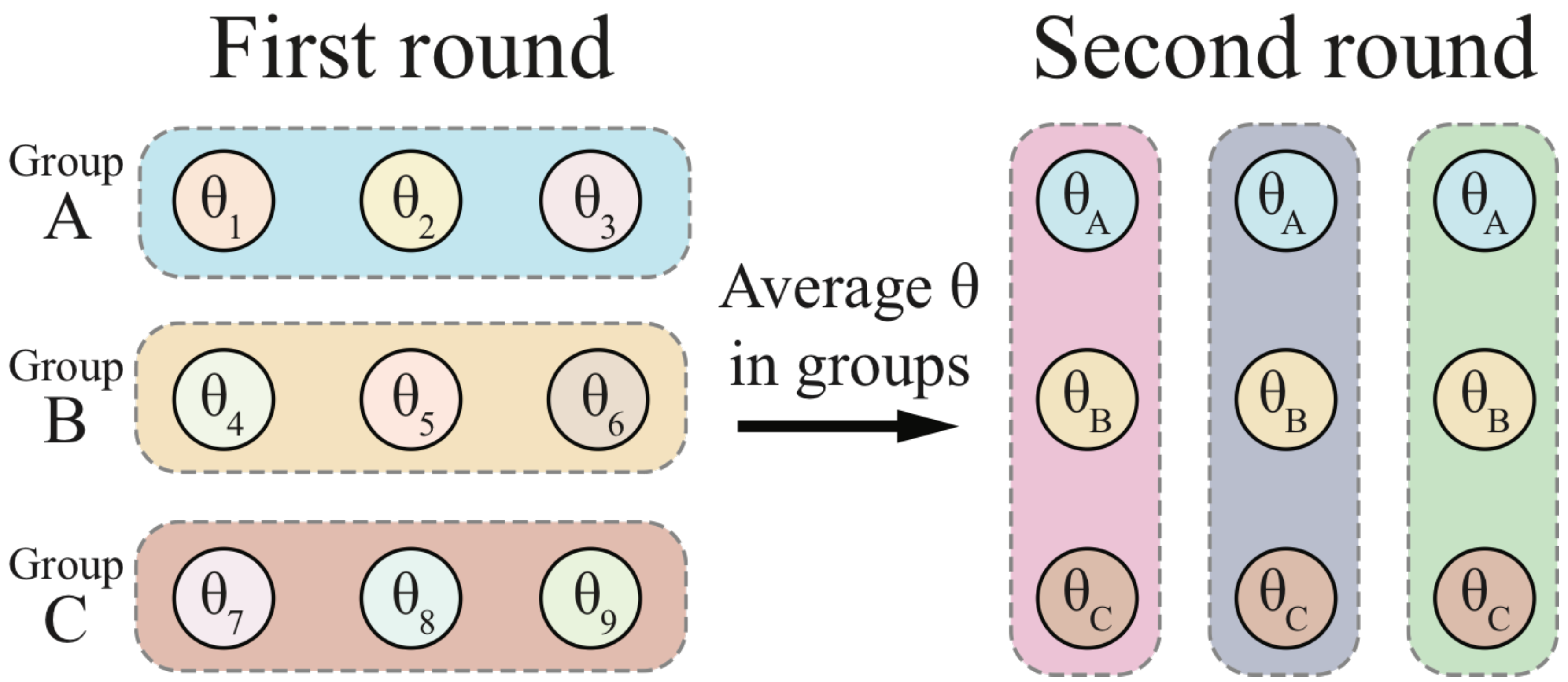
Summary

- We propose a scalable method for data-parallel training on unreliable devices
- It iteratively performs All-Reduce in non-overlapping groups to average the gradients
- Has strong theoretical guarantees



Summary

- We propose a scalable method for data-parallel training on unreliable devices
- It iteratively performs All-Reduce in non-overlapping groups to average the gradients
- Has strong theoretical guarantees
- Pretrain ResNet-50 and ALBERT on preemptible nodes faster than gossip-based strategies



Background

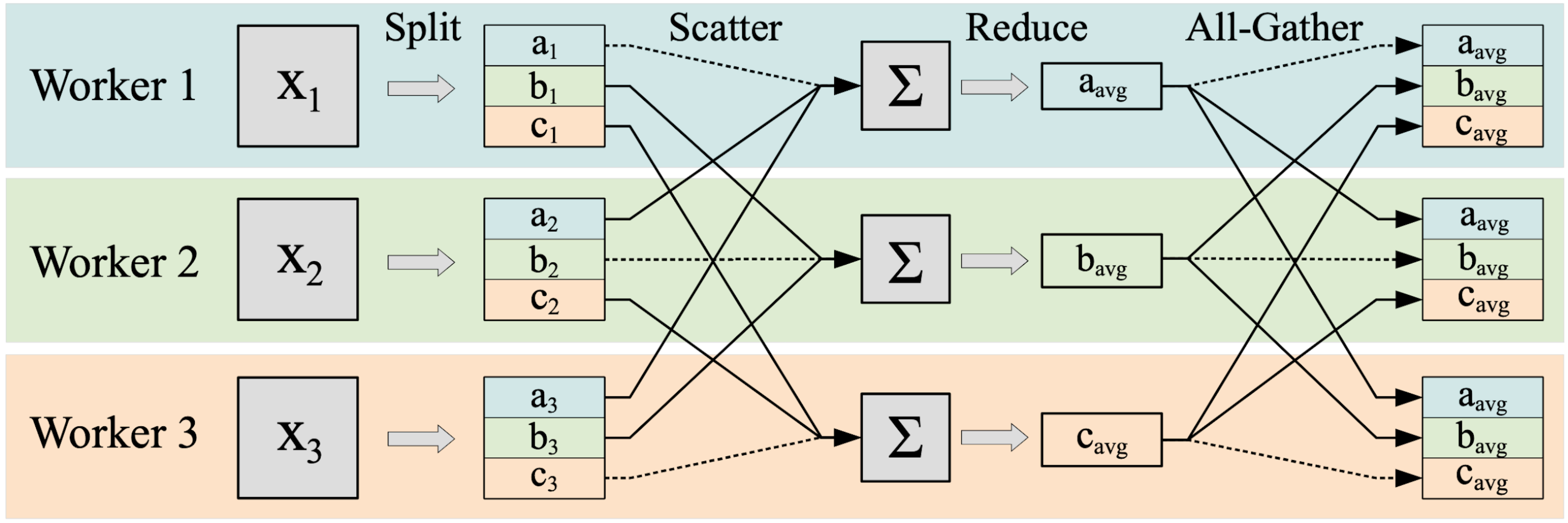
- Large-scale training is done in a distributed manner

Background

- Large-scale training is done in a distributed manner
- For the data-parallel case, you need to exchange gradients

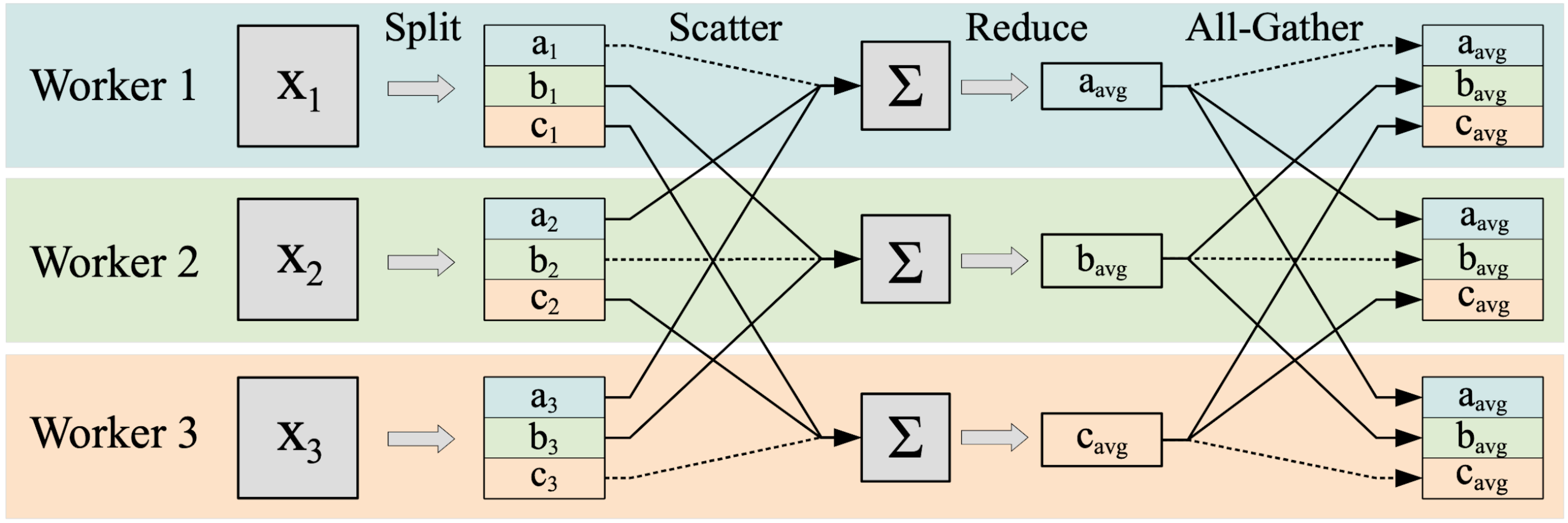
Background

- Large-scale training is done in a distributed manner
- For the data-parallel case, you need to exchange gradients
- Naive method would be O(n^2) in workers, faster AllReduce protocols are used in practice



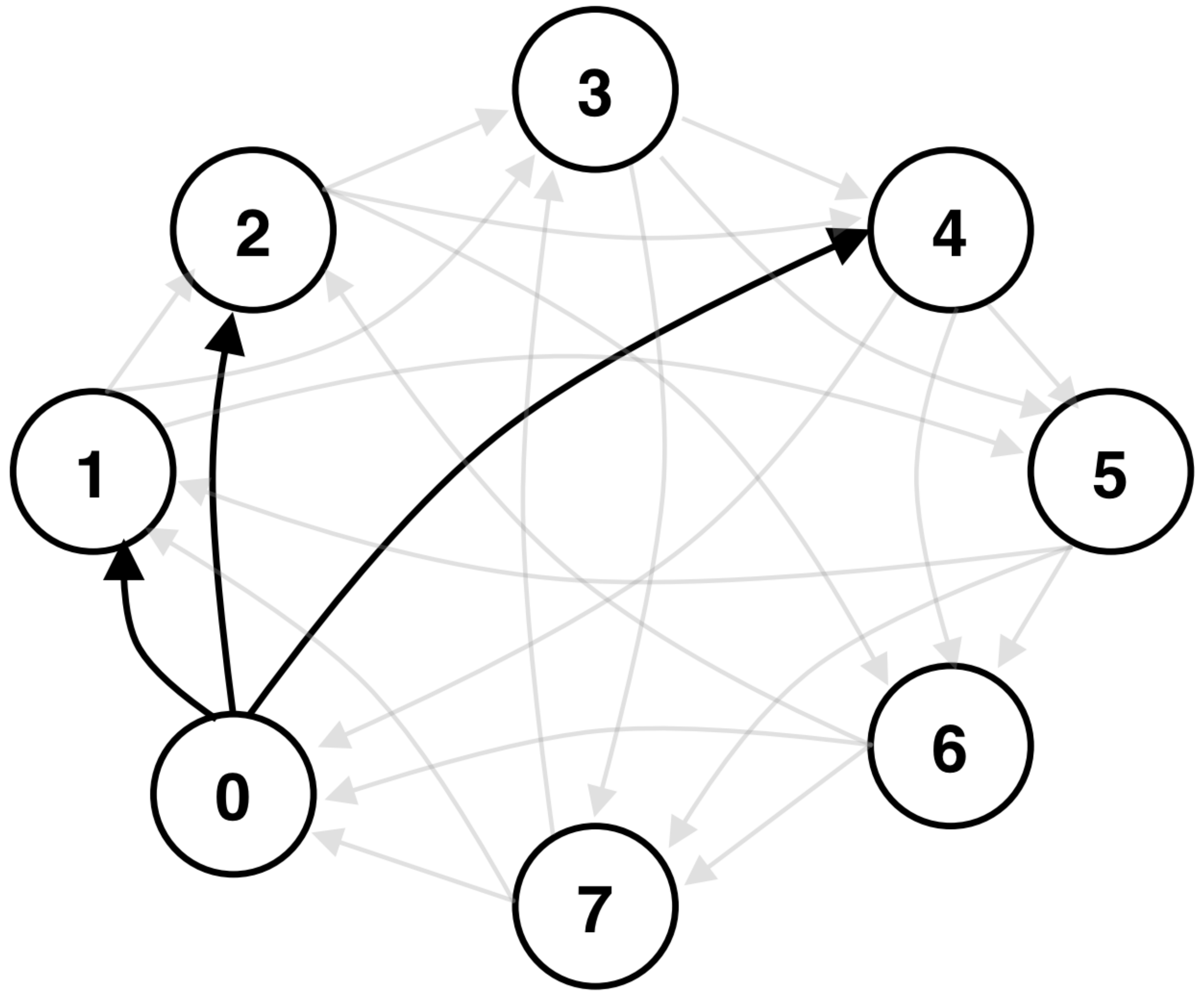
Background

- Large-scale training is done in a distributed manner
- For the data-parallel case, you need to exchange gradients
- Naive method would be O(n^2) in workers, faster AllReduce protocols are used in practice
- However, they are more fragile and need expensive high-speed network



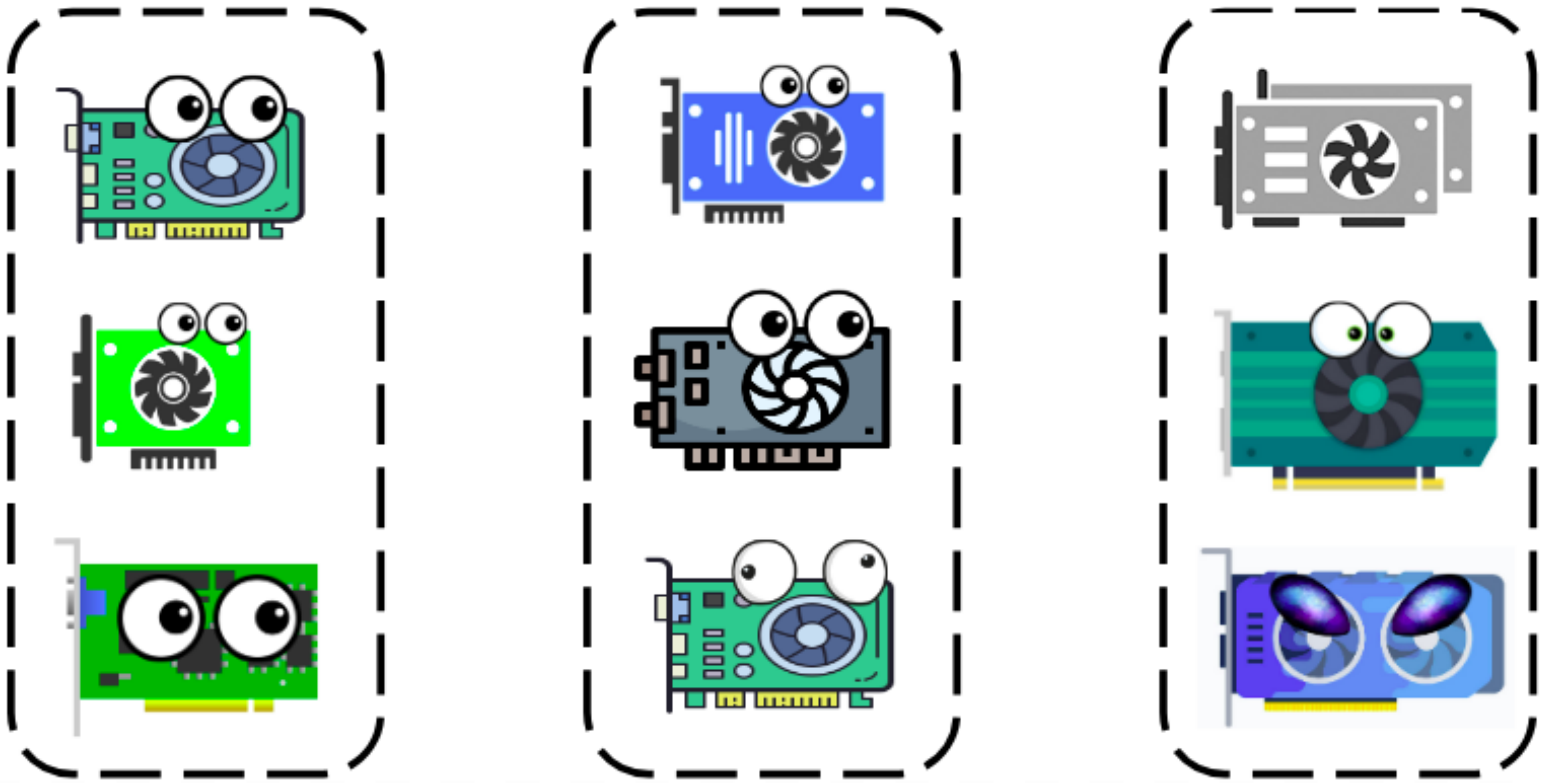
Background

- Large-scale training is done in a distributed manner
- For the data-parallel case, you need to exchange gradients
- Naive method would be O(n^2) in workers, faster AllReduce protocols are used in practice
- However, they are more fragile and need expensive high-speed network
- Gossip methods are more fault-tolerant, but less communication-efficient and converge slower

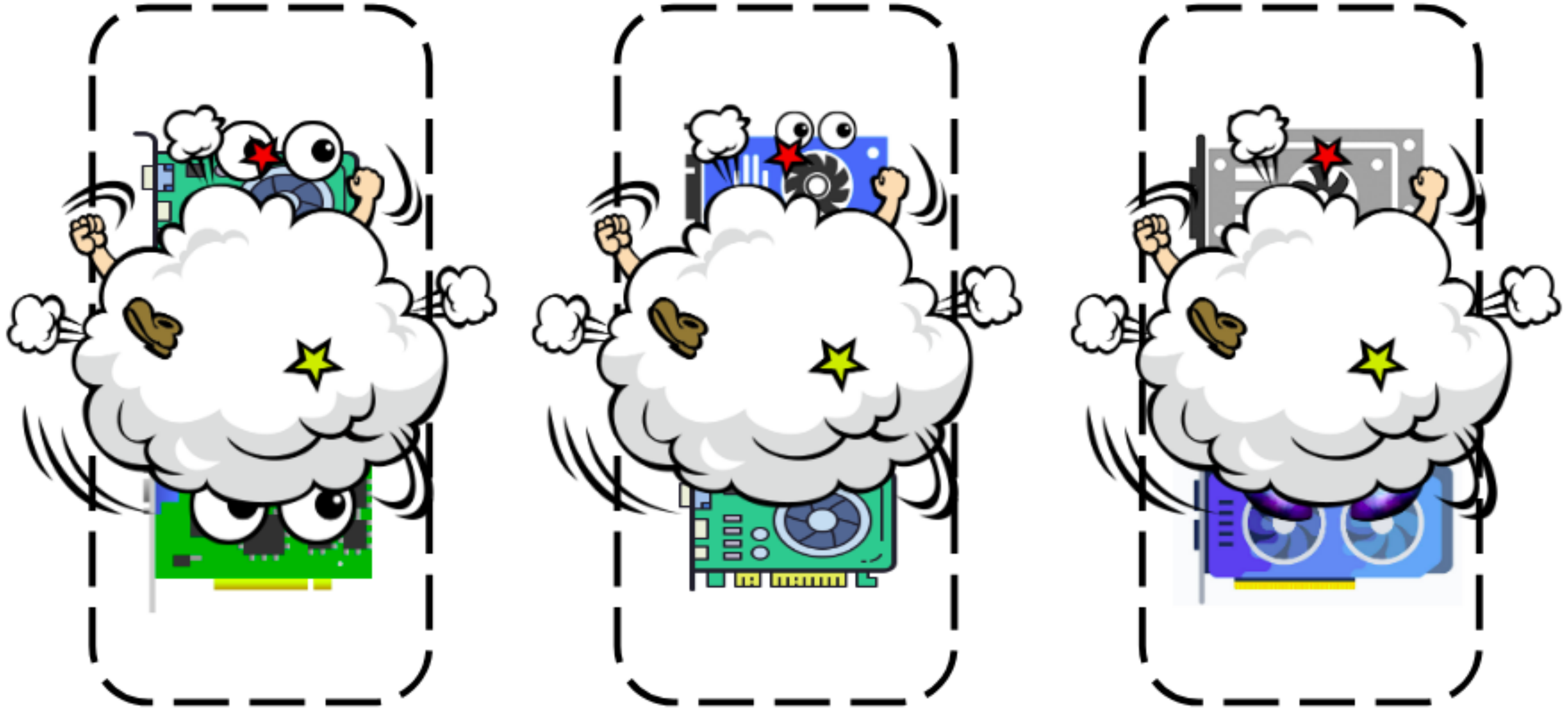


img src: Stochastic Gradient Push for Distributed Deep Learning.
Mahmoud Assran, Nicolas Loizou, Nicolas Ballas, Michael Rabbat. ICML 2019

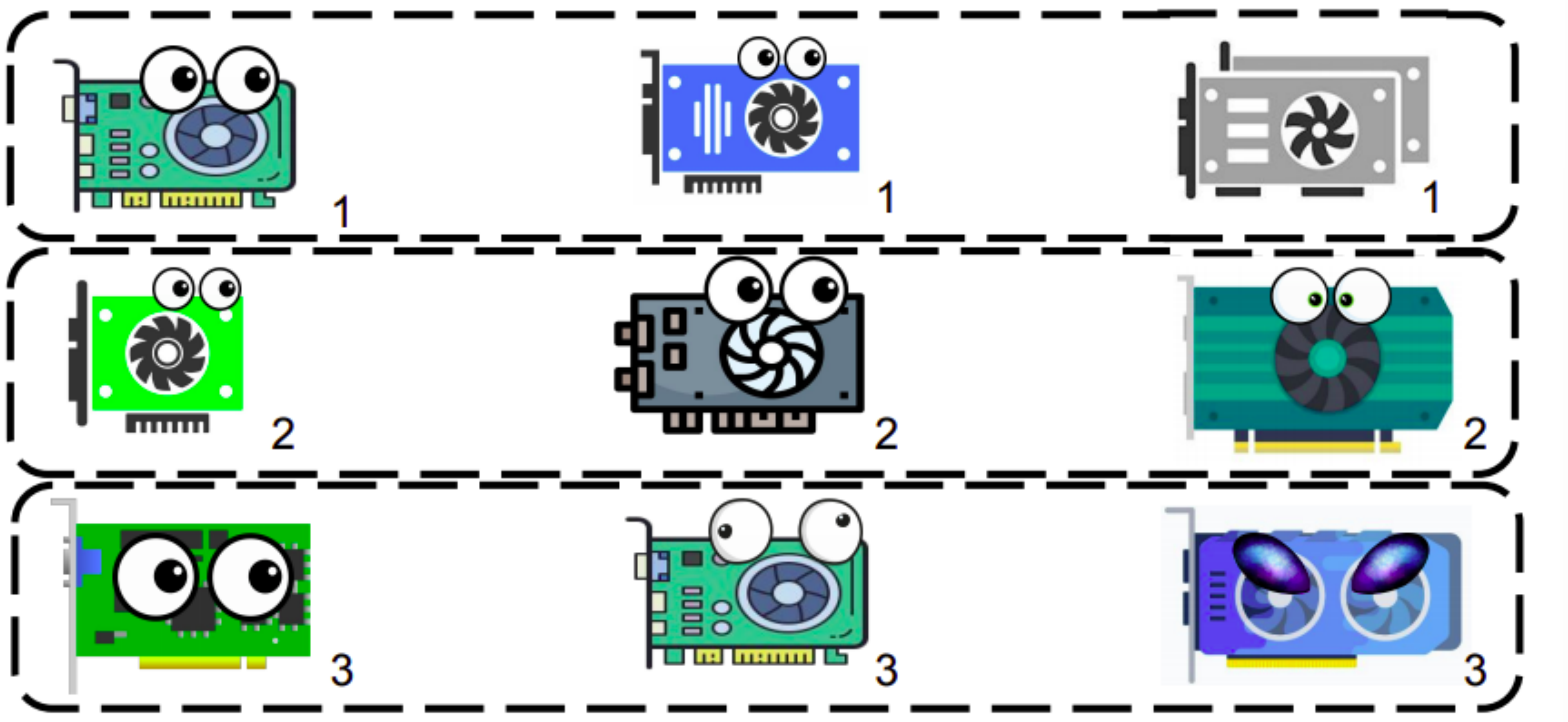
Moshpit All-Reduce: core idea



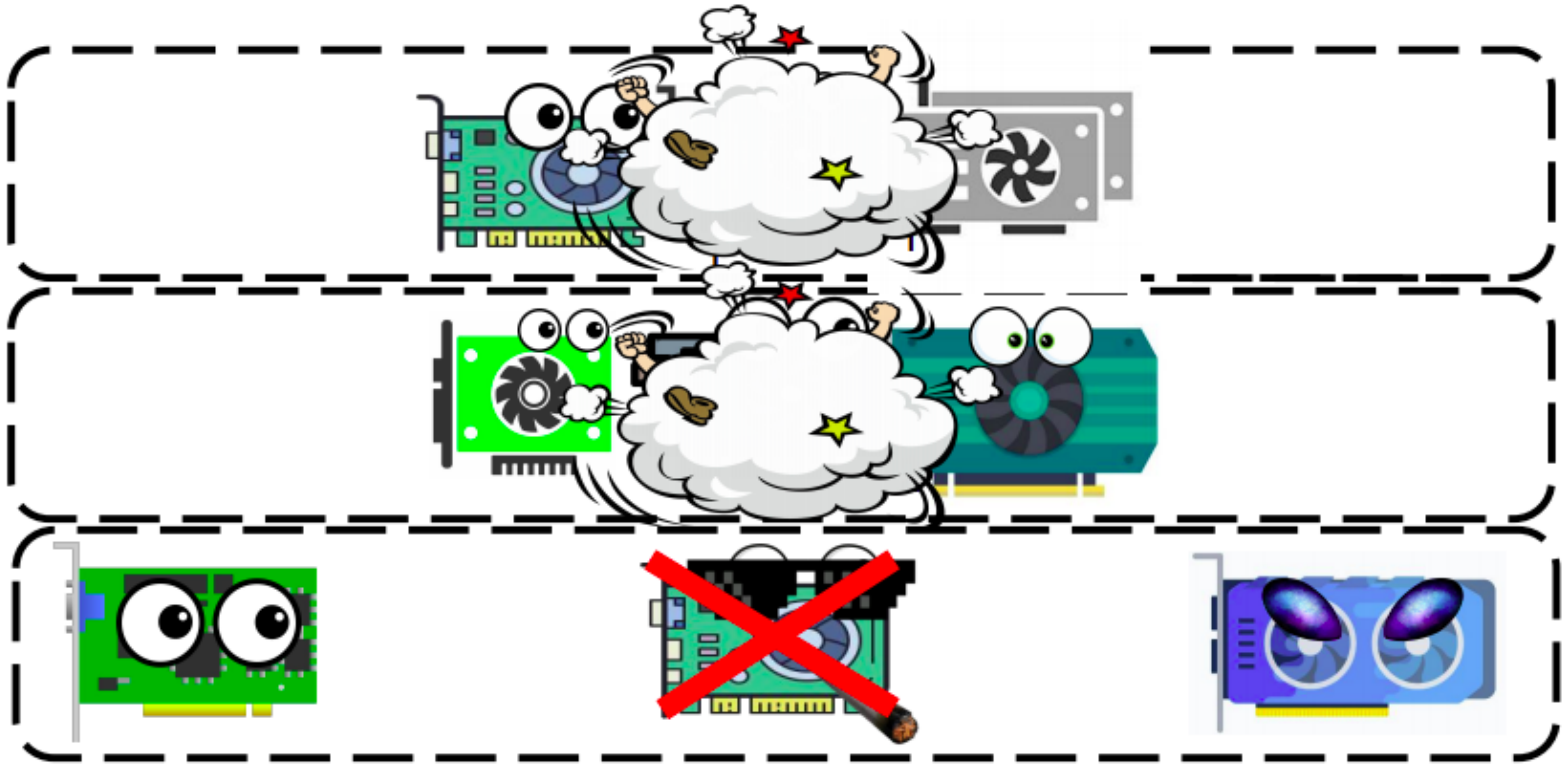
Moshpit All-Reduce: core idea



Moshpit All-Reduce: core idea

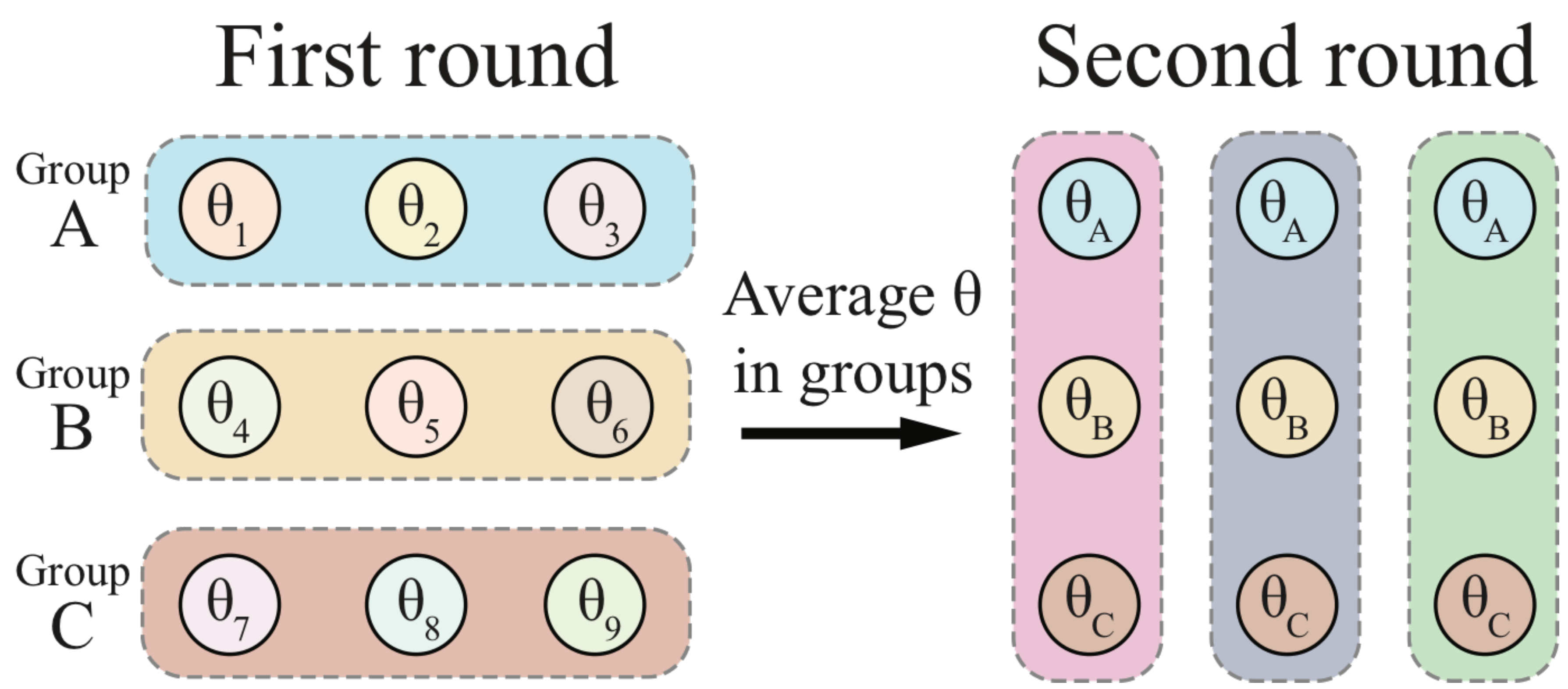


Moshpit All-Reduce: core idea



Moshpit All-Reduce

- Instead of running All-Reduce across all workers at once, let's do it in several steps with smaller groups
- Arrange peers in a (virtual) grid, average across one axis at once
- Workers find others via Distributed Hash Table — an efficient decentralized data structure
- Each single round is efficient because of All-Reduce, and multiple parallel groups give us fault tolerance!



Moshpit All-Reduce

Algorithm 1 Moshpit All-Reduce (for i-th peer)

Input: parameters $\{\theta_j\}_{j=1}^{N}$, number of peers N, d, M, number of iterations T, peer index i

$\theta_i^0 := \theta_i$

$C_i^0 :=$ `get_initial_index(i)`

for $t \in 1 \dots T$ **do**

  `DHT`$[C_i^{t-1}, t]$`.add`$($`address`$_i)$

  `Matchmaking()` // wait for peers to assemble

  `peers`$_t :=$ `DHT.get`$([C_i^{t-1}, t])$

  $\theta_i^t, c_i^t :=$ `AllReduce`$(\theta_i^{t-1},$ `peers`$_t)$

  $C_i^t := (C_i^{t-1}$`[1:]`$, c_i^t)$ // same as eq. (1)

end for

Return θ_i^T

$$\text{get\_initial\_index}(i) = \left(\left\lfloor i/M^{j-1}\right\rfloor \bmod M\right)_{j\in\{1,\ldots,d\}}$$

$$C_i^t := \left(c_i^{t-d+1}, c_i^{t-d+2}, \ldots, c_i^t\right)$$

Theoretical properties

- If $N = M^d$ and there are no faults, then Moshpit All-Reduce finds an exact average after d steps

Theoretical properties

- If $N = M^d$ and there are no faults, then Moshpit All-Reduce finds an exact average after d steps
- **Correctness:** if all workers have a non-zero probability of successfully running a communication round and the order of peers is random, then all local vectors converge to the global average with probability 1:

$$\forall i \quad \left\| \theta_i^t - \frac{1}{N} \sum_i \theta_i^0 \right\|_2^2 \xrightarrow[t \to \infty]{} 0$$

Theoretical properties

- If $N = M^d$ and there are no faults, then Moshpit All-Reduce finds an exact average after d steps
- **Correctness:** if all workers have a non-zero probability of successfully running a communication round and the order of peers is random, then all local vectors converge to the global average with probability 1:

$$\forall i \quad \left\| \theta_i^t - \frac{1}{N} \sum_i \theta_i^0 \right\|_2^2 \xrightarrow[t \to \infty]{} 0$$

- **Exponential convergence to the average:** for a version of Moshpit All-Reduce with random splitting into r groups at each step, we have

$$\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\left\|\theta_i^T - \bar{\theta}\right\|^2\right] = \left(\frac{r-1}{N} + \frac{r}{N^2}\right)^T \frac{1}{N}\sum_{i=1}^{N}\left\|\theta_i - \bar{\theta}\right\|^2$$

Theoretical properties

- If $N = M^d$ and there are no faults, then Moshpit All-Reduce finds an exact average after d steps

- **Correctness:** if all workers have a non-zero probability of successfully running a communication round and the order of peers is random, then all local vectors converge to the global average with probability 1:

$$\forall i \quad \left\| \theta_i^t - \boxed{\frac{1}{N}\sum_i \theta_i^0} \right\|_2^2 \xrightarrow[t\to\infty]{} 0$$

- **Exponential convergence to the average:** for a version of Moshpit All-Reduce with random splitting into r groups at each step, we have

$$\mathbb{E}\left[\frac{1}{N}\sum_{i=1}^{N}\left\|\theta_i^T - \boxed{\bar{\theta}}\right\|^2\right] = \left(\frac{r-1}{N} + \frac{r}{N^2}\right)^T \frac{1}{N}\sum_{i=1}^{N}\left\|\theta_i - \boxed{\bar{\theta}}\right\|^2$$

Optimization problem

$$\min_{x \in \mathbb{R}^n} f(x)$$

- Function *f(x)* is available through stochastic gradients only
- Each worker has an access to the stochastic gradients of *f(x)*

Moshpit SGD

$$x_i^{k+1} = \begin{cases} x_i^k - \gamma g_i^k, & \text{if } k+1 \bmod \tau \neq 0 \\ \texttt{Moshpit All-Reduce}_{j \in P_{k+1}}(x_j - \gamma g_j^k), & \text{if } k+1 \bmod \tau = 0 \end{cases}$$

Number of active workers at iteration *k+1*

Moshpit SGD

$$x_i^{k+1} = \begin{cases} x_i^k - \gamma g_i^k, & \text{if } k+1 \bmod \tau \neq 0 \\ \texttt{Moshpit All-Reduce}_{j \in P_{k+1}} (x_j - \gamma g_j^k), & \text{if } k+1 \bmod \tau = 0 \end{cases}$$

Number of active workers at iteration *k+1*

Local-SGD with Moshpit All-Reduce instead of averaging

Moshpit SGD: assumptions

- Homogeneity: $$f_1(x) = f_2(x) = \ldots = f_N(x) = f(x)$$

Moshpit SGD: assumptions

- Homogeneity: $$f_1(x) = f_2(x) = \ldots = f_N(x) = f(x)$$
- Bounded variance: $$\mathbb{E}\left[\left\|g_i^k - \nabla f_i\left(x_i^k\right)\right\|^2 \mid x_i^k\right] \leq \sigma^2$$

Moshpit SGD: assumptions

- Homogeneity: $$f_1(x) = f_2(x) = \ldots = f_N(x) = f(x)$$
- Bounded variance: $$\mathbb{E}\left[\left\|g_i^k - \nabla f_i\left(x_i^k\right)\right\|^2 \mid x_i^k\right] \leq \sigma^2$$
- Effect of peers' vanishing is bounded:

$$\mathbb{E}\left[\left\langle \boxed{x^{k+1}} - \boxed{\widehat{x}^{k+1}}, \boxed{x^{k+1}} + \boxed{\widehat{x}^{k+1}} - 2x^*\right\rangle\right] \leq \Delta_{pv}^k$$

$$\boxed{N_k = |P_k|}$$

$$\boxed{x^{k+1} = \frac{1}{\boxed{N_{k+1}}} \sum_{i \in P_{k+1}} x_i^{k+1}}$$

$$\boxed{\widehat{x}^{k+1} = \frac{1}{N_k} \sum_{i \in P_k} \left(x_i^k - \gamma g_i^k\right)}$$

Moshpit SGD: assumptions

- Homogeneity: $$f_1(x) = f_2(x) = \ldots = f_N(x) = f(x)$$
- Bounded variance: $$\mathbb{E}\left[\left\|g_i^k - \nabla f_i\left(x_i^k\right)\right\|^2 \mid x_i^k\right] \le \sigma^2$$
- Effect of peers' vanishing is bounded:

$$\mathbb{E}\left[\left\langle \boxed{x^{k+1}} - \boxed{\widehat{x}^{k+1}}, \boxed{x^{k+1}} + \boxed{\widehat{x}^{k+1}} - 2x^*\right\rangle\right] \le \Delta_{pv}^k$$

$$\boxed{N_k = |P_k|} \qquad \boxed{x^{k+1} = \frac{1}{\boxed{N_{k+1}}} \sum_{i \in P_{k+1}} x_i^{k+1}} \qquad \boxed{\widehat{x}^{k+1} = \frac{1}{N_k} \sum_{i \in P_k} \left(x_i^k - \gamma g_i^k\right)}$$

- Averaging quality: $$\mathbb{E}\left[\frac{1}{n_{a\tau}} \sum_{i \in P_{a\tau}} \left\|x_i^{a\tau} - x^{a\tau}\right\|^2\right] \le \gamma^2 \delta_{aq}^2$$

Moshpit SGD: convergence

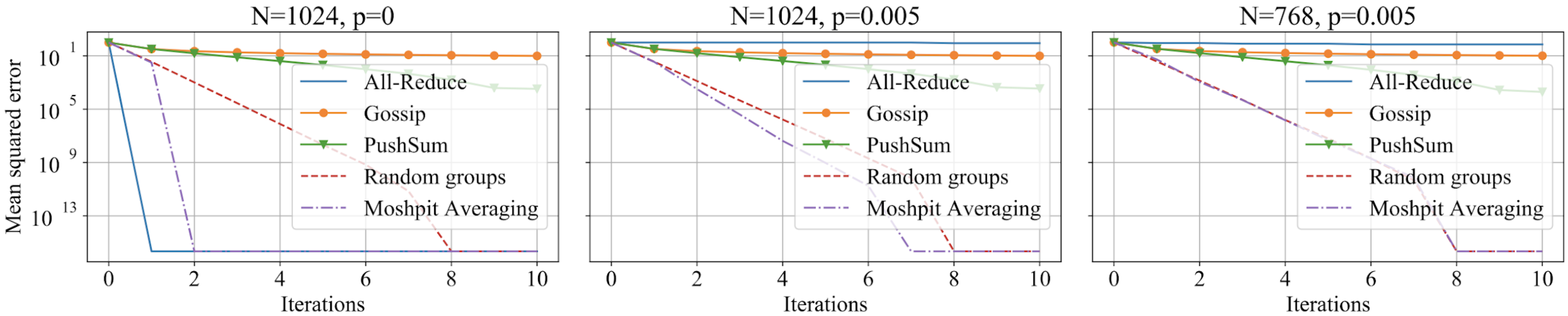
Under these assumptions we recover guarantees for ***centralized*** Local SGD:

- For convex problems, equivalent to [1,2]
- For non-convex problems — as in [3,4]

[1] Tighter Theory for Local SGD on Identical and Heterogeneous Data. Khaled et al., AISTATS 2020
[2] Is Local SGD Better than Minibatch SGD? Woodworth et al., ICML 2020
[3] A Unified Theory of Decentralized SGD with Changing Topology and Local Updates. Koloskova et al., ICML 2020
[4] Communication-Efficient Local Decentralized SGD Methods. Li et al., 2019

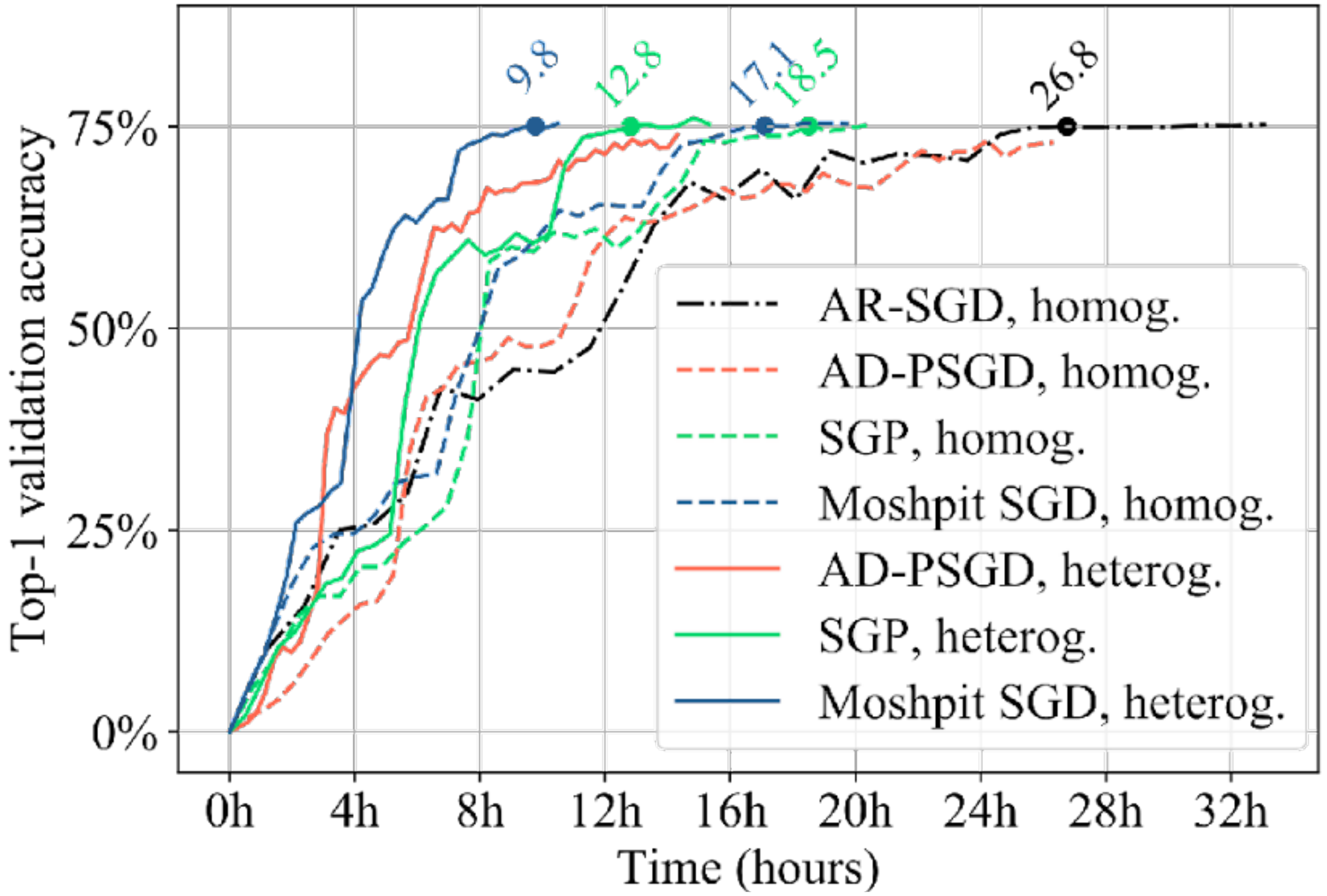
Experiments: averaging

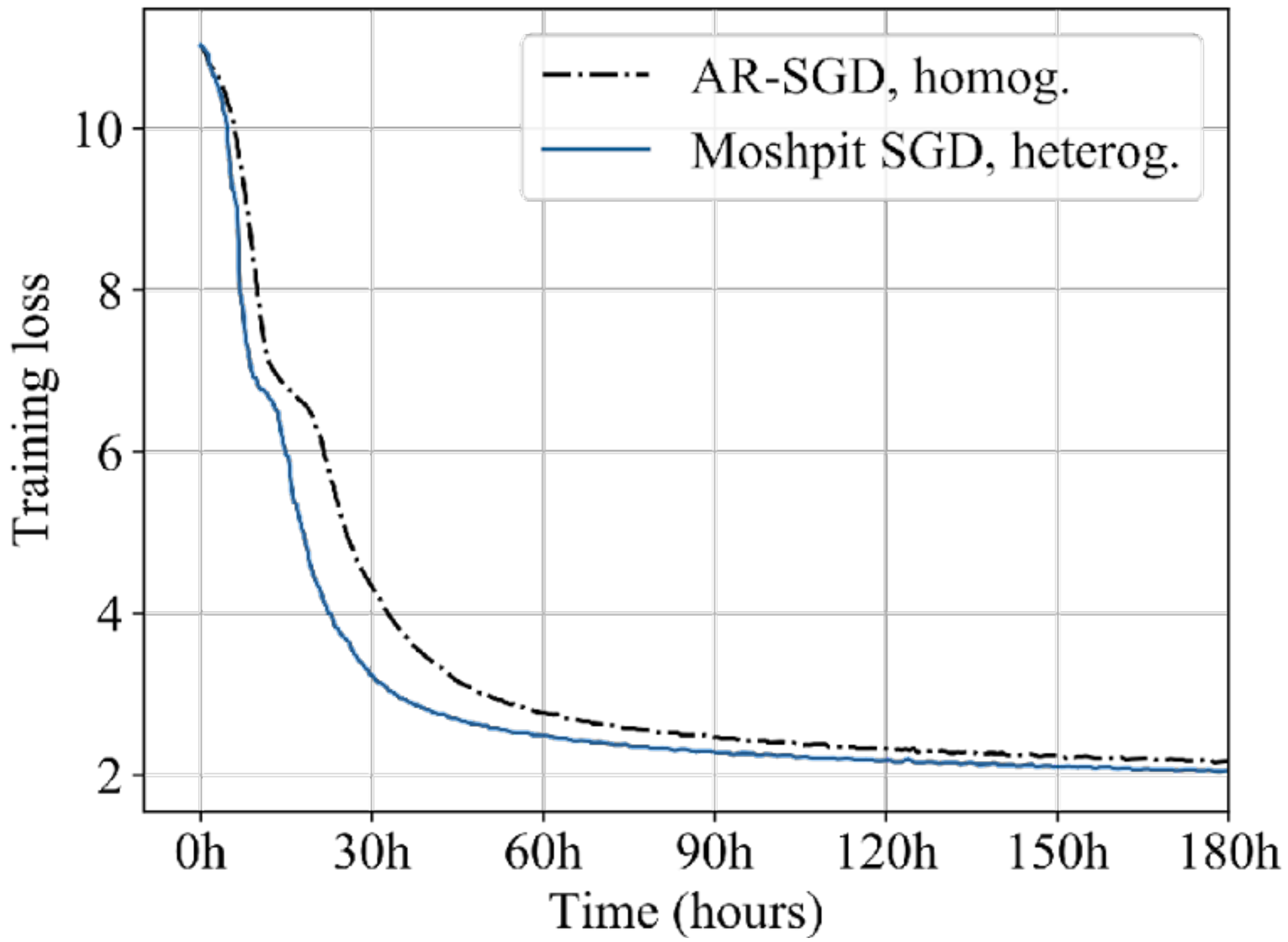
- First, we verify the performance gains in a controlled setting
- With non-zero failure probability, All-Reduce takes too many retries!
- On the other hand, Gossip-based methods converge very slowly
- Moshpit Averaging outperforms baselines with p>0 and gets the average in two rounds with p=0



Experiments: distributed training

- We train ResNet-50 and ALBERT-large on unreliable devices (e.g. spot instances)
- Baselines include both standard data-parallel training and decentralized methods
- Achieve the same quality faster and cheaper





Conclusion

- We propose a simple method for communication-efficient distributed training
- Built-in fault tolerance, convergence similar to standard methods
- Learn more:

Paper

arxiv.org/abs/2103.03239

Code

github.com/yandex-research/moshpit-sgd