Cooperative SGD

Towards Robust and Scalable Distributed Deep Learning

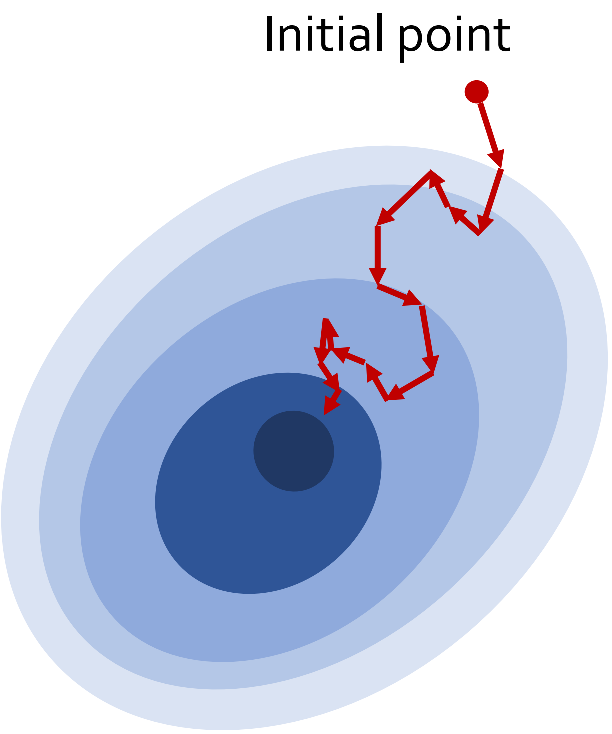
Jianyu Wang

Joint work with Gauri Joshi

18-847F

Stochastic Gradient Descent

Stochastic gradient descent (SGD) is the backbone of ML, especially deep learning



Loss incurred by the i-th sample

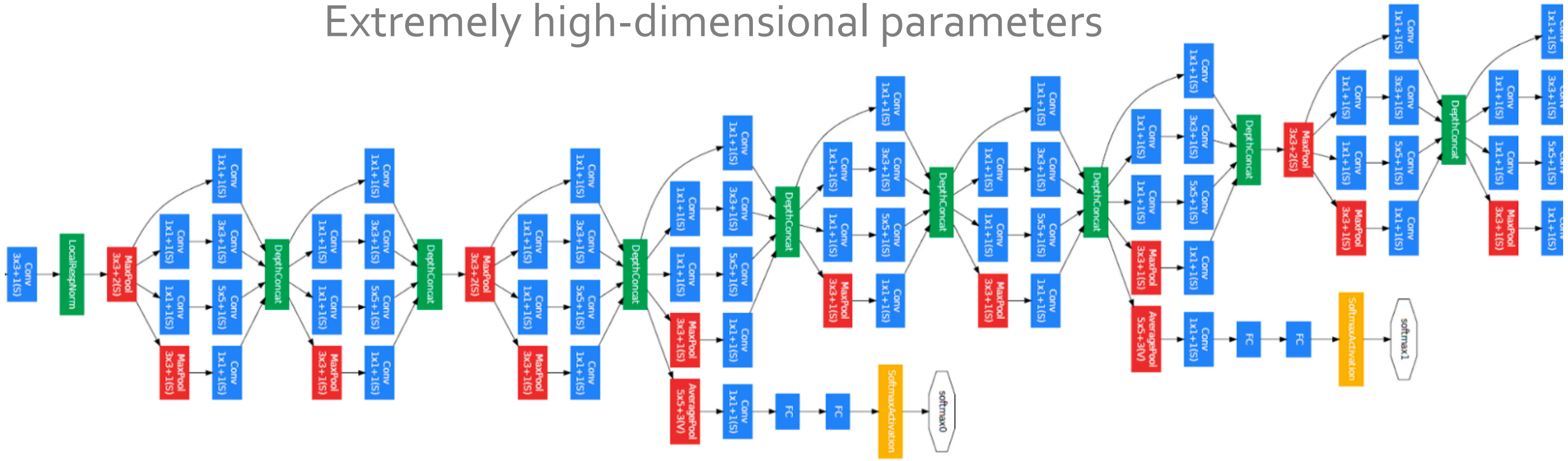
Empirical Risk $$F(\mathbf{x}) = \frac{1}{N}\sum_{i=1}^{N} \boxed{f_i(\mathbf{x})}$$

Mini-batch SGD $$\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \cdot \boxed{\frac{1}{|\xi_k|}\sum_{j\in\xi_k} \nabla f_j(\mathbf{x}_k)}$$

Stochastic gradient

Big Model, Big Data

Extremely high-dimensional parameters



IMAGENET

Extremely large training datasets

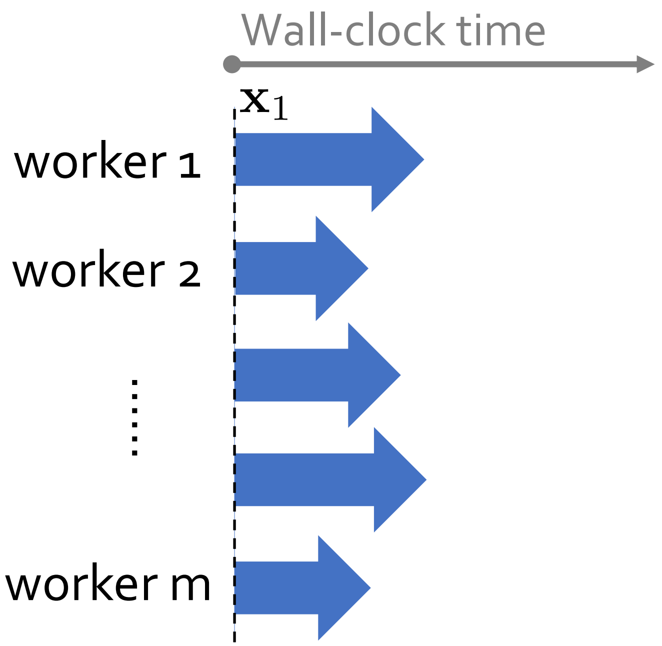
Training on a single machine
can takes several days or even weeks.

It is imperative to distribute SGD across multiple machines!

Classic Method: Fully Synchronous SGD

Execution pipeline:

1. Local stochastic gradients computation



Gradient at **k-th** iteration and **i-th** worker:

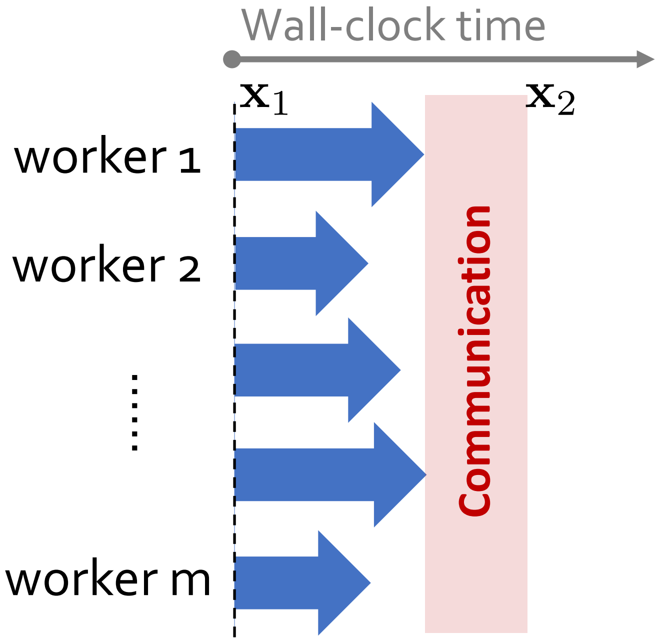
$$g(\mathbf{x}_k;\xi_k^{(i)}) = \frac{1}{|\xi_k^{(i)}|}\sum_{j\in\xi_k^{(i)}} \nabla f_j(\mathbf{x}_k)$$

- **Blue arrows:** gradient computation time

Classic Method: Fully Synchronous SGD

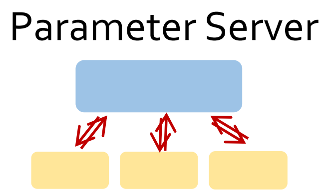
Execution pipeline:

1. Local stochastic gradients computation
2. Average local models across all nodes

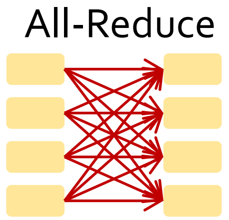


$$\frac{1}{m}\sum_{i=1}^{m}\mathbf{x}^{(i)}$$

Communication can be implemented via:



Li et al. **Scaling Distributed Machine Learning with the Parameter Server,** In *OSDI 2014*



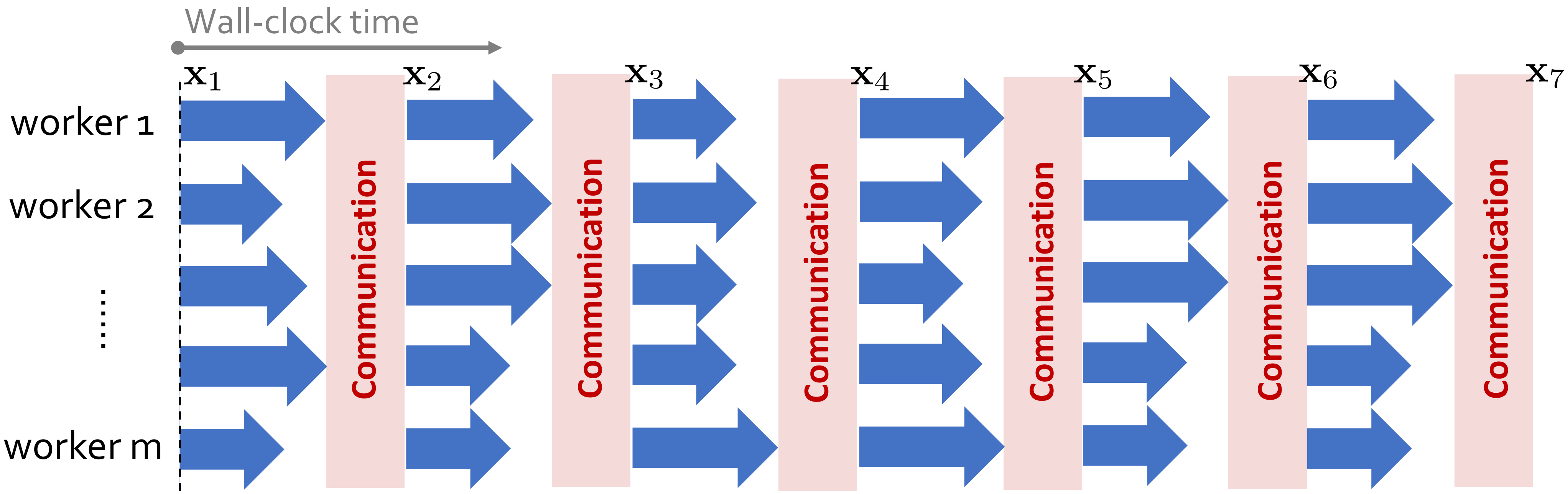
Goyal et al. **Accurate, Large Mini-Batch SGD: Training ImageNet in 1 Hour,** *ArXiv preprint 2017*

- **Blue arrows:** gradient computation time
- **Red blocks:** communication time

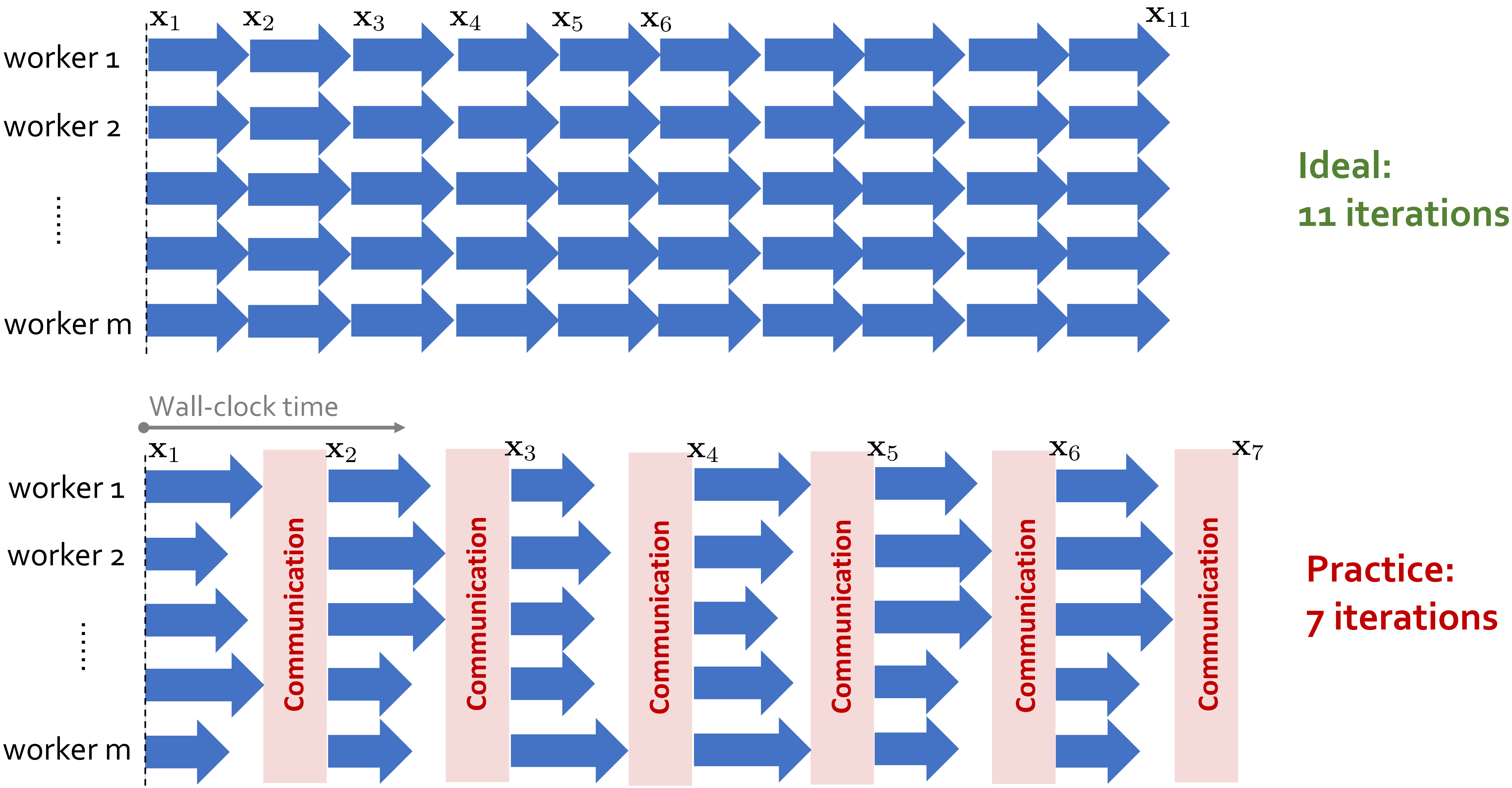
Classic Method: Fully Synchronous SGD

Execution pipeline:

1. Local stochastic gradients computation
2. Average local models across all nodes
3. Repeat the above steps until convergence



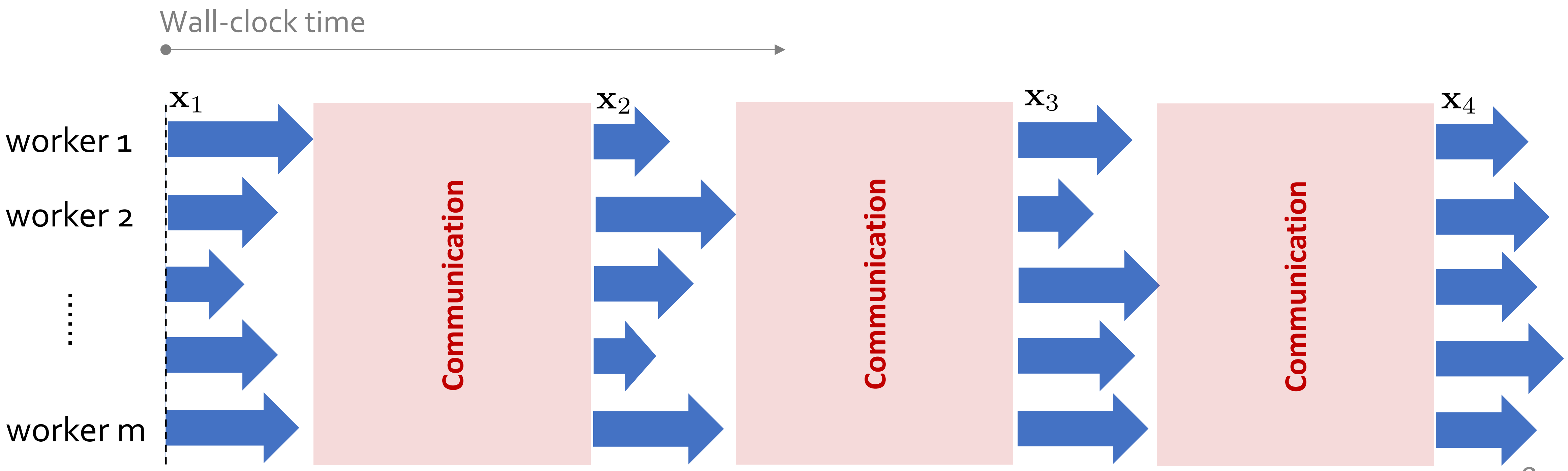
- **Blue arrows:** gradient computation time
- **Red blocks:** communication time



- **Blue arrows:** gradient computation time
- **Red blocks:** communication time

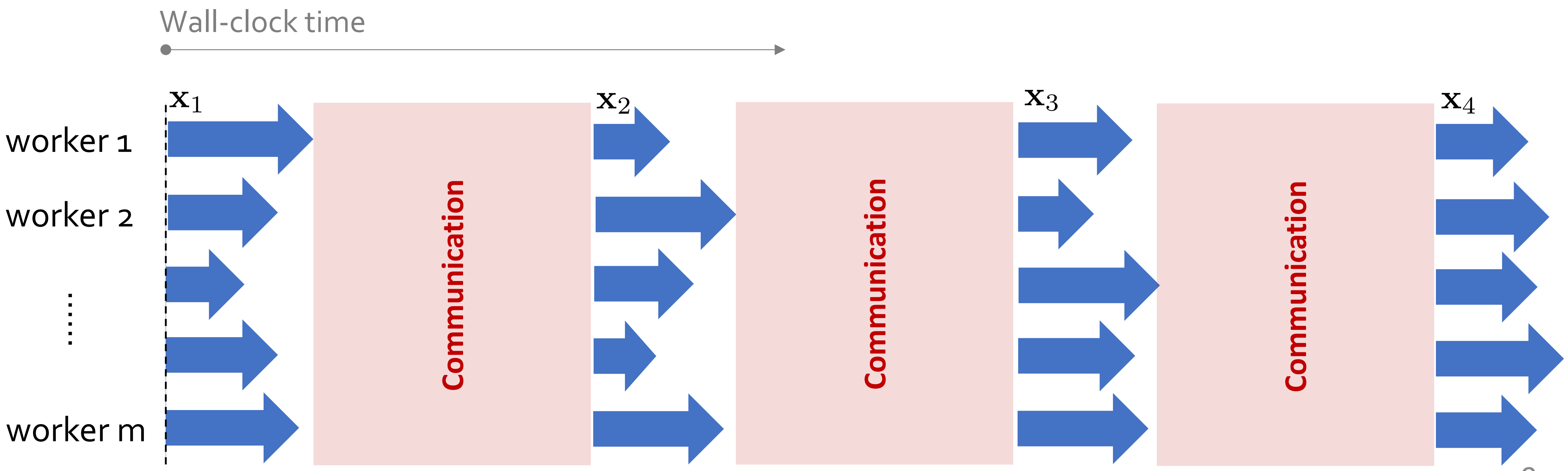
Communication is the Bottleneck in DNN Training

In deep neural nets training, the **communication time can be even larger than computation time.** [Harlap et al. ArXiv preprint 2018; Wang and Joshi, SysML 2019]



Communication is the Bottleneck in DNN Training

It is critical to develop **communication-efficient distributed SGD**



Previous works

- Periodic Averaging SGD / Local SGD

  Stich, S.U., **Local SGD Converges Fast and Communicates Little**, In *ICLR 2019*

Recall the Update Rule of Sync. SGD

Motivation of Periodic Averaging

Initial point $\mathbf{x}_1$

- **(Computation)** Worker nodes perform local steps

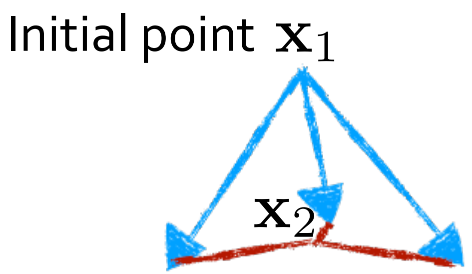
$$\mathbf{x}_k - \eta g(\mathbf{x}_k; \xi_k^{(i)})$$

Gradient step

Communication step

Recall the Update Rule of Sync. SGD

Motivation of Periodic Averaging



- **(Computation)** Worker nodes perform local steps

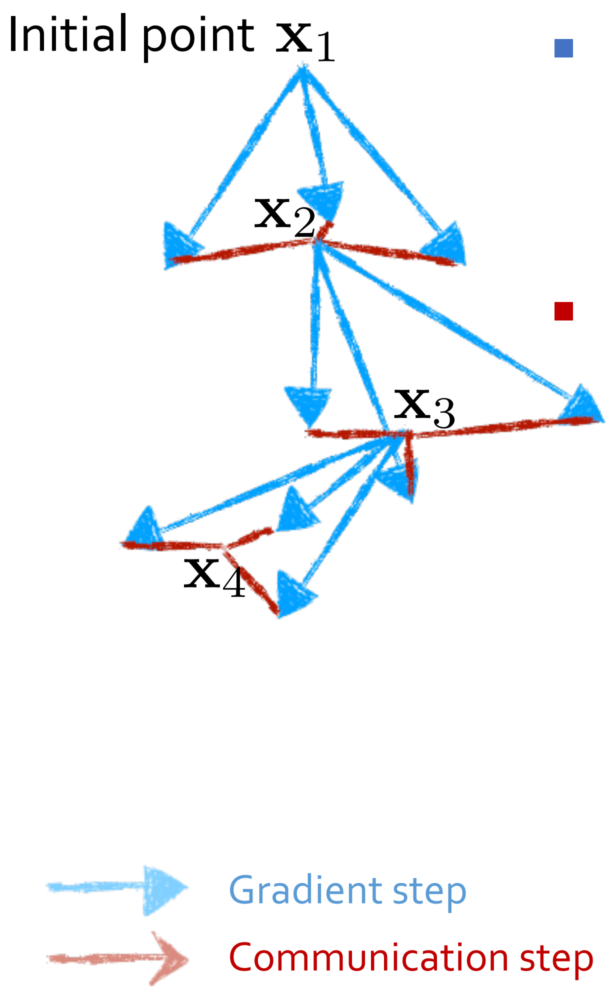
$$\mathbf{x}_k - \eta g(\mathbf{x}_k; \xi_k^{(i)})$$

- **(Communication)** Local models are averaged across all the nodes

$$\mathbf{x}_{k+1} = \frac{1}{m} \sum_{i=1}^{m} \left[\mathbf{x}_k - \eta g(\mathbf{x}_k; \xi_k^{(i)}) \right]$$

Recall the Update Rule of Sync. SGD

Motivation of Periodic Averaging



- **(Computation)** Worker nodes perform local steps

$$\mathbf{x}_k - \eta g(\mathbf{x}_k; \xi_k^{(i)})$$

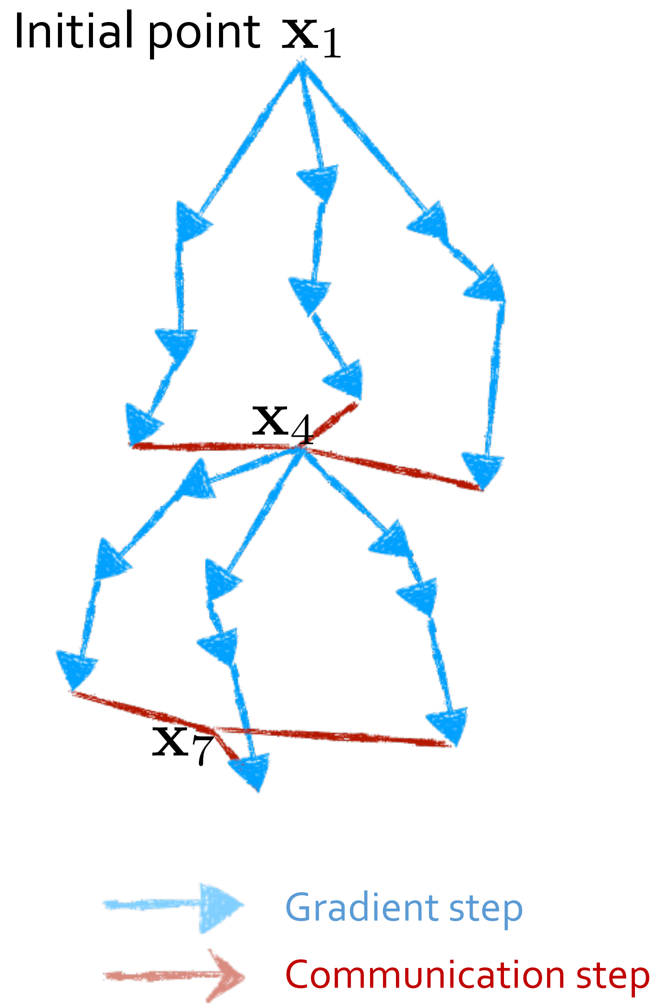
- **(Communication)** Local models are averaged across all the nodes

$$\mathbf{x}_{k+1} = \frac{1}{m} \sum_{i=1}^{m} \left[\mathbf{x}_k - \eta g(\mathbf{x}_k; \xi_k^{(i)}) \right]$$

Communicate at every iteration?

Periodic Averaging SGD (PASGD)

[Stich ICLR 2019]



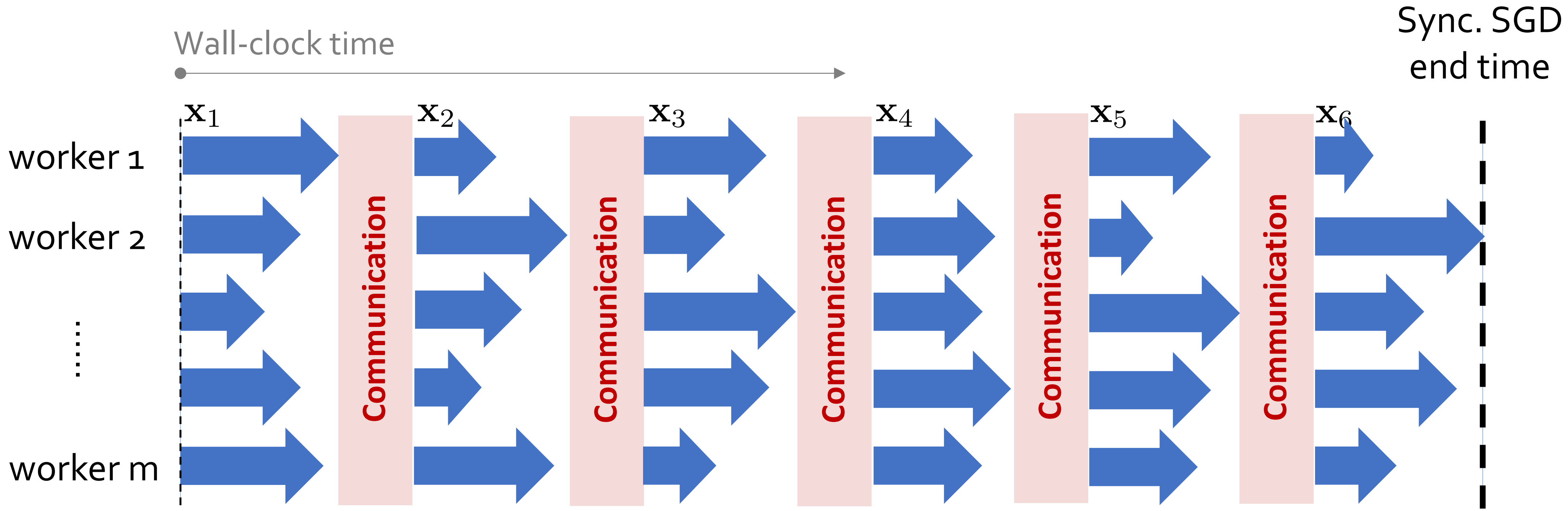
- Workers perform $\boldsymbol{\tau}$ local updates
- Local models are averaged **after every $\boldsymbol{\tau}$ local steps**

Communication period

Periodic Averaging Can Greatly Reduce Training Time

Blue arrow: gradient computation time

Red block: communication time



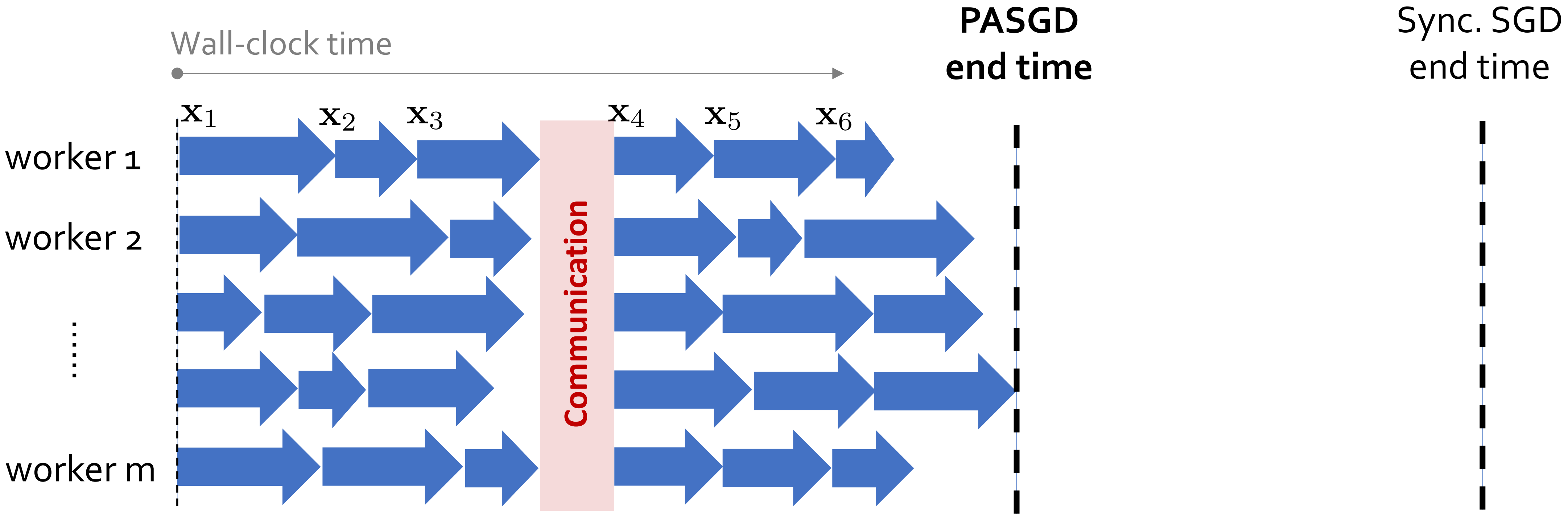
Periodic Averaging Can Greatly Reduce Training Time

✗ **Still converge?**

Blue arrow: gradient computation time

Red block: communication time

✓ Communication delay is reduced by τ times



Previous works

- Periodic Averaging SGD / Local SGD

  Stich, S.U., **Local SGD Converges Fast and Communicates Little**, In *ICLR 2019*

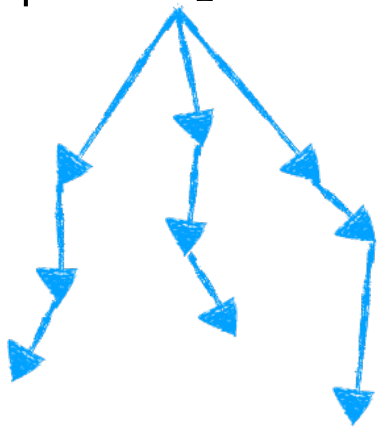
- Elastic Averaging SGD

  Zhang, S. et al., **Deep Learning with Elastic Averaging SGD**, In *NeurIPS 2015*

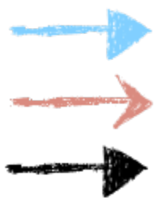
Elastic Averaging SGD (EASGD)

[Zhang et al. NeurIPS 2015]

Initial point $\mathbf{z}_1$



- **Key Idea:** Local models are guided towards to an anchor model after every τ local steps



Gradient step

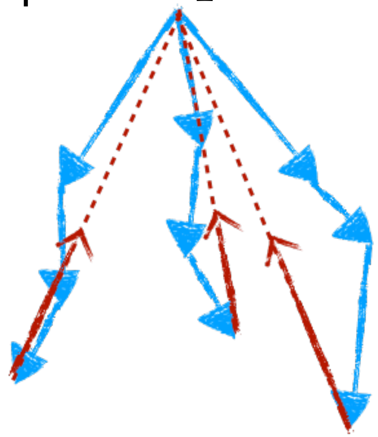
Communication step

Anchor model update

Elastic Averaging SGD (EASGD)

[Zhang et al. NeurIPS 2015]

Initial point $\mathbf{z}_1$

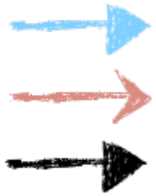


- **Key Idea:** Local models are guided towards to an anchor model after every τ local steps

- **Workers update rule:**

$$\mathbf{x}_{k+\tau}^{(i)} \leftarrow (1 - \alpha)\mathbf{x}_{k+\tau}^{(i)} + \alpha \mathbf{z}_k$$

Elasticity parameter



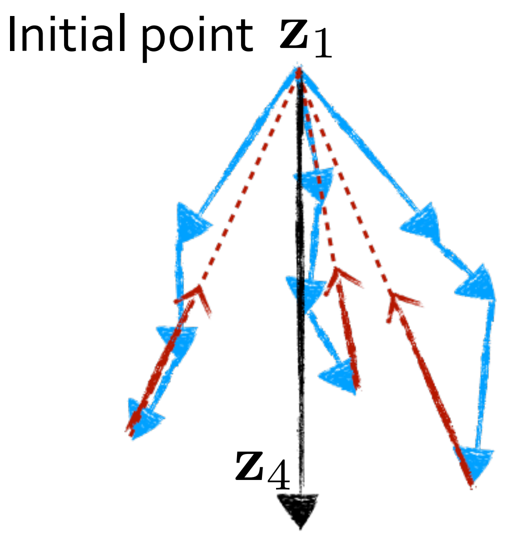
Gradient step

Communication step

Anchor model update

Elastic Averaging SGD (EASGD)

[Zhang et al. NeurIPS 2015]



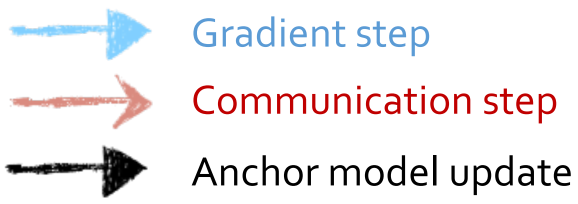
- **Key Idea:** Local models are guided towards to an anchor model after every τ local steps

- **Workers update rule:**

$$\mathbf{x}_{k+\tau}^{(i)} \leftarrow (1-\alpha)\mathbf{x}_{k+\tau}^{(i)} + \alpha \mathbf{z}_k$$

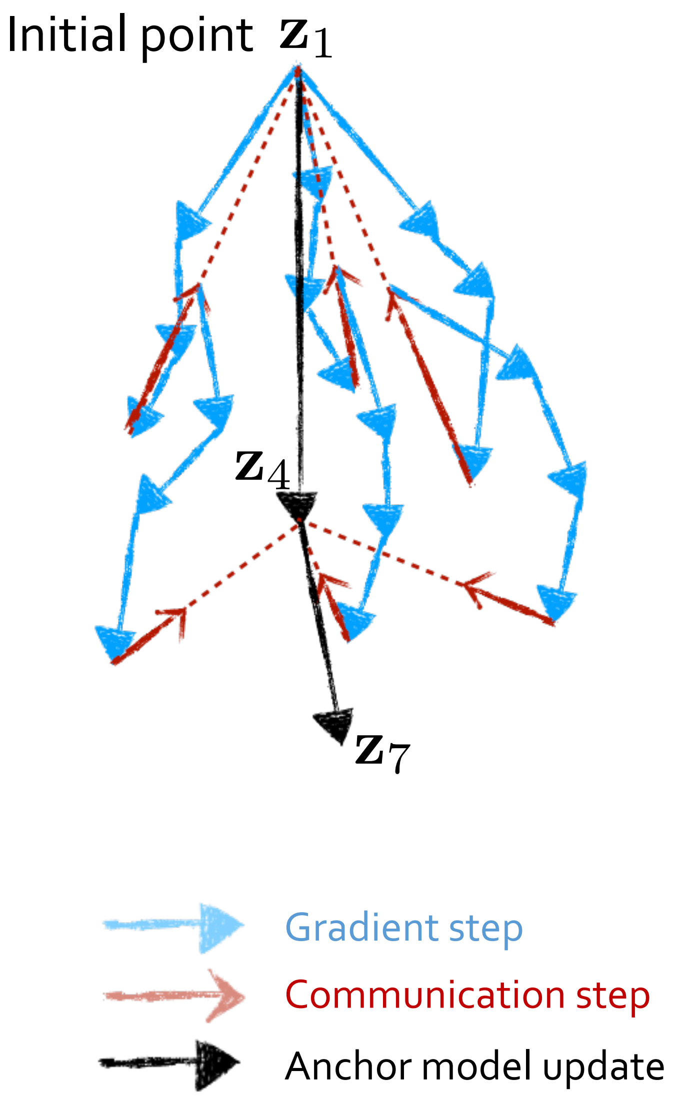
- **Anchor update rule:**

$$\mathbf{z}_{k+\tau} \leftarrow (1-m\alpha)\mathbf{z}_k + \alpha \sum_{i=1}^{m} \mathbf{x}_{k+\tau}^{(i)}$$



Elastic Averaging SGD (EASGD)

[Zhang et al. NeurIPS 2015]



- **Key Idea:** Local models are guided towards to an anchor model after every τ local steps

✔ Communication delay is reduced by τ times

✘ Convergence analysis

- Limited in **quadratic** objective function
- How to select the elasticity parameter α?

Previous works

- Periodic Averaging SGD / Local SGD

  Stich, S.U., **Local SGD Converges Fast and Communicates Little**, In *ICLR 2019*

- Elastic Averaging SGD

  Zhang, S. et al., **Deep Learning with Elastic Averaging SGD**, In *NeurIPS 2015*

- Decentralized Parallel SGD

  Lian, X. et al., **Can Decentralized Algorithms Outperform Centralized Algorithms?**, In *NeurIPS 2017 (oral)*

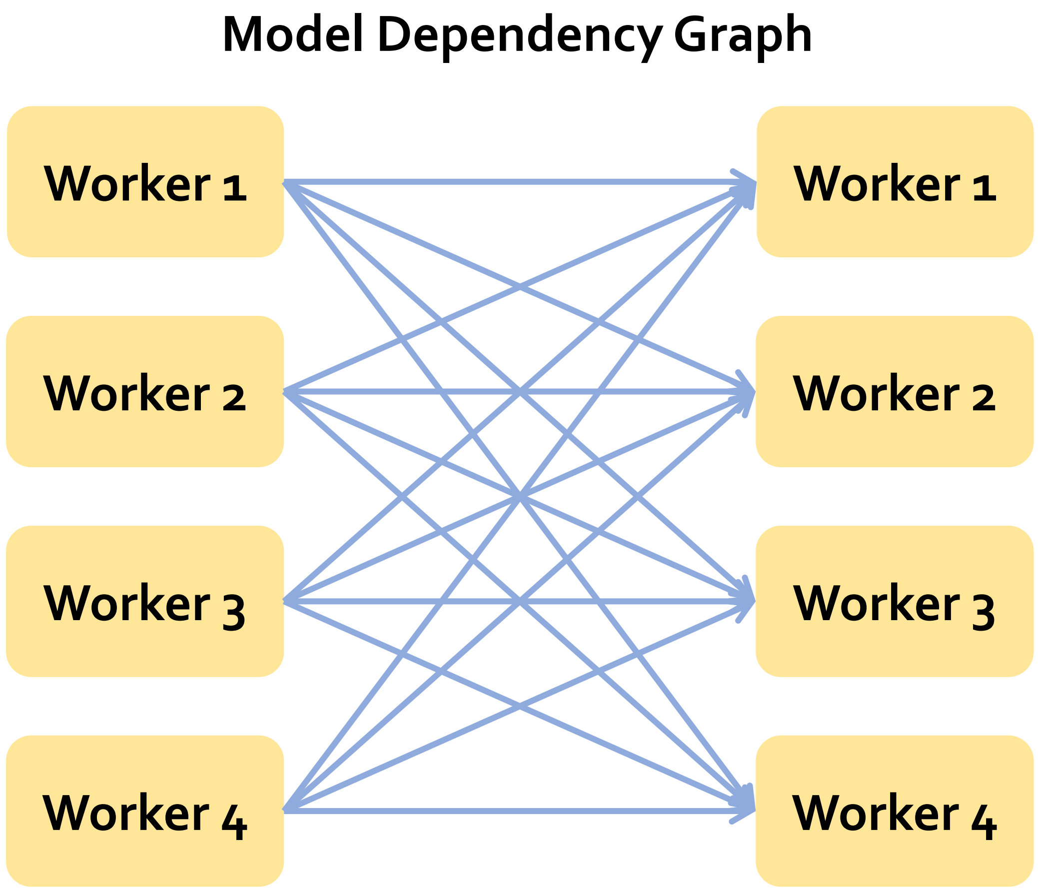
Model Dependencies in Fully Sync. SGD

Motivation of Decentralized Averaging

$$\mathbf{x}_{k+1} = \frac{1}{m}\sum_{i=1}^{m}\left[\mathbf{x}_k - \eta g(\mathbf{x}_k; \xi_k^{(i)})\right]$$

- In sync. SGD, each worker requires information from all others

Communicate with all others?



Decentralized Parallel SGD (D-PSGD)

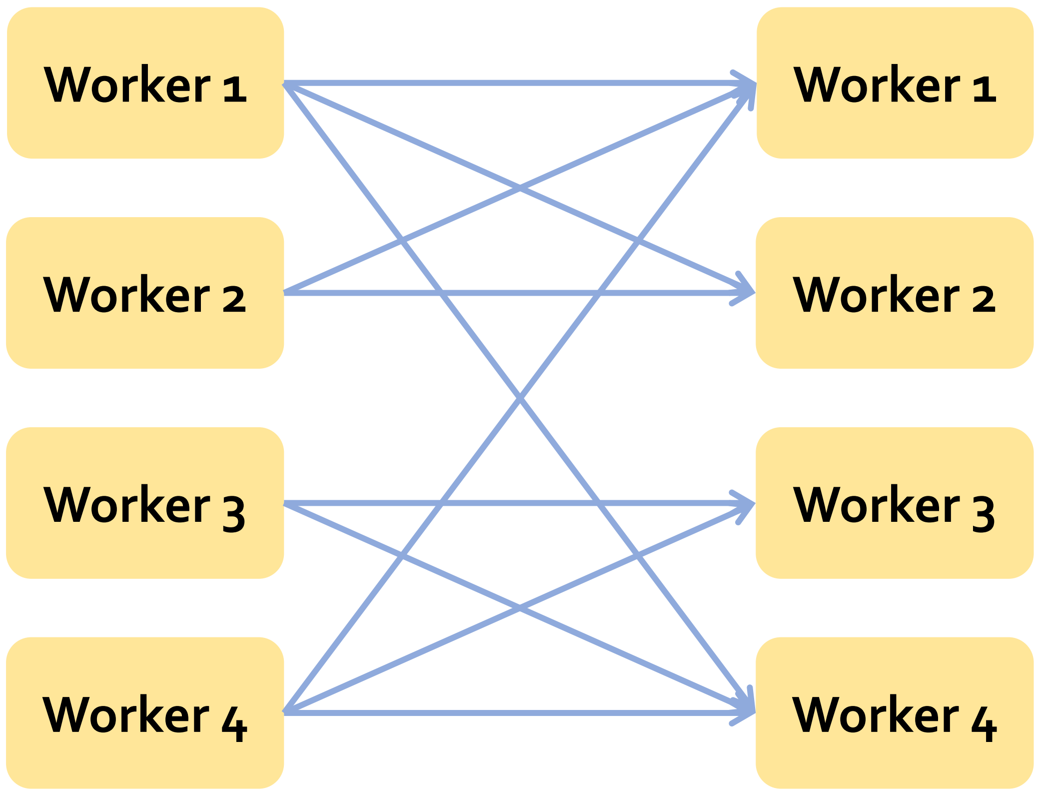
[Lian et al. NeurIPS 2017]

$$\mathbf{x}_{k+1}^{(i)} = \sum_{j \in \mathcal{N}_i} W_{ji} \left[\mathbf{x}_k^{(j)} - \eta g(\mathbf{x}_k^{(j)}; \xi_k^{(j)}) \right]$$

Key Idea:

- Each worker requires information from **very few neighbors**

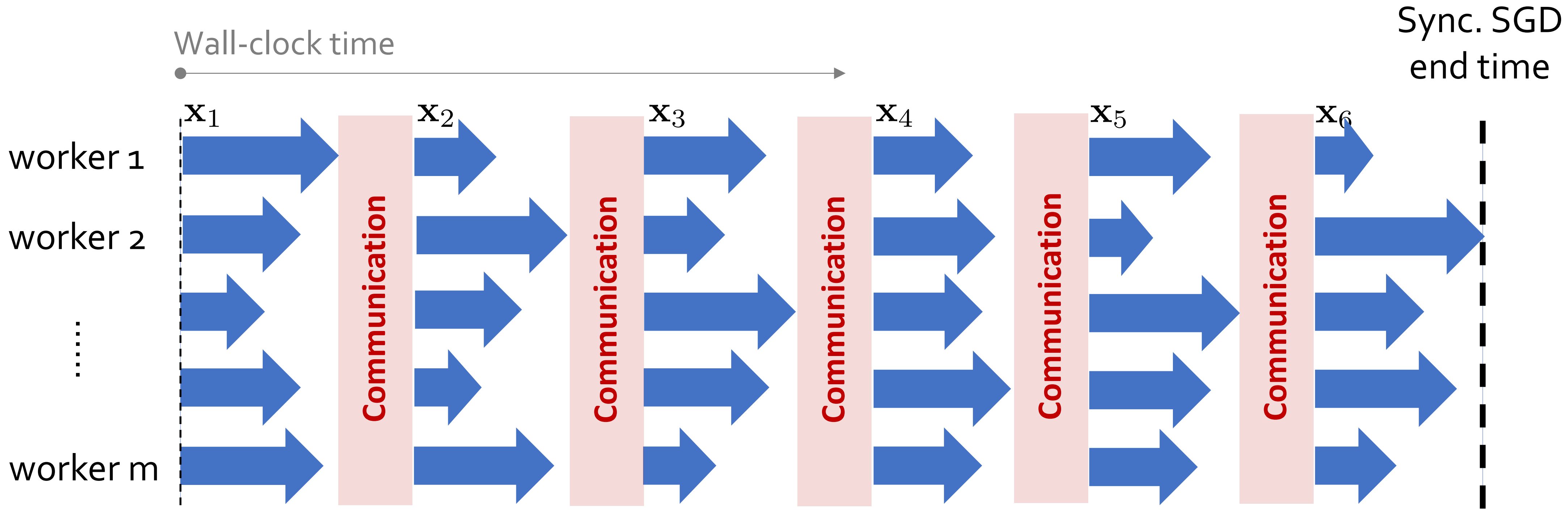
Model Dependency Graph



Decentralized Averaging Can Greatly Reduce Training Time

Blue arrow: gradient computation time

Red block: communication time

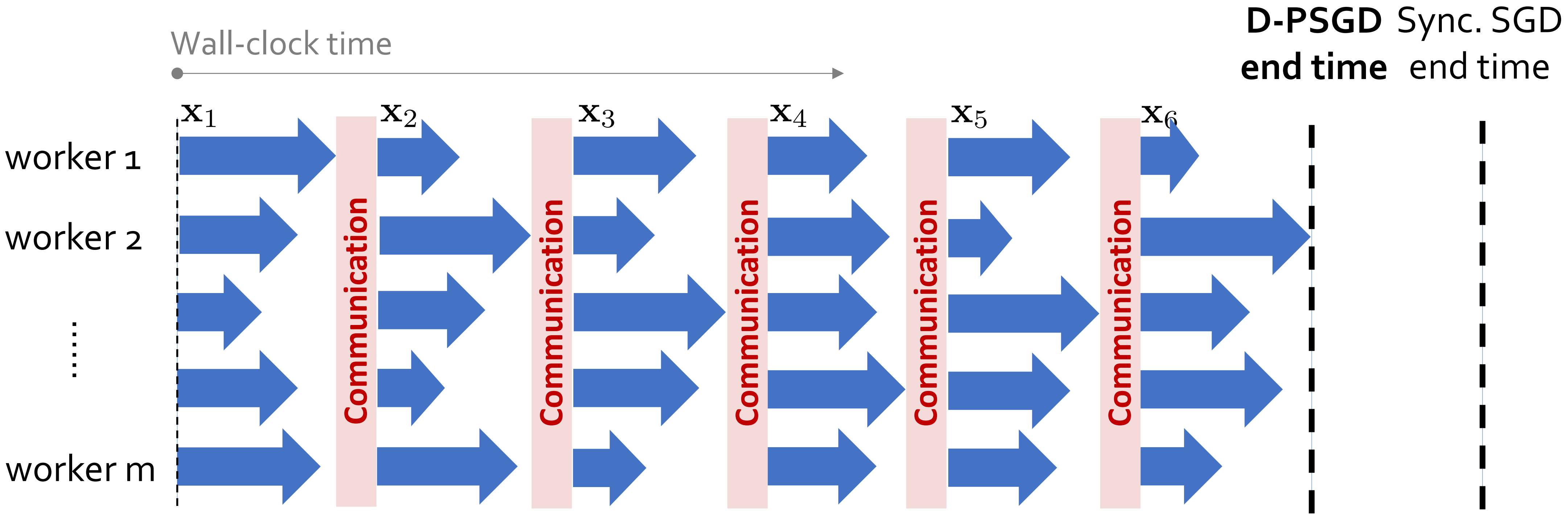


Decentralized Averaging Can Greatly Reduce Training Time

Blue arrow: gradient computation time

Red block: communication time

✔ Reduce communication complexity



Common Thread in Previous Works: Sparse Model-Averaging

- Periodic Averaging SGD / Local SGD

  Stich, S.U., **Local SGD Converges Fast and Communicates Little**, In *ICLR 2019*

- Elastic Averaging SGD

  Zhang, S. et al., **Deep Learning with Elastic Averaging SGD**, In *NeurIPS 2015*

Temporal-sparse communication

- Decentralized Parallel SGD

  Lian, X. et al., **Can Decentralized Algorithms Outperform Centralized Algorithms?**, In *NeurIPS 2*

Spatial-sparse communication

What else?

Common Thread in Previous Works: Sparse Model-Averaging

- Periodic Averaging SGD / Local SGD

  Limited in convex case;

  Stich, S.U., **Local SGD Converges Fast and Communicates Little**, In *ICLR 2019*

  Has strong bounded gradient assumption

- Elastic Averaging SGD

  Limited in quadratic case

  Zhang, S. et al., **Deep Learning with Elastic Averaging SGD**, In *NeurIPS 2015*

Temporal-sparse communication

- Decentralized Parallel SGD

  Only for $\tau = 1$

  Lian, X. et al., **Can Decentralized Algorithms Outperform Centralized Algorithms?**, In *NeurIPS 2*

Spatial-sparse communication

What else?

Our Solution: Cooperative SGD

A **General Framework** for

- Design
- Analysis

of **communication-efficient** distributed SGD algorithms

Key Elements in Cooperative SGD Framework

- Workers have different local model versions

Anchor models in EASGD

$$\mathbf{X}_k = [\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(m)}, \mathbf{z}_k^{(1)}, \ldots, \mathbf{z}_k^{(v)}]$$

- Local updates at m workers, no updates to auxiliary variables

$$\mathbf{G}_k = \left[g(\mathbf{x}_k^{(1)}), \ldots, g(\mathbf{x}_k^{(m)}), \mathbf{0}, \ldots, \mathbf{0}\right]$$

- Synchronization matrix

mixing matrix (spatial-sparsity)

$$\mathbf{W}_k = \begin{cases} \mathbf{W}, & k \bmod \tau = 0 \\ \mathbf{I}_{(m+v)\times(m+v)}, & \text{otherwise} \end{cases}$$

Communication period (temporal-sparsity)

Update Rule $\mathbf{X}_{k+1} = (\mathbf{X}_k - \eta \mathbf{G}_k)\mathbf{W}_k$

Previous Algorithms Are Just Special Cases

mixing matrix (spatial-sparsity)

$$\mathcal{A}(\tau, \mathbf{W}, v)$$

Communication period (temporal-sparsity)

# of auxiliary variables

- $\mathcal{A}(\tau, \mathbf{1}\mathbf{1}^T/m, 0)$ Periodic Averaging SGD / Local SGD
- $\mathcal{A}(\tau, \mathbf{W}_\alpha, 1)$ Elastic Averaging SGD
- $\mathcal{A}(1, \mathbf{W}, 0)$ Decentralized Parallel SGD
- $\mathcal{A}(1, \mathbf{1}\mathbf{1}^T/m, 0)$ Fully Synchronous SGD

Many more algorithms can be designed by varying hyper-parameters!

Our Solution: Cooperative SGD

A **General Framework** for

- Design
- Analysis   Unified convergence analysis for all algorithms

of **communication-efficient** distributed SGD algorithms

Assumptions

Identical to sync. SGD analysis

(A1) Lipschitz smooth: $\|\nabla F(\mathbf{x}) - \nabla F(\mathbf{y})\| \leq L \|\mathbf{x} - \mathbf{y}\|$

(A2) Unbiased gradient estimation: $\mathbb{E}_{\xi|\mathbf{x}}[g(\mathbf{x};\xi)] = \nabla F(\mathbf{x})$

(A3) Bounded variance: $\mathbb{E}_{\xi|\mathbf{x}}\left[\|g(\mathbf{x};\xi) - \nabla F(\mathbf{x})\|^2\right] \leq \beta \|\nabla F(\mathbf{x})\|^2 + \sigma^2$

Assumptions (cont'd)

Constraints on mixing matrix

Cooperative SGD

$$\mathbf{X}_{k+1} = [\mathbf{X}_k - \eta \mathbf{G}_k]\mathbf{W}_k$$

mixing matrix

$$\mathbf{W}_k = \begin{cases} \mathbf{W} & \text{if } k \bmod \tau = 0 \\ \mathbf{I} & \text{otherwise} \end{cases}$$

Requirements on mixing matrix:

- Symmetric and doubly stochastic

$$\zeta = \max\{|\lambda_2(\mathbf{W})|, |\lambda_{m+v}(\mathbf{W})|\} < 1$$

Larger for sparser topology

Example:

- Fully connected: $\mathbf{W} = \mathbf{J}$ $\quad \zeta = 0$
- Ring topology: $\mathbf{W} = \mathbf{W}_{\text{ring}}$ $\quad 0 < \zeta < 1$
- Independent workers: $\mathbf{W} = \mathbf{I}$ $\quad \zeta = 1$

Preliminaries: Simplification of the Update Rule

Update rule: $$\mathbf{X}_{k+1} = [\mathbf{X}_k - \eta \mathbf{G}_k]\mathbf{W}_k$$

Multiplying one vector on both sides,

$$\mathbf{X}_{k+1}\frac{\mathbf{1}_{m+v}}{m+v} = [\mathbf{X}_k - \eta \mathbf{G}_k]\mathbf{W}_k\frac{\mathbf{1}_{m+v}}{m+v}$$

$$= [\mathbf{X}_k - \eta \mathbf{G}_k]\boxed{\frac{\mathbf{1}_{m+v}}{m+v}}$$

Because W_k is stochastic

Vector-form updates: $$\mathbf{X}_{k+1}\frac{\mathbf{1}_{m+v}}{m+v} = \mathbf{X}_k\frac{\mathbf{1}_{m+v}}{m+v} - \eta \mathbf{G}_k\frac{\mathbf{1}_{m+v}}{m+v}$$

Preliminaries: Simplification of the Update Rule

Vector-form updates: $\boxed{\mathbf{X}_{k+1}\frac{\mathbf{1}_{m+v}}{m+v}} = \mathbf{X}_k\frac{\mathbf{1}_{m+v}}{m+v} - \eta\boxed{\mathbf{G}_k\frac{\mathbf{1}_{m+v}}{m+v}}$

Averaged model

$\mathbf{u}_{k+1}$

Averaged gradient

$$\frac{1}{m+v}\sum_{i=1}^{m} g(\mathbf{x}_k^{(i)}; \xi_k^{(i)})$$

Update rule on average model: $\mathbf{u}_{k+1} = \mathbf{u}_k - \boxed{\frac{m}{m+v}\eta} \cdot \left[\frac{1}{m}\sum_{i=1}^{m} g(\mathbf{x}_k^{(i)}; \xi_k^{(i)})\right]$

Effective learning rate

η_{eff}

Preliminaries: Comparison to Fully Sync. SGD

- Cooperative SGD $\mathbf{u}_{k+1} = \mathbf{u}_k - \eta_{\text{eff}} \cdot \left[\frac{1}{m} \sum_{i=1}^{m} g\left(\mathbf{x}_k^{(i)}; \xi_k^{(i)}\right) \right]$
- Fully sync. SGD $\mathbf{x}_{k+1} = \mathbf{x}_k - \eta \left[\frac{1}{m} \sum_{i=1}^{m} g\left(\mathbf{x}_k; \xi_k^{(i)}\right) \right]$

The key differences:

- **local models are different! → Stochastic gradients are biased!**

Key Idea in the Proof

- According to the Lipschitz smooth assumption (A1), we have

$$F(\mathbf{u}_{k+1}) - F(\mathbf{u}_k) \leq \langle \nabla F(\mathbf{u}_k), \mathbf{u}_{k+1} - \mathbf{u}_k \rangle + \frac{L}{2} \left\| \mathbf{u}_{k+1} - \mathbf{u}_k \right\|^2$$

- Plugging in the update rule of cooperative SGD,

$$F(\mathbf{u}_{k+1}) - F(\mathbf{u}_k) \leq -\eta_{\text{eff}} \boxed{\left\langle \nabla F(\mathbf{u}_k), \frac{1}{m} \sum_{i=1}^{m} g(\mathbf{x}_k^{(i)}) \right\rangle} + \frac{\eta_{\text{eff}}^2 L}{2} \boxed{\left\| \frac{1}{m} \sum_{i=1}^{m} g(\mathbf{x}_k^{(i)}) \right\|^2}$$

"Similarity"
Lower-bounded

"Noise"
Upper-bounded

Key Idea in the Proof

LEMMA

When learning rate is fixed and satisfies certain constraints:

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\|\nabla F(\mathbf{u}_k)\|^2\right] \leq \frac{2[F(\mathbf{u}_1) - F_{\text{inf}}]}{\eta_{\text{eff}} K} + \frac{\eta_{\text{eff}} L \sigma^2}{m} + \frac{L^2}{Km}\sum_{k=1}^{K}\mathbb{E}\left[\|\mathbf{X}_k(\mathbf{J} - \mathbf{I})\|_F^2\right]$$

Optimization Error

Fully sync SGD Error

Network Error

"Price" pay for comm. reduction

Recall that:

- Fully synchronization matrix: $\mathbf{J} = \frac{\mathbf{1}\mathbf{1}^\top}{m}$
- $\mathbf{X}_k \mathbf{J} = [\mathbf{u}_k, \mathbf{u}_k, \ldots, \mathbf{u}_k]$ represents the averaged model

Main Result: Discrepancies Among Local Models Hurt Convergence.

When learning rate is fixed and satisfies certain constraints:

$$\mathbb{E}\left[\frac{1}{K}\sum_{k=1}^{K}\left\|\nabla F(\mathbf{u}_k)\right\|^2\right] \leq \frac{2[F(\mathbf{u}_1) - F_{\inf}]}{\eta_{\text{eff}} K} + \frac{\eta_{\text{eff}} L \sigma^2}{m} + \eta^2 L^2 \sigma^2 \left(\frac{1+\zeta^2}{1-\zeta^2}\tau - 1\right)$$

Optimization Error

Fully sync SGD Error

Network Error

"Price" pay for comm. reduction

Recall that:

- Sparsity of comm. Topology ζ
- Communication period τ

Recover Synchronous SGD:

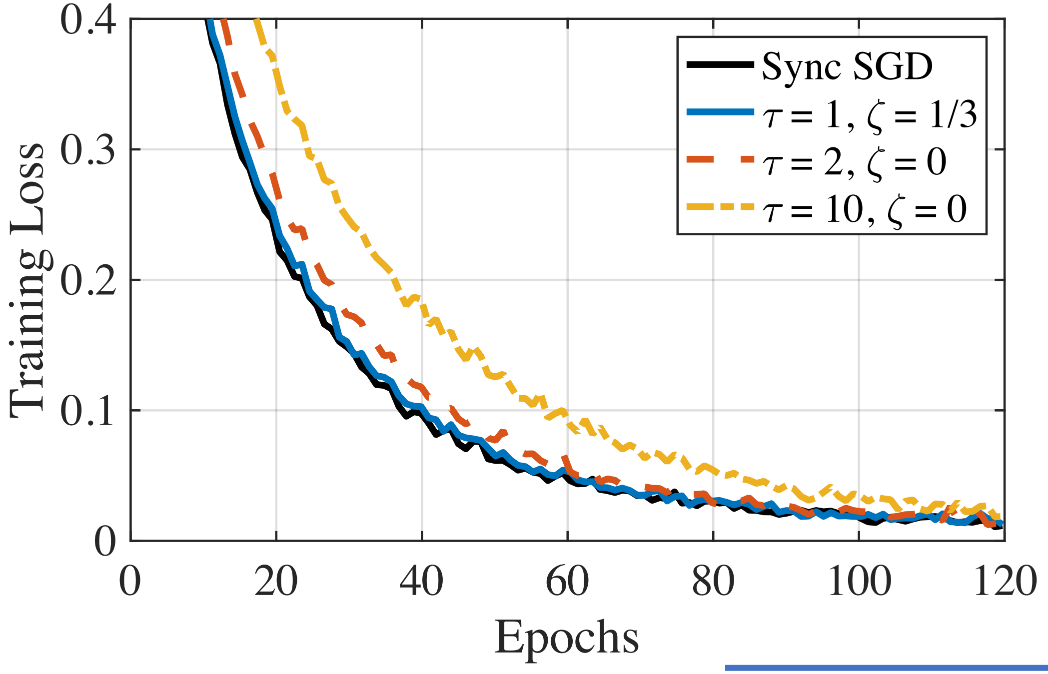
$$v = 0, \zeta = 0, \tau = 1$$

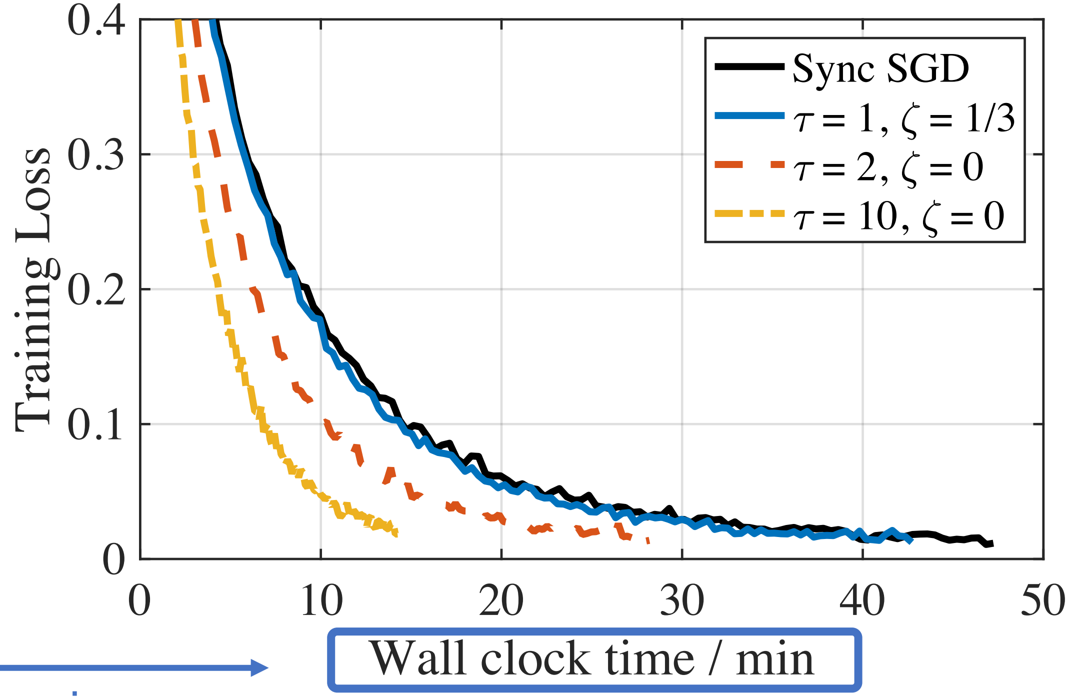
Network error = 0

Advantages of Cooperative SGD

Relax synchronization among local models may increase the convergence error

But it can significantly reduce the communication overhead





Change x-axis

Novel Analyses of Existing Algorithms

- Rely on strong bounded gradient assumption.

Periodic Averaging SGD: $\mathcal{A}(\tau, \mathbf{1}\mathbf{1}^T/m, 0)$

- Additional network error is proportional to $\boldsymbol{\tau - 1}$ instead of $\boldsymbol{\tau^2}$

- Only for quadratic case.
- Strong assumptions.

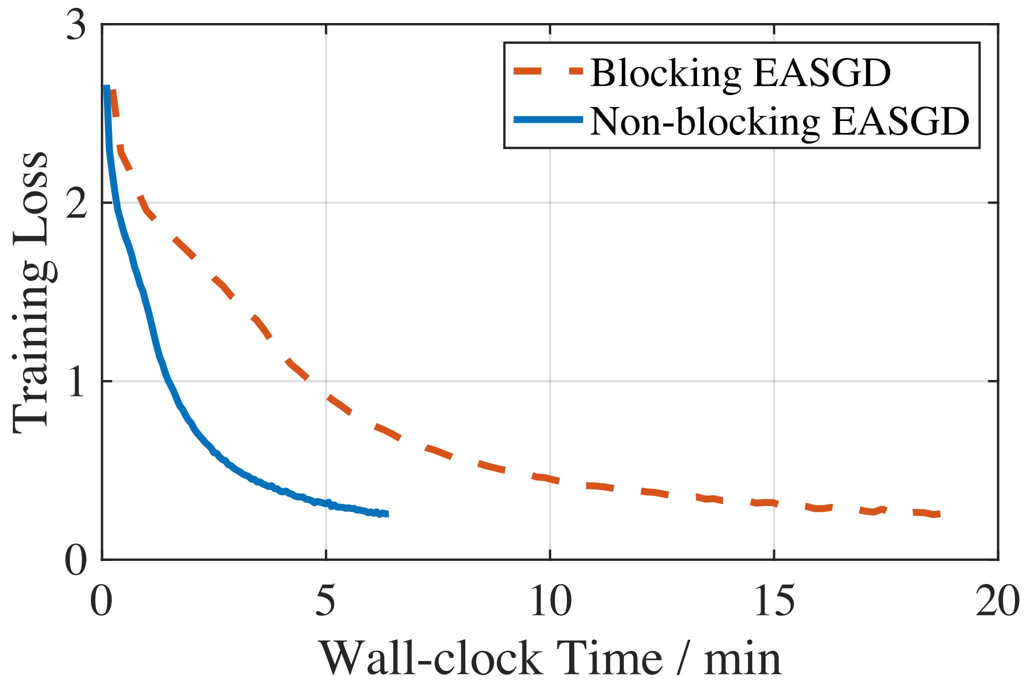
Elastic Averaging SGD: $\mathcal{A}(\tau, \mathbf{W}_\alpha, 1)$

- The **first convergence analysis** on non-convex objectives
- We show that there is a **best value of elasticity parameter $\boldsymbol{\alpha}$** which can yields lowest error floor at convergence

Novel Analyses of Existing Algorithms

Elastic Averaging SGD: ✗ ▪ Only use periodic avg. strategy

- By non-blocking execution, it achieves nearly **3x speedup** over the blocking counterpart.



VGG-16, CIFAR-10
8 workers
Pytorch 1.0 + Gloo

Conclusions

A general framework for the design and analysis of comm-efficient SGD!

- Instead of averaging gradient, average local models
- Local models can be synchronized infrequently or in a sparse way

A unified convergence analysis for non-convex objective functions!

- Discrepancies among local models may hurt convergence
- But the communication efficiency is significantly improved

Thanks for attention!