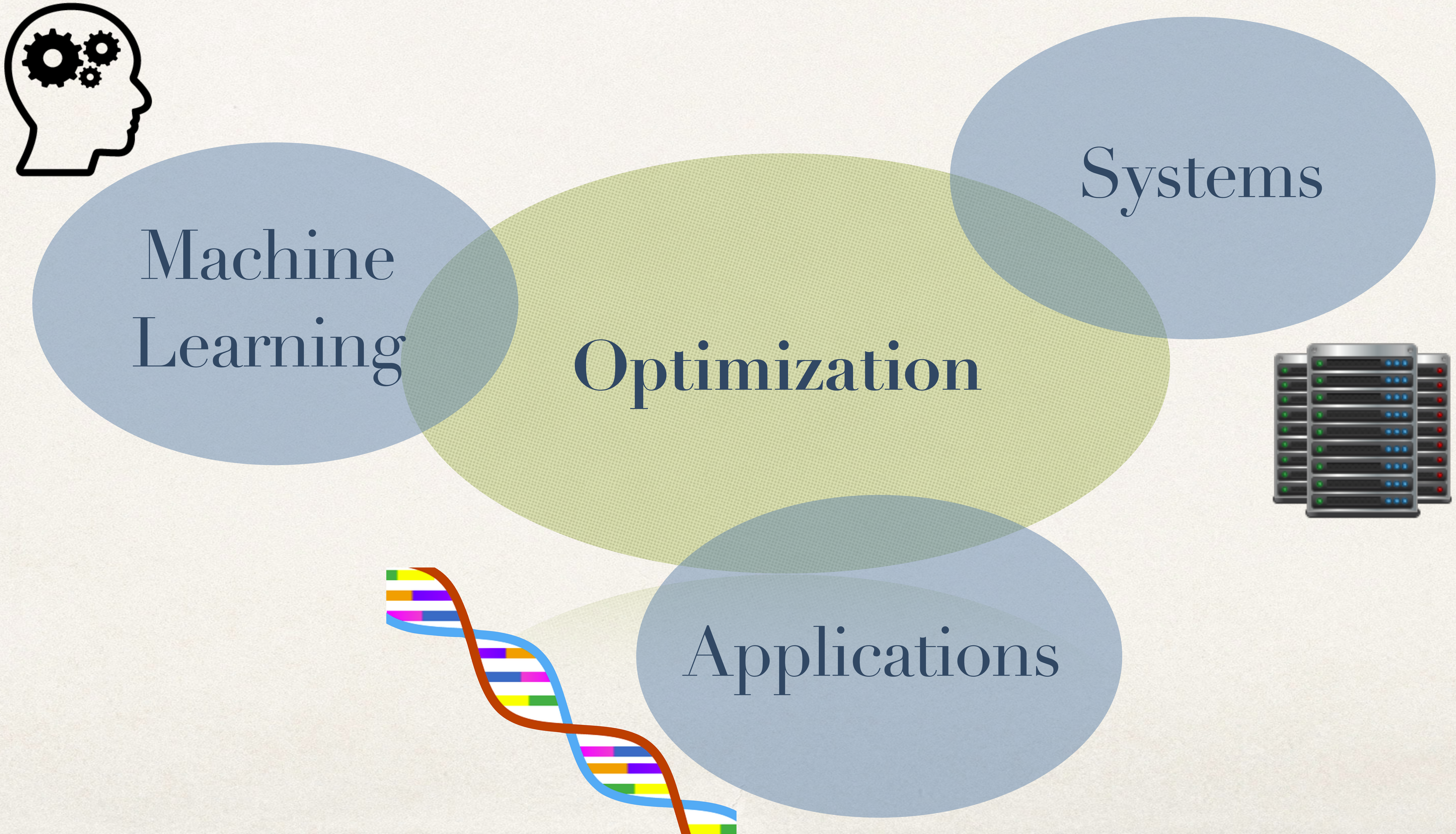
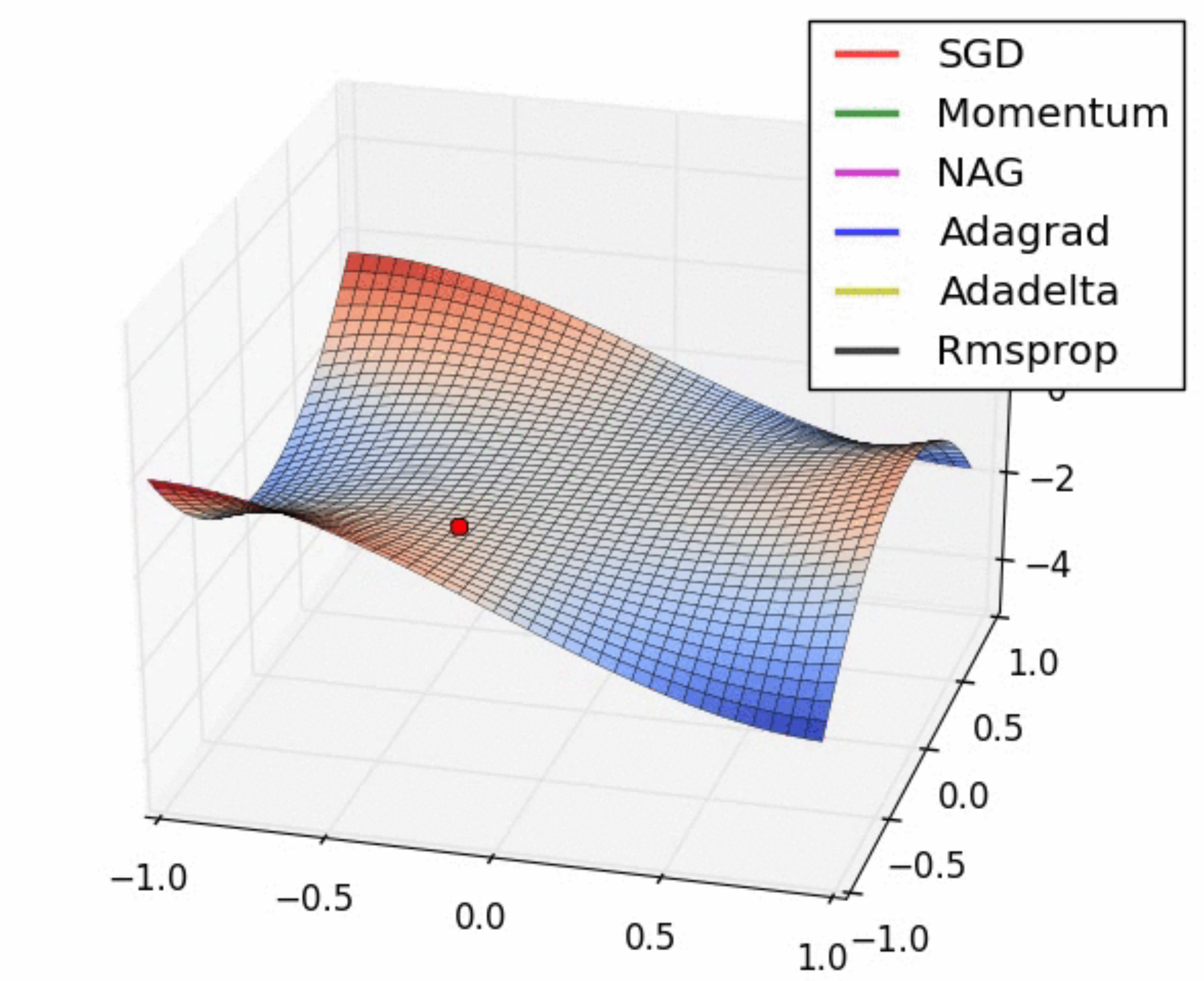
Optimization
for Machine Learning in Practice I

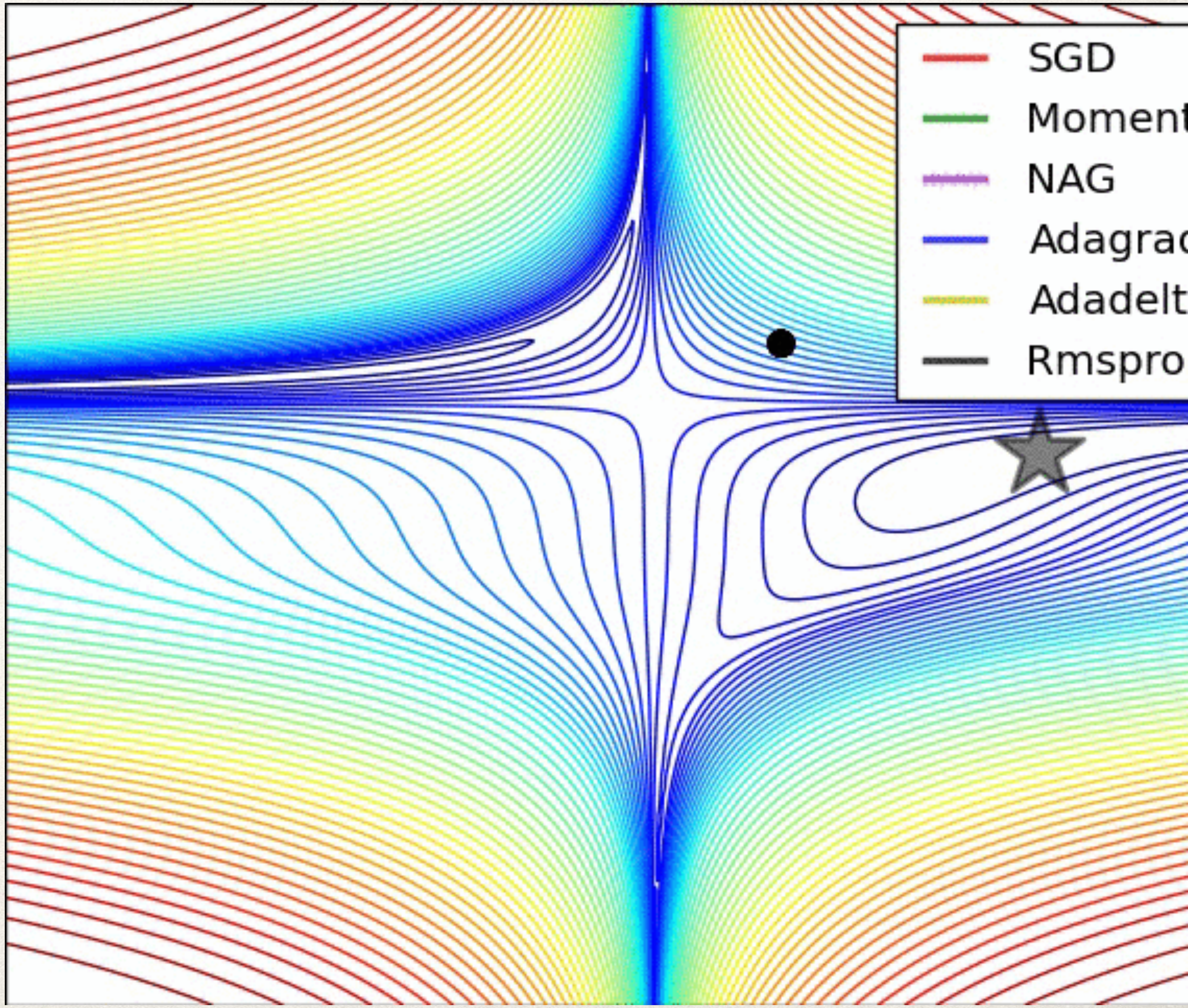
Martin Jaggi
EPFL
Machine Learning and Optimization Laboratory
mlo.epfl.ch

Where are we?



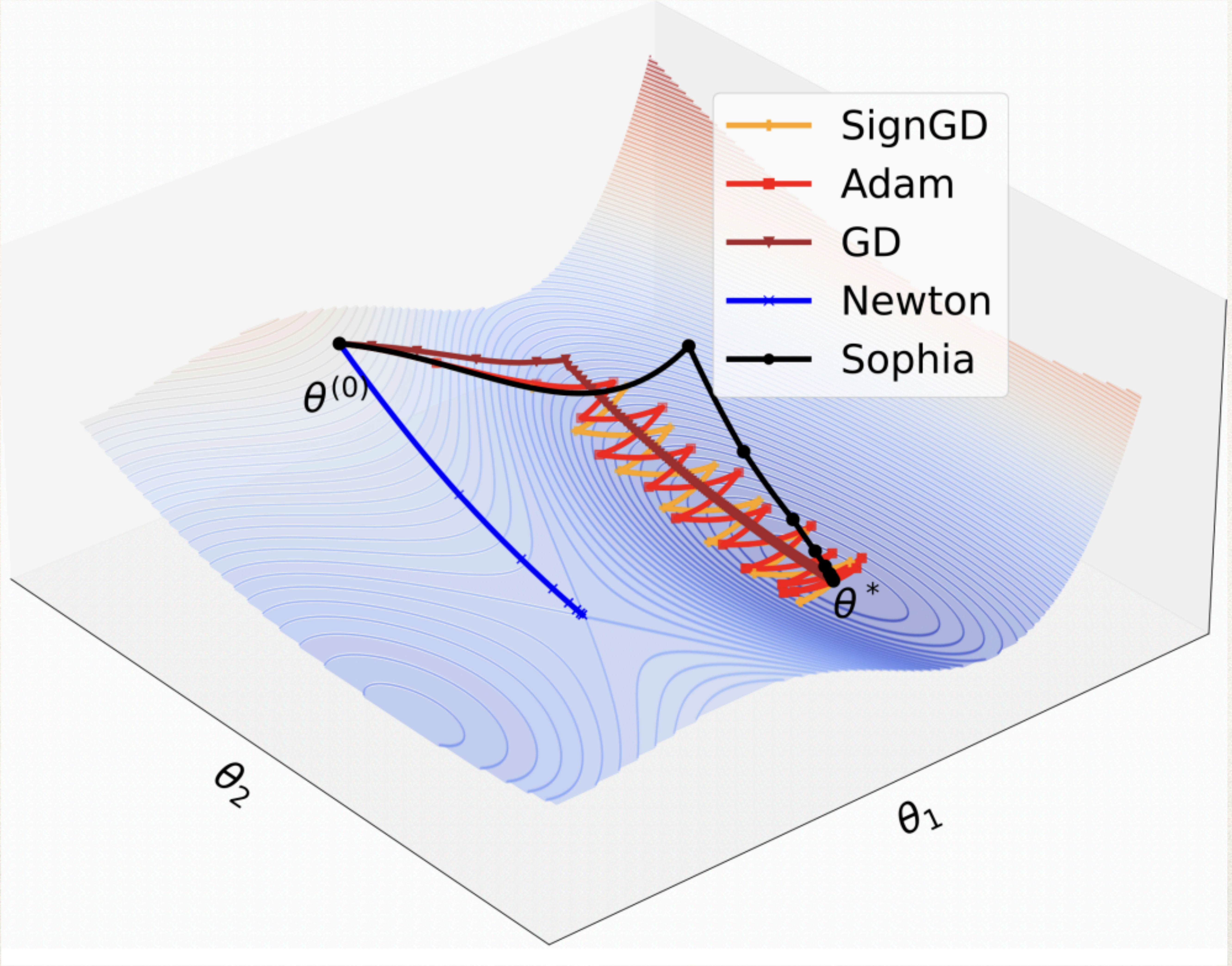
Practical comparison of algorithms





https://imgur.com/a/Hqolp#2dKCQHh

Practical comparison of algorithms



Trends - General

- **Foundation models / LLMs**
- **Custom AI hardware & systems**
- **Federated or decentralized training**
- **Privacy**
- **Interpretability**
- **trust, fairness and robustness in ML**

  (e.g. robust & secure against adversaries)

Optimization is a key element of most above topics

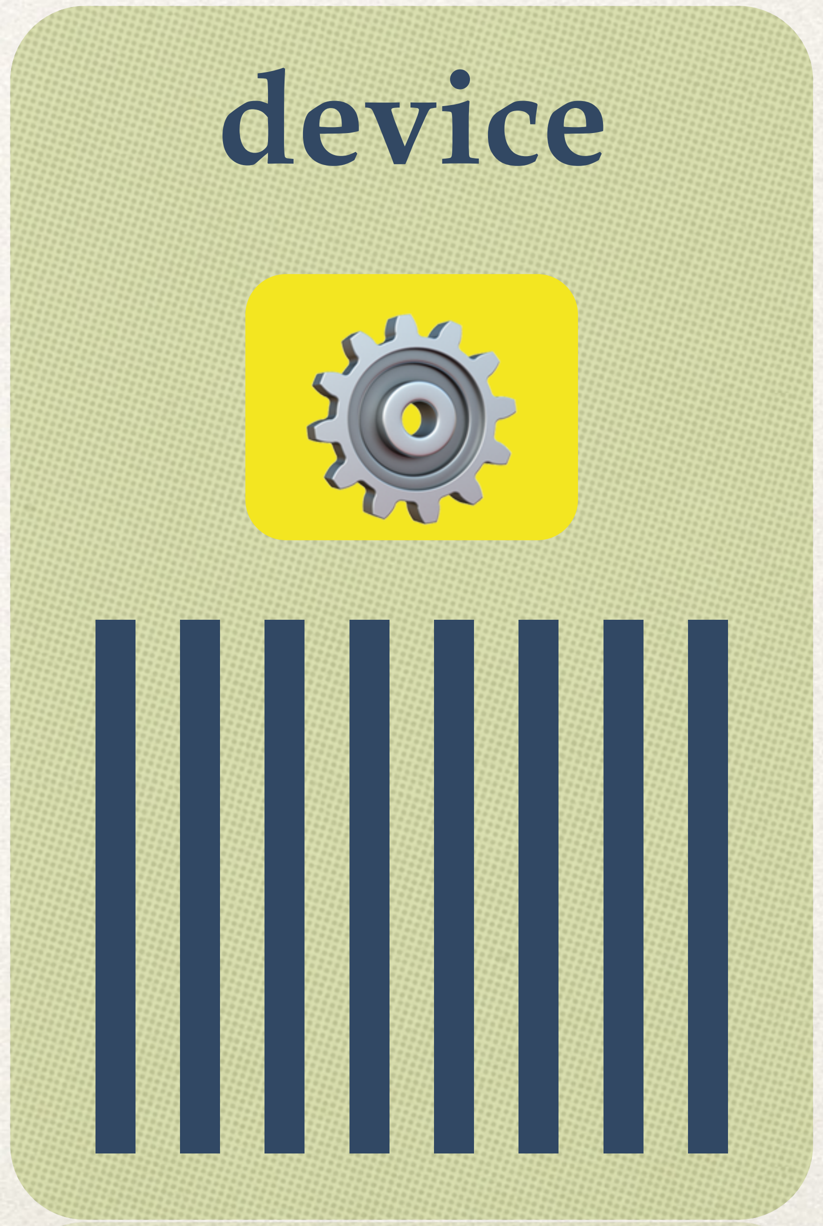
ML Training

$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}) = \frac{1}{|data|} \sum_{i \in data} f_i(\boldsymbol{x})$$

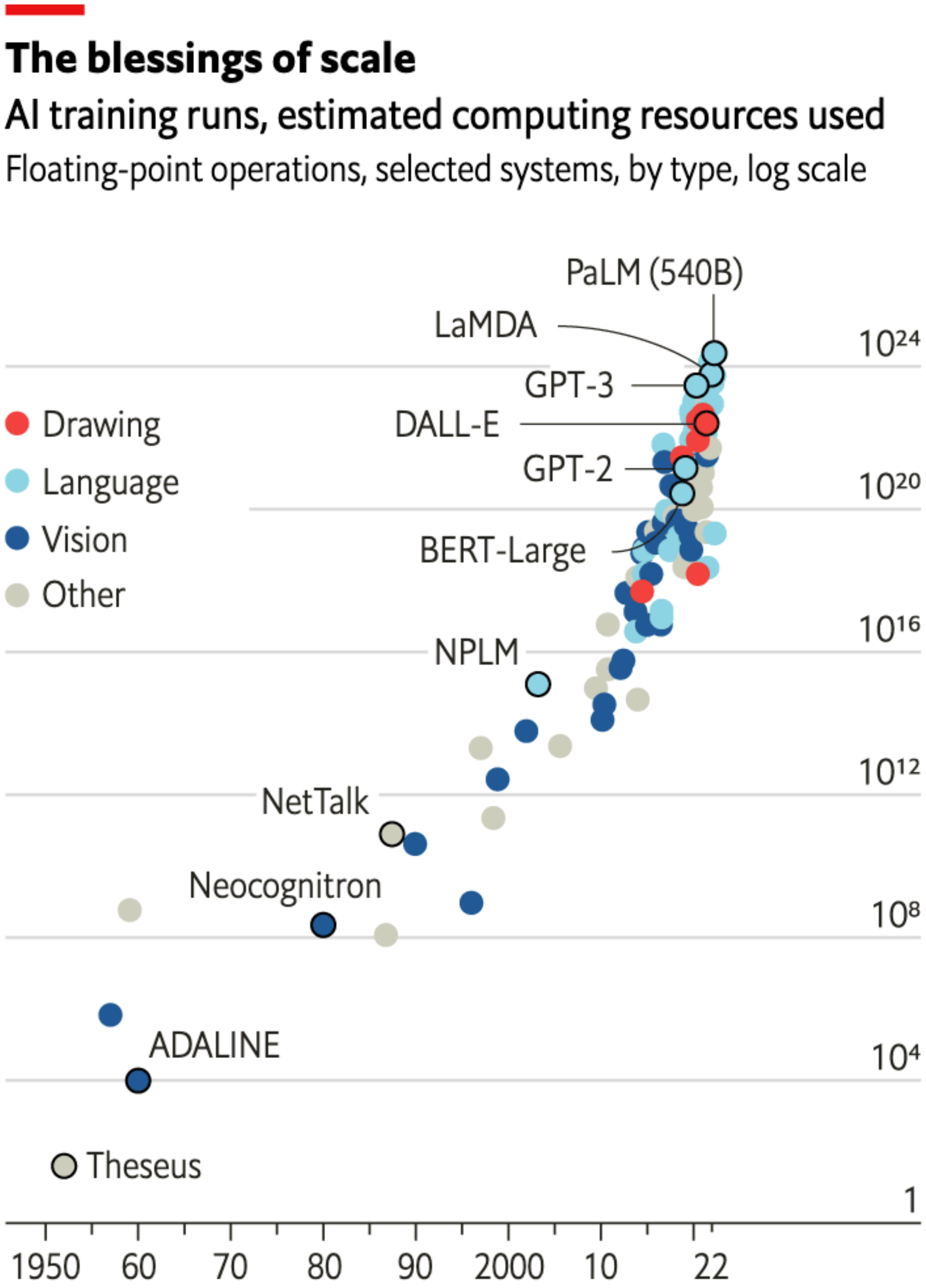
Training algorithms: SGD-based

$$i_t \sim \text{Uniform}(1, |data|)$$

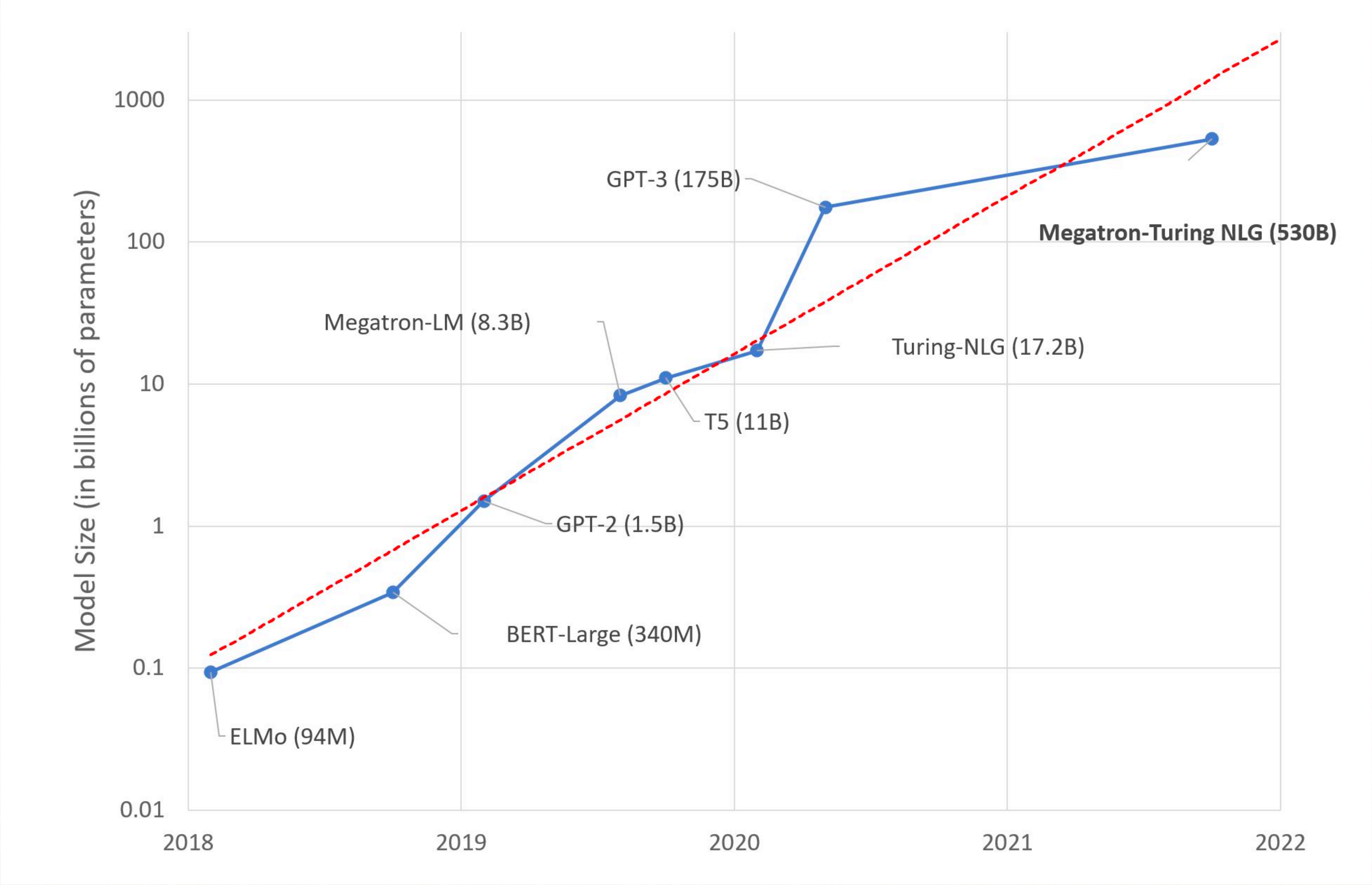
$$\boldsymbol{x}_{t+1} := \boldsymbol{x}_t - \gamma_t \nabla f_{i_t}(\boldsymbol{x}_t)$$



Training: Comp

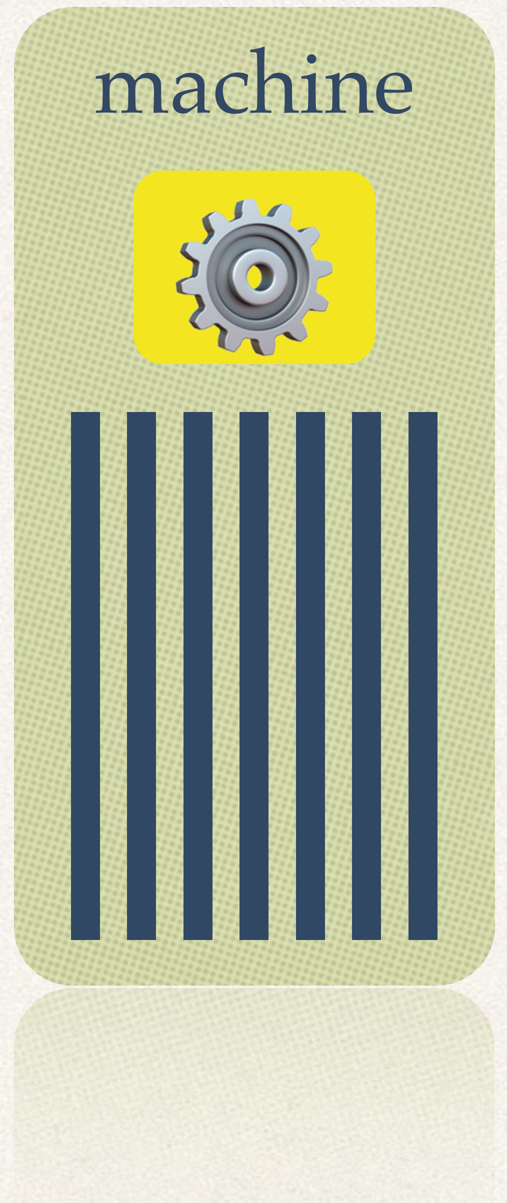


Model Sizes (Transformer Models)

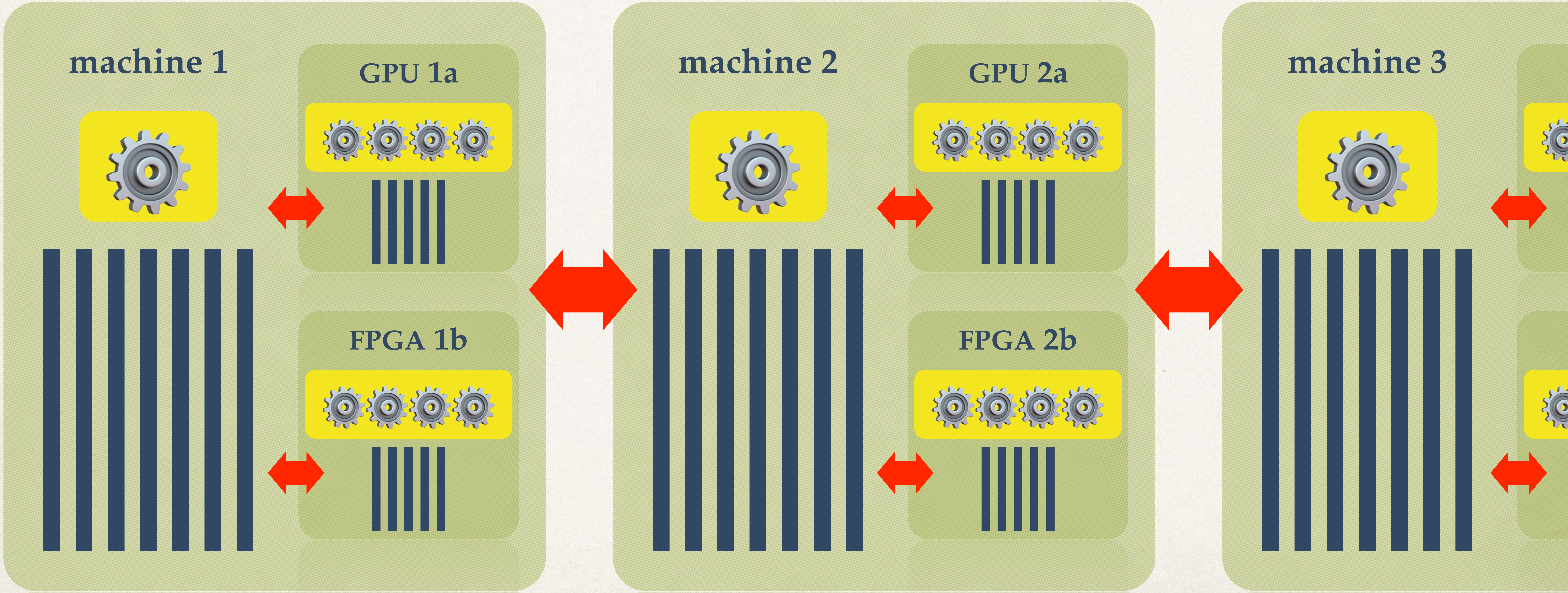


https://www.arxiv-vanity.com/papers/2104.04473/

Systems …then



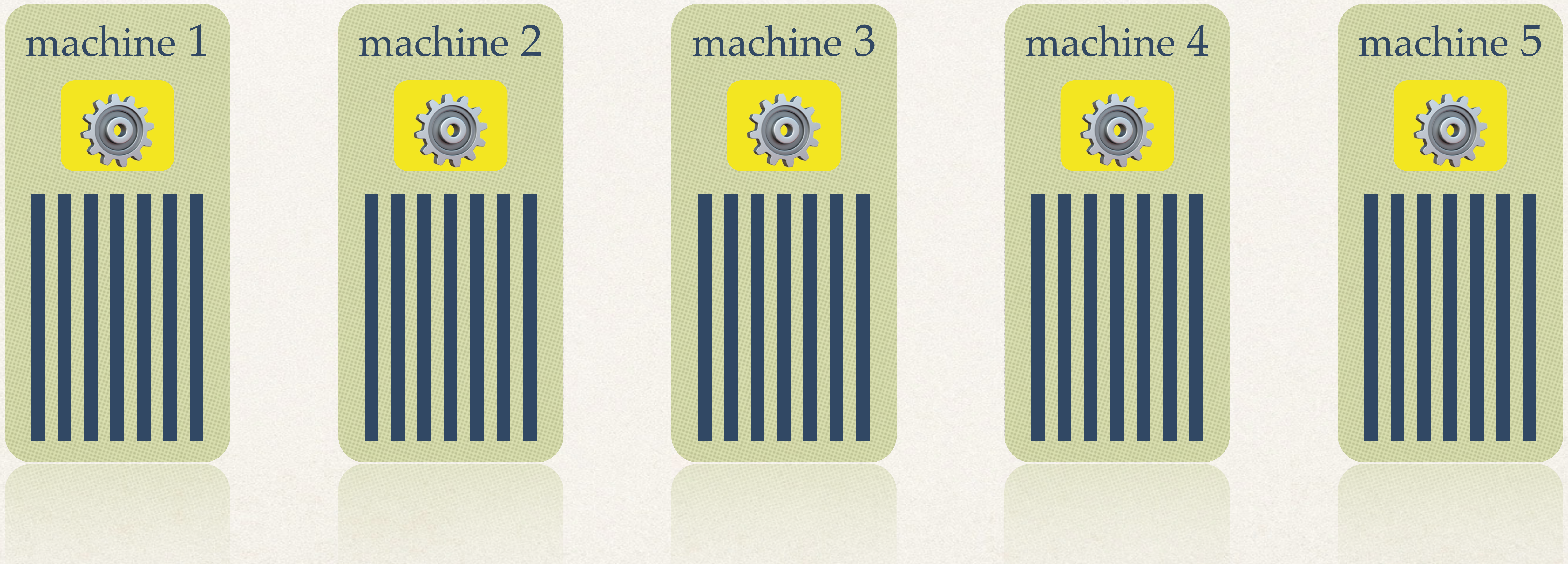
Systems …now



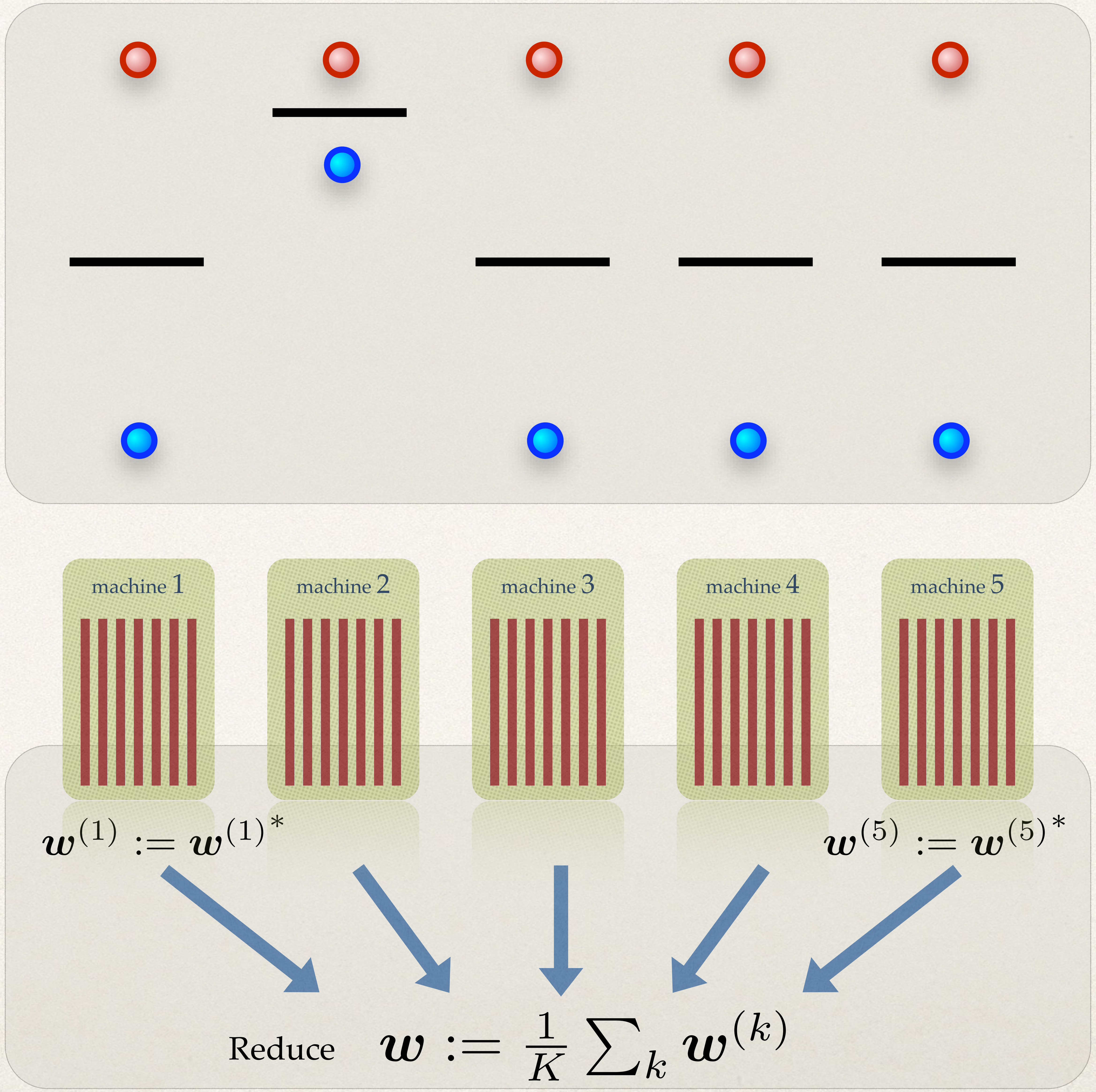
What are the fundamental limits of **parallelizing** the training of neural networks?

1 Parallel & Distributed Training

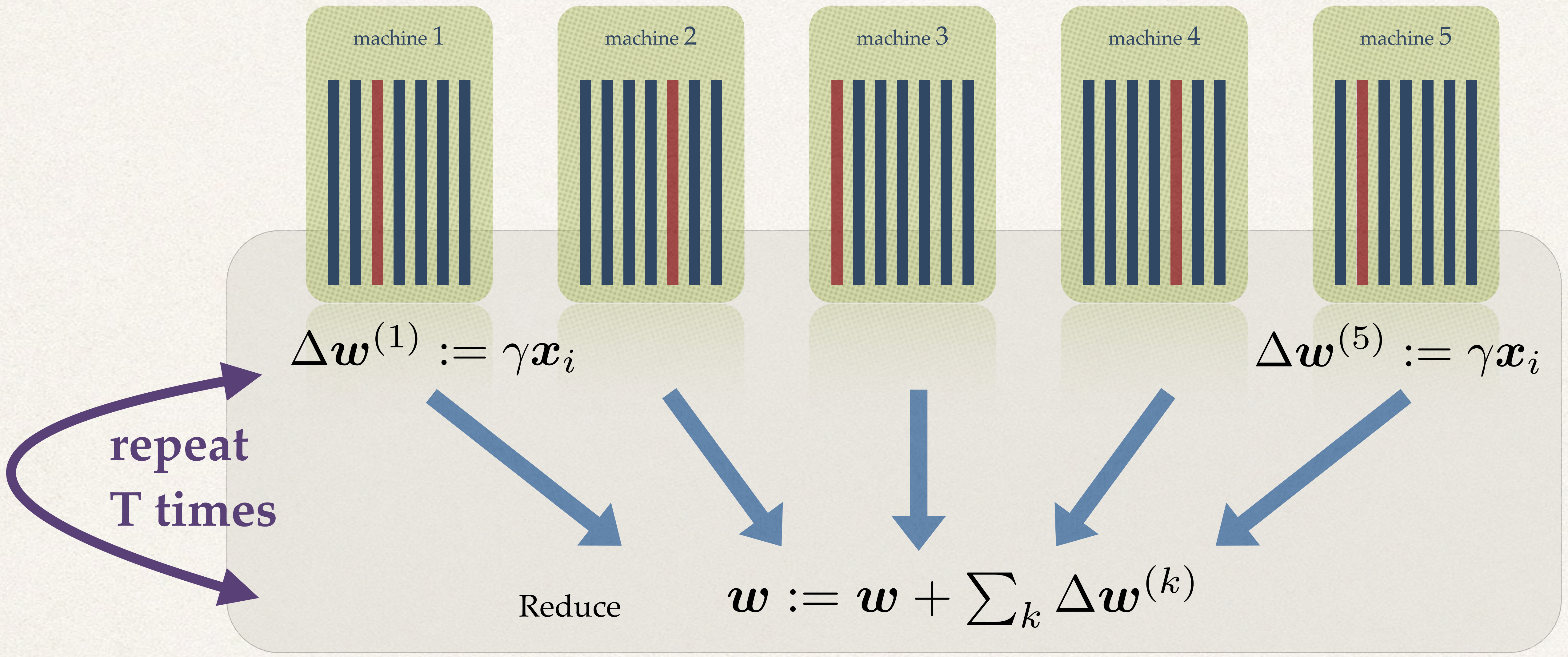
Distribute compute & memory across many devices



One-Shot Model Averaging Does Not Work



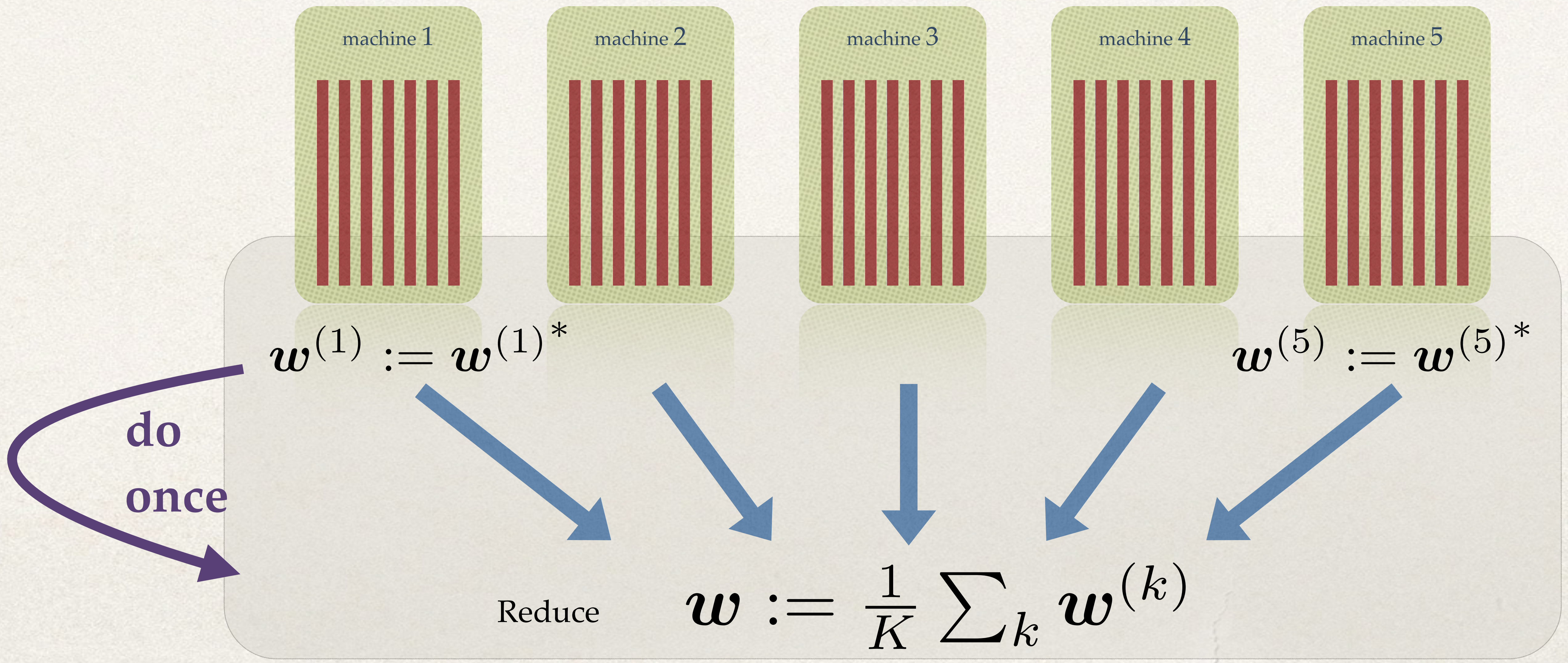
Communication: Always / Never



Naive Distributed SGD

 # local datapoints read: T
 # communications: T
convergence: ✓

"always communicate"

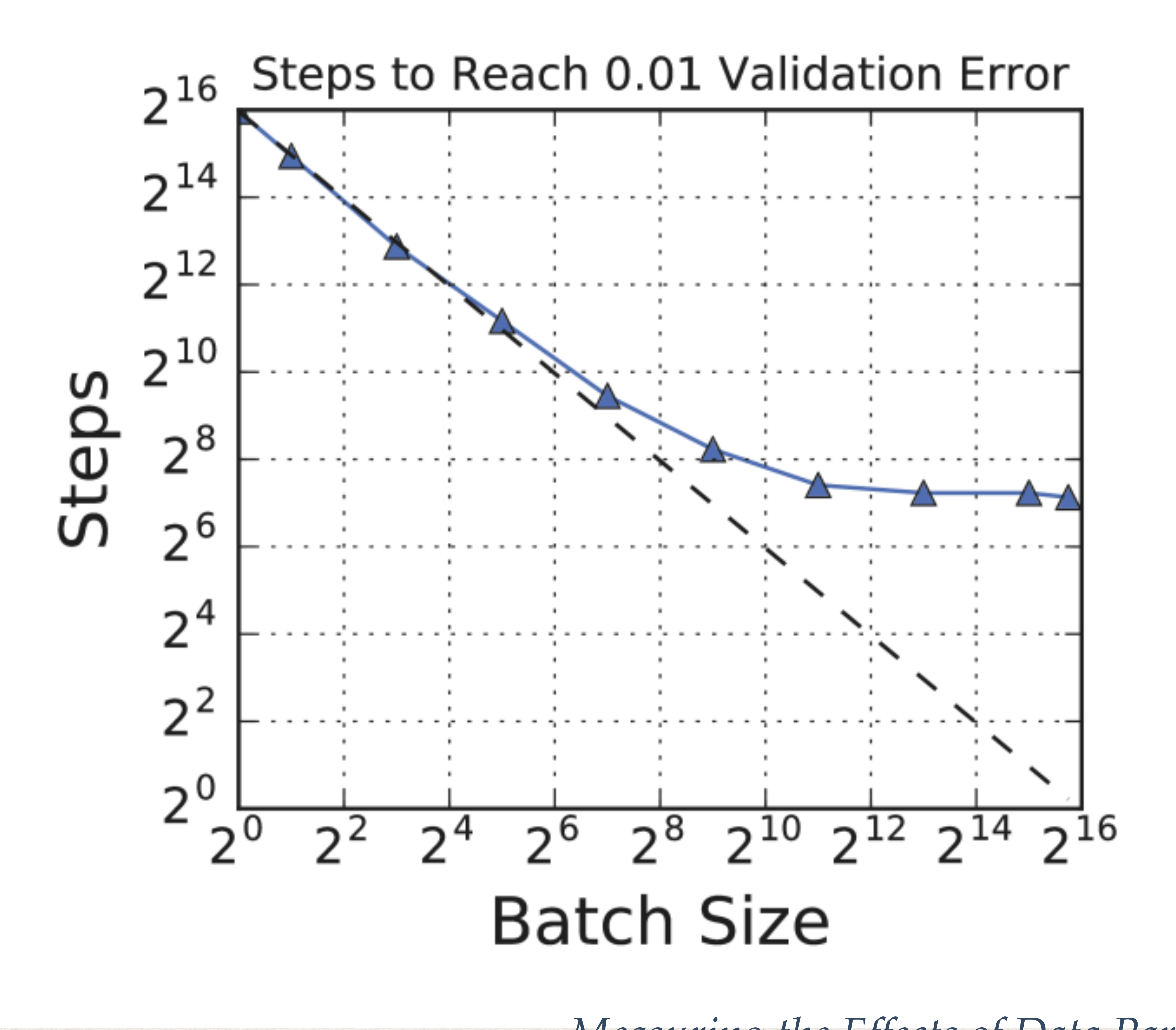


One-Shot Averaged Distributed Optimization

 # local datapoints read: T
 # communications: 1
convergence: ✗

"never communicate"

Just increase the batch size!



Measuring the Effects of Data Parallelism on Neural Network Training
Shallue et al., 2018

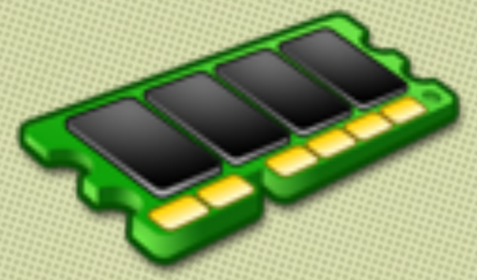
Challenge

The Cost of Communication

$\boldsymbol{v} \in \mathbb{R}^{100}$

- Reading $\boldsymbol{v}$ from memory (RAM)

  100 ns



- Sending $\boldsymbol{v}$ to another machine

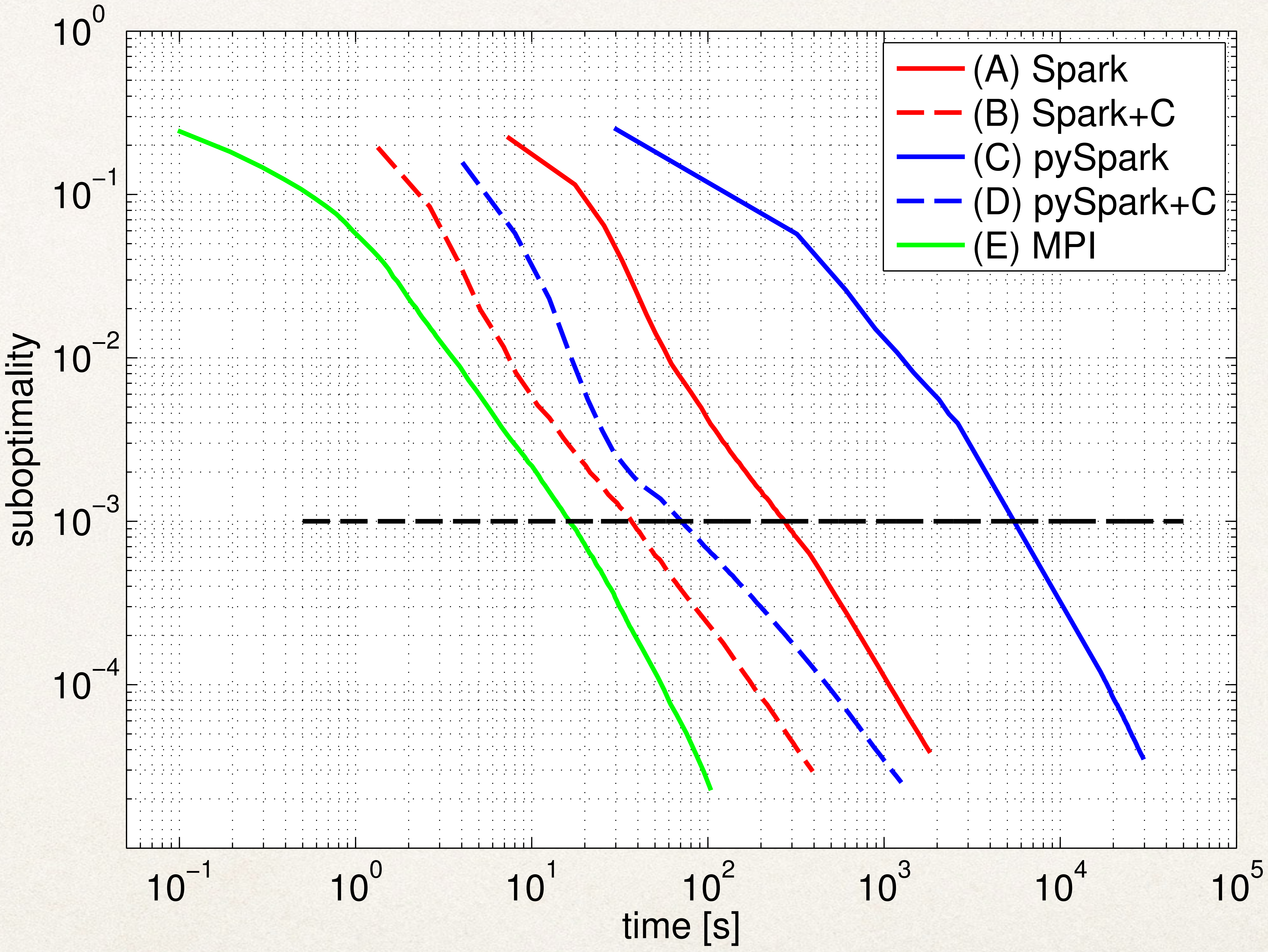
  500’000 ns

- Typical Map-Reduce iteration

  10’000’000’000 ns

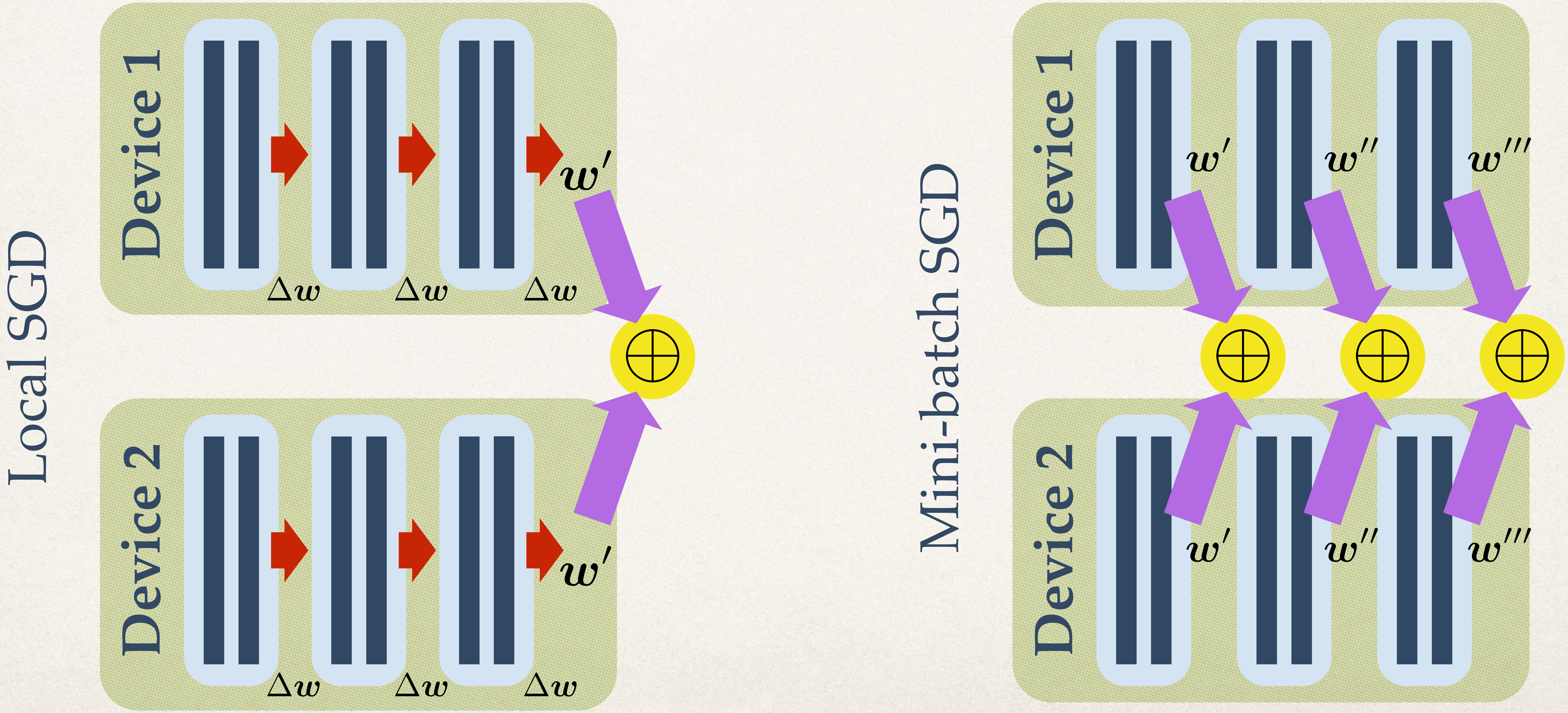
Challenge

The Cost of Communication



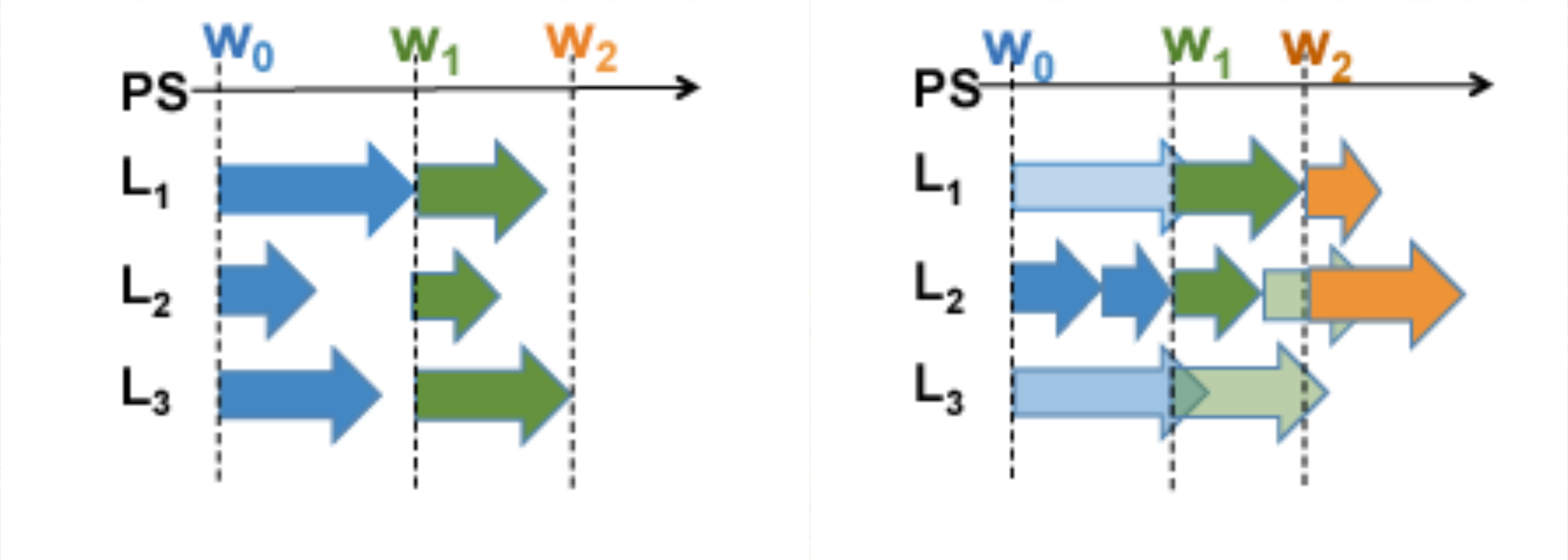
High-Performance Distributed Machine Learning using Apache Spark
Dünner et al. 2016, arxiv.org/abs/1612.01437

Data Parallel DL, Local Update Steps



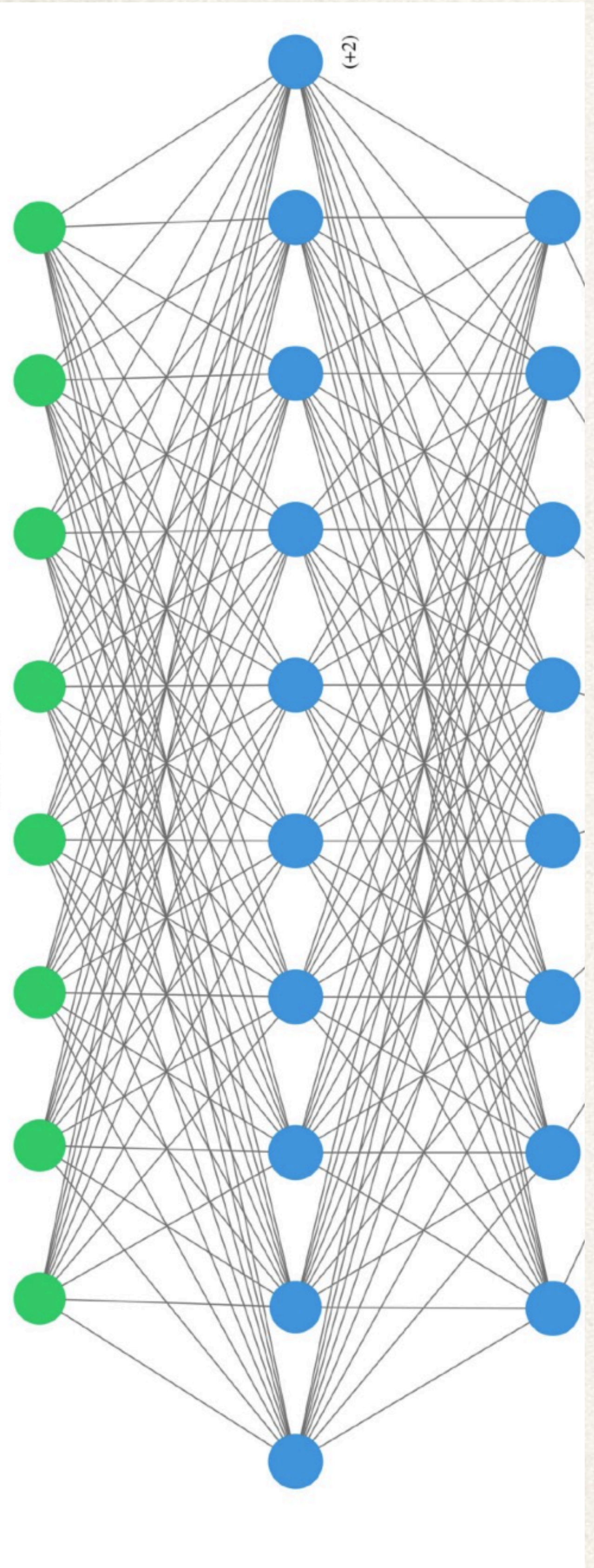
Asynchronous Parallel SGD

- **Synchronous**
- **Asynchronous**



Mini-Batch!

Communication Compression

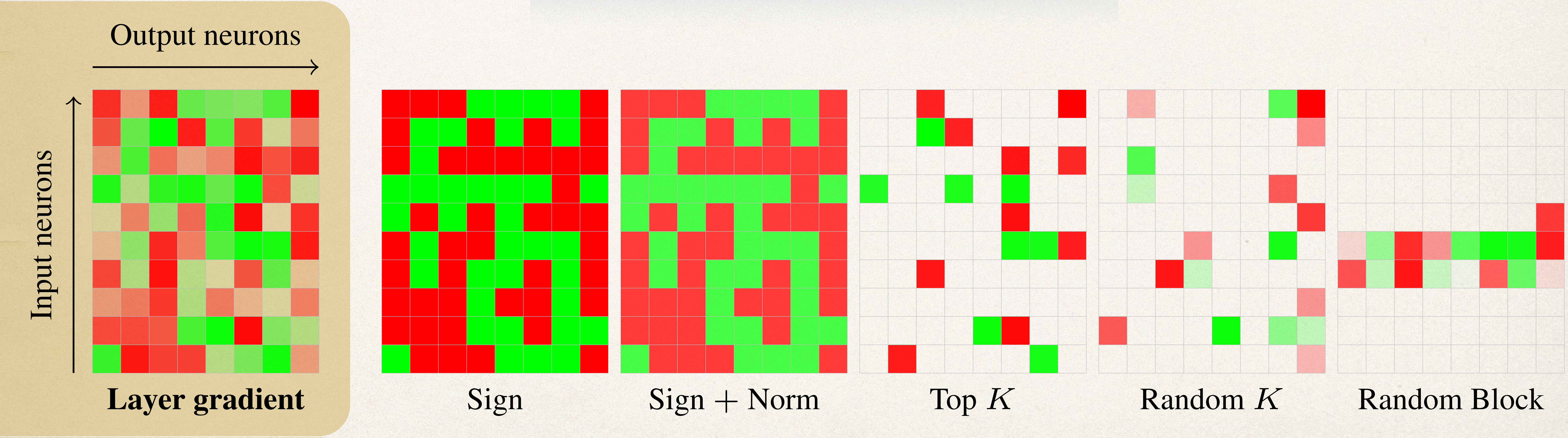


A compressed version of model updates?

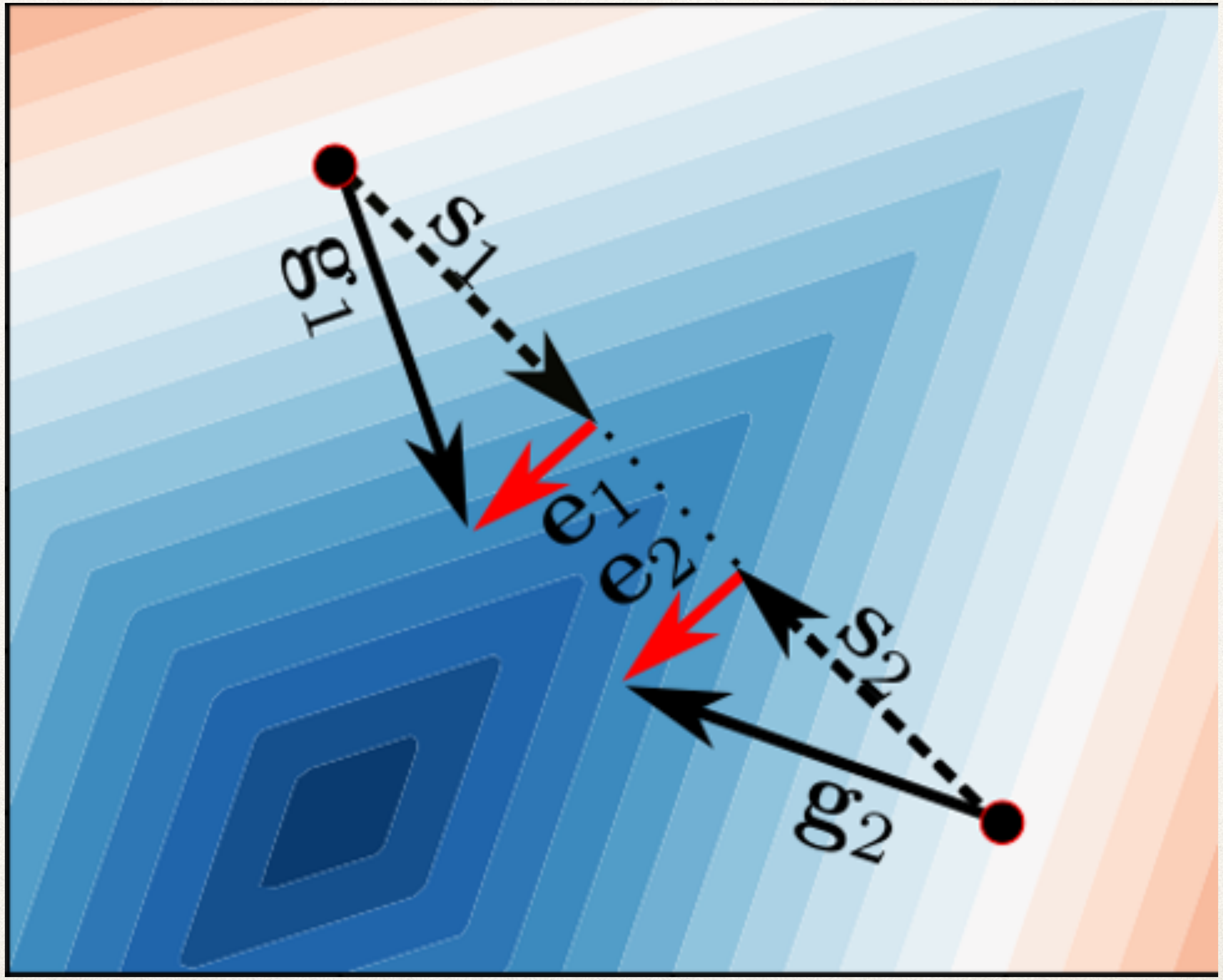
Examples:	**Communication Reduction**
✤ quantization (e.g. 1-bit SGD)	**32x**
✤ top k=1% of all the entries	**100x**
✤ rank-1 approximation	**>100x**

Gradient Compression

A compressed version of model updates?

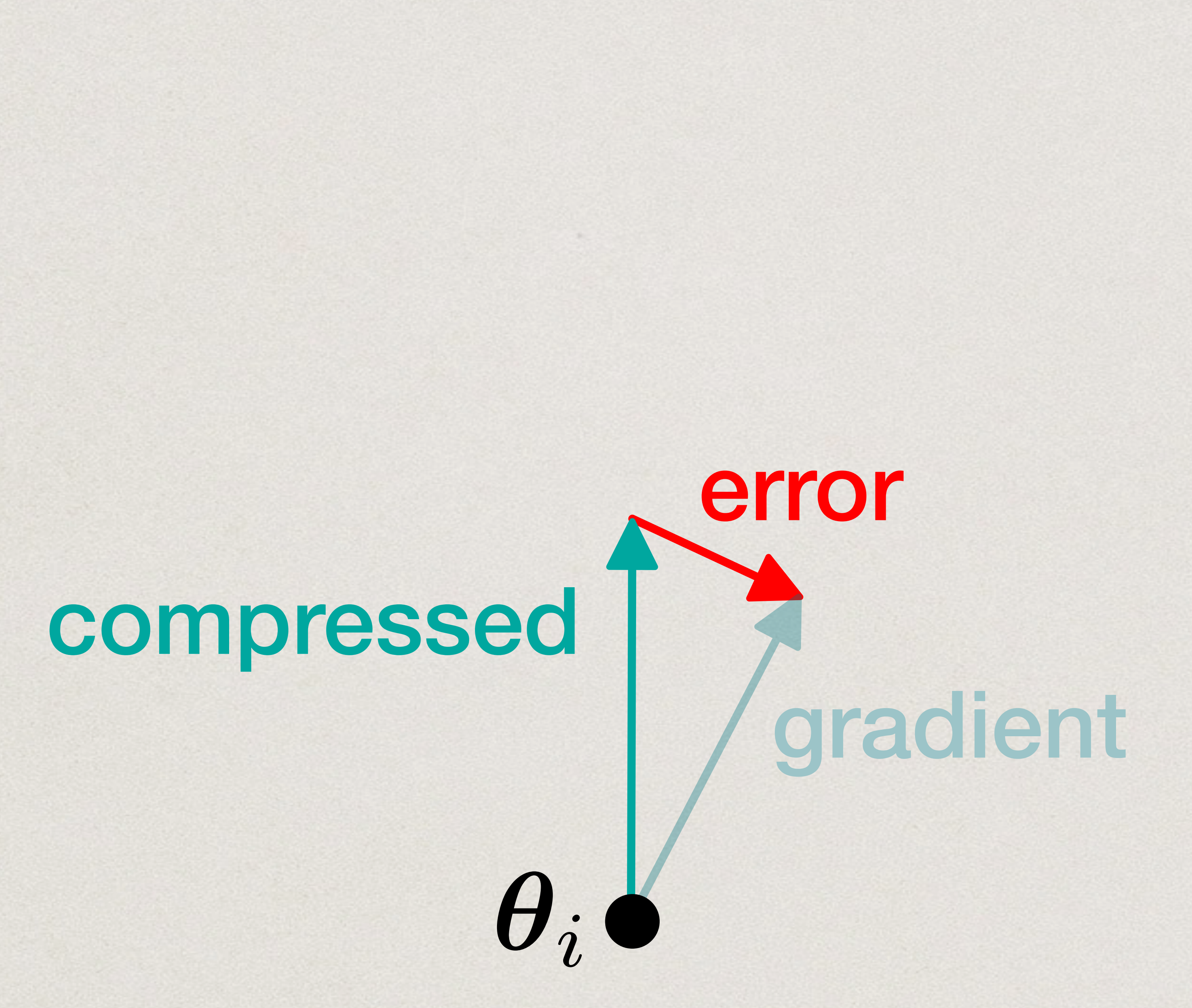


SGD fails with naive/biased compressors

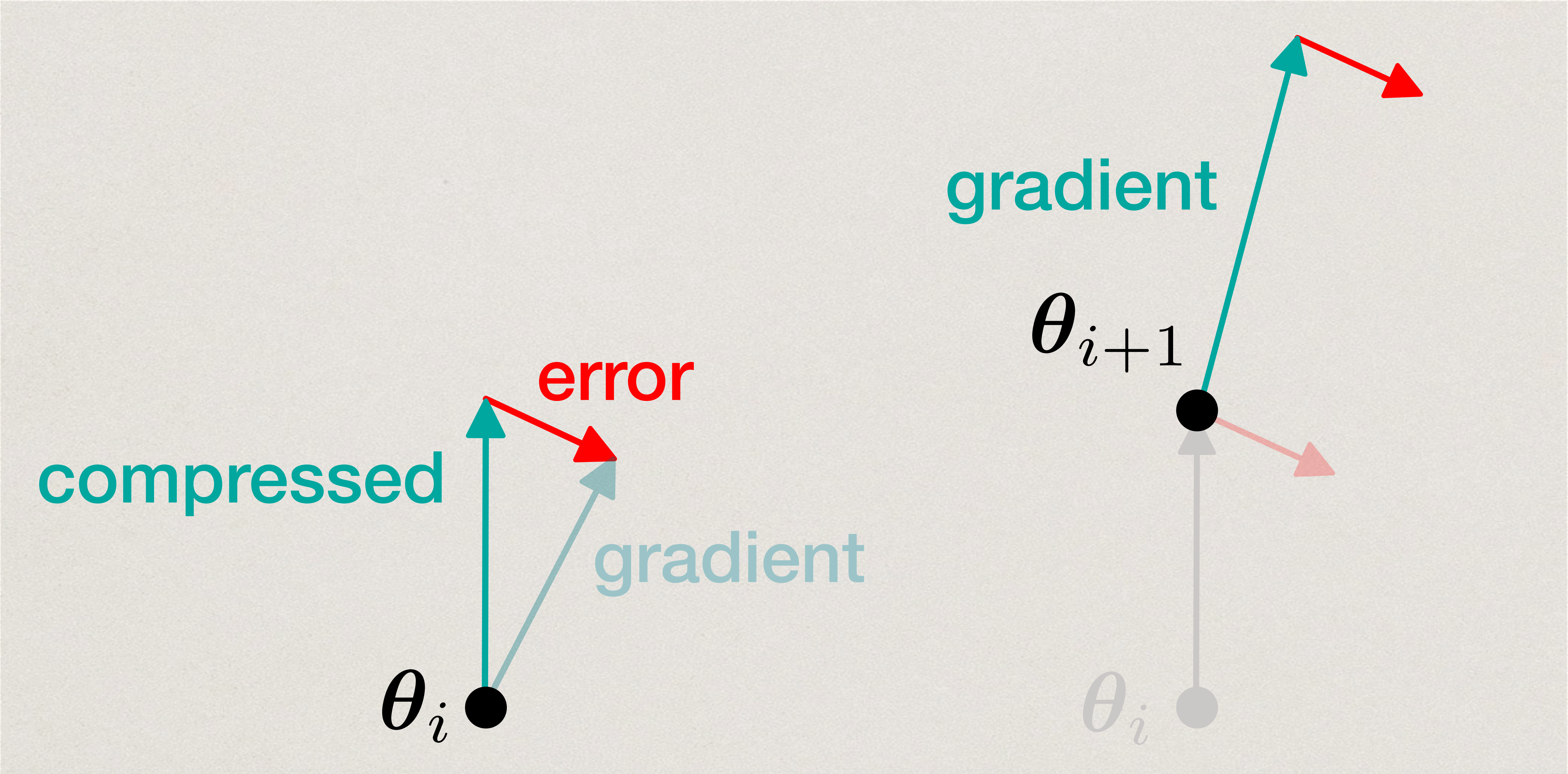


$$\min_{x \in \mathbb{R}^2} \; |x_1 + x_2| + 2|x_1 - x_2|$$

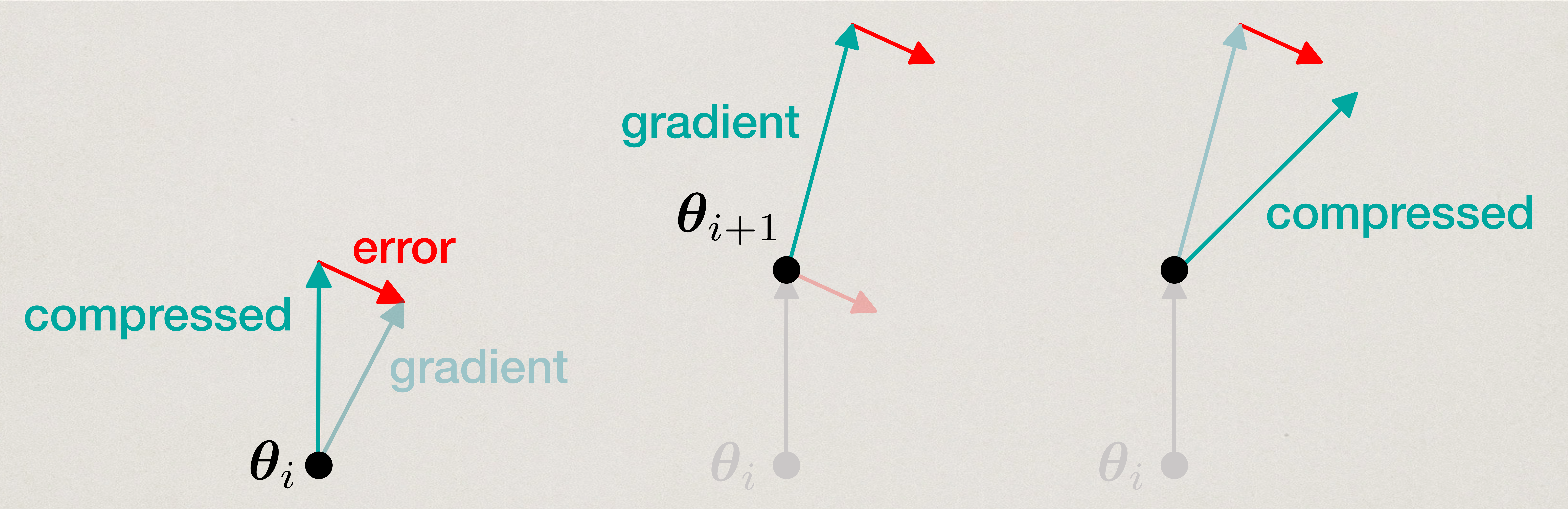
Error Feedback



Error Feedback



Error Feedback



Error Feedback: Convergence Rate

δ: compression ratio

$$\|\mathcal{C}(\mathbf{x}) - \mathbf{x}\|_2^2 \leq (1-\delta)\|\mathbf{x}\|_2^2$$

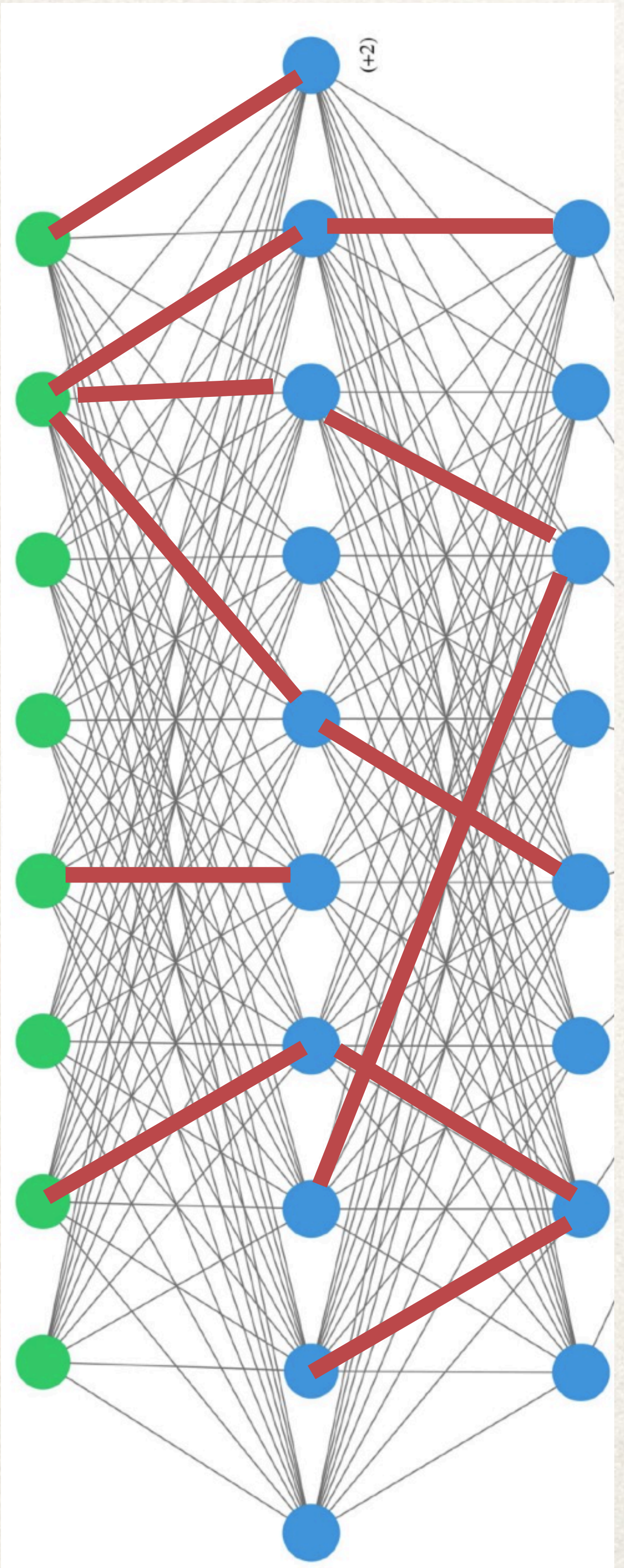
SGD on smooth non-convex objectives (w/ central coordinator)

$$\mathbb{E}\|\nabla f(\bar{x}_t)\|^2 \leq \mathcal{O}\left(\frac{1}{\sqrt{nT}} + \frac{1}{\delta^2 T}\right)$$

Can we also save Compute and Memory?

e.g. for deployment on low-resource devices

Model Compression with Error Feedback



Prune most weights (set to zero)

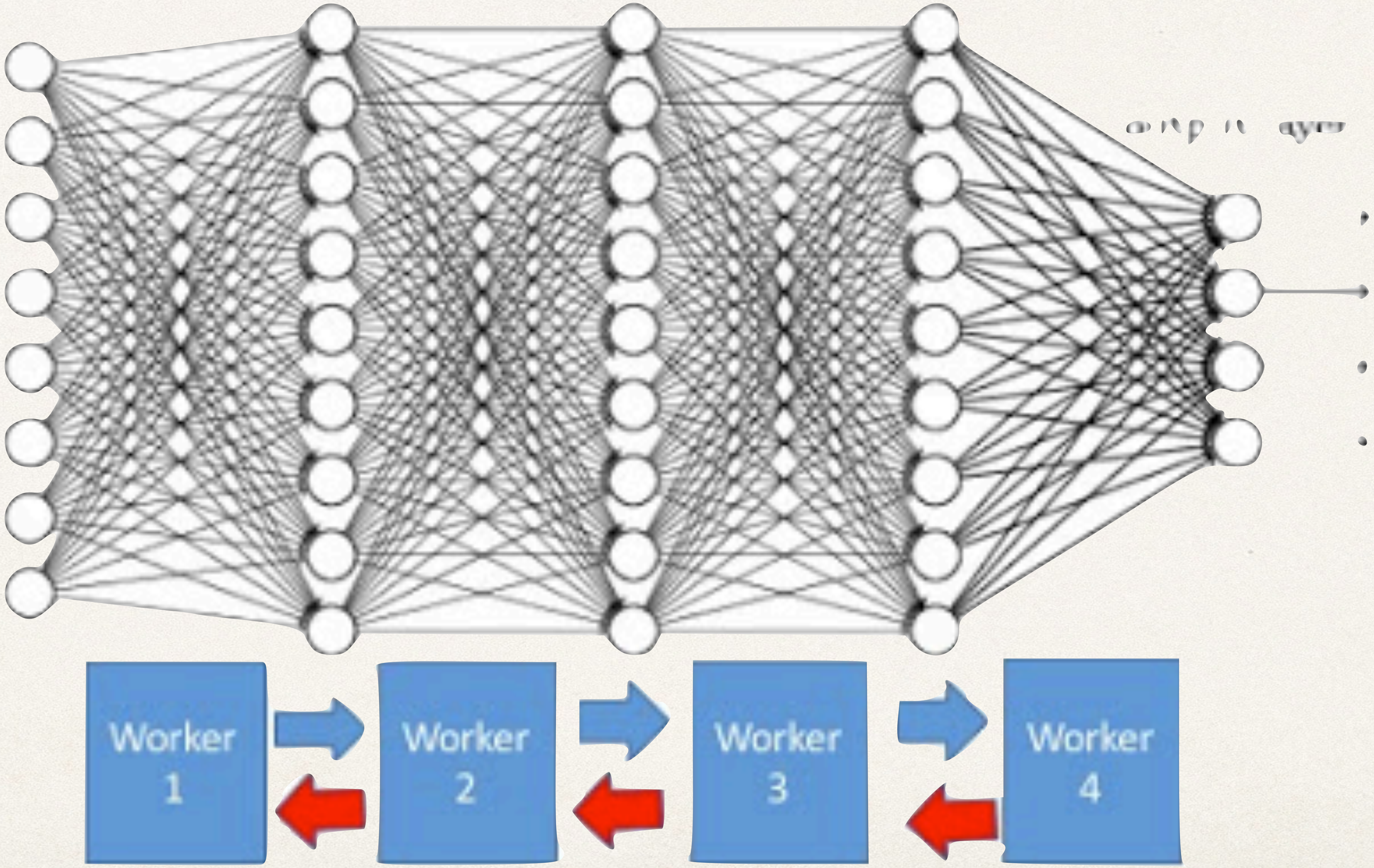
set to limited precision

interactive while training

Dynamic Model Pruning with Feedback

(Model Parallel)

Model-Parallel DL



http://kuozhangub.blogspot.com/2017/08/data-parallel-and-model-parallel.html

2

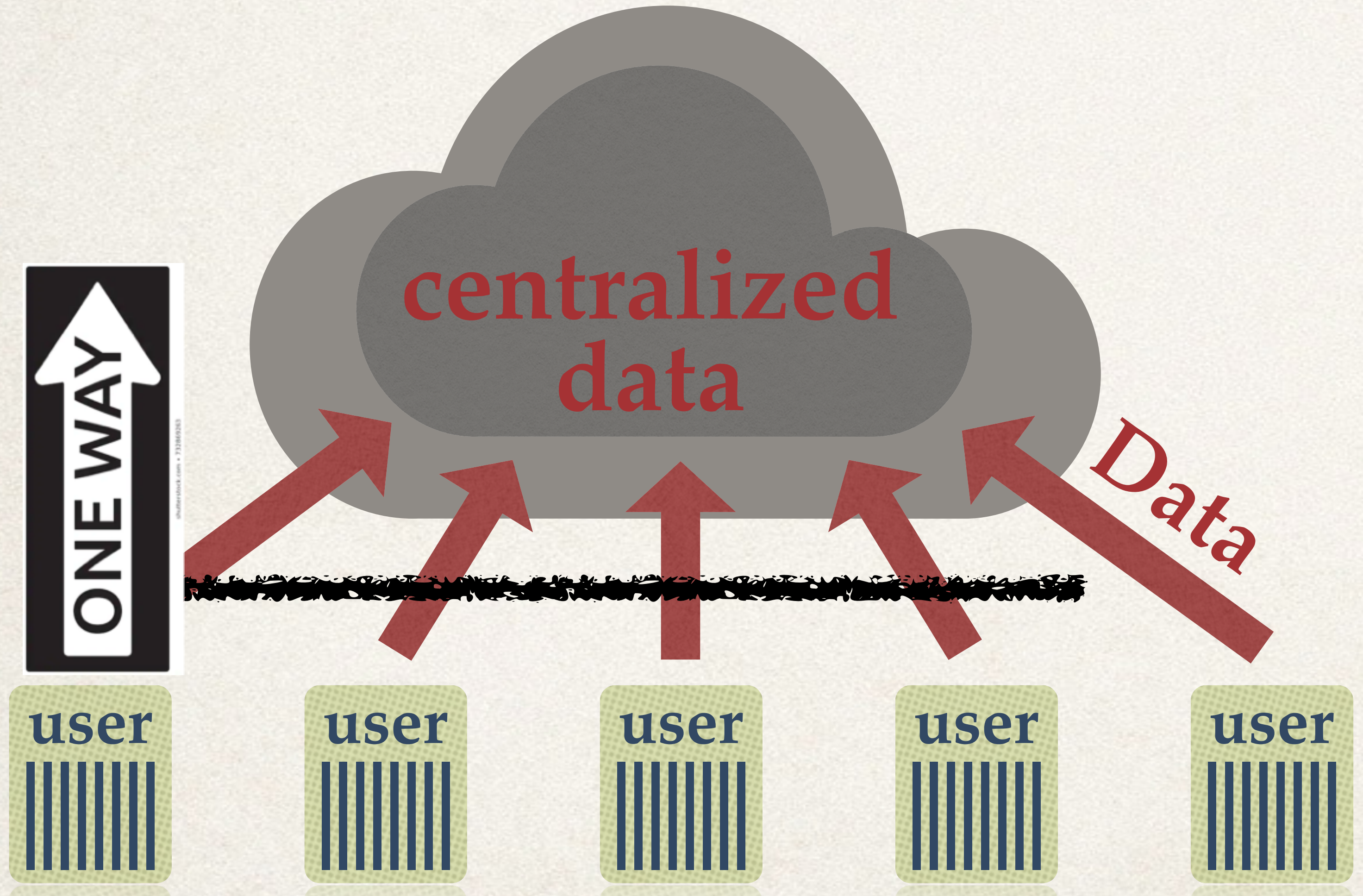
Gradients from collaborators:
- Federated Learning

Collaborative & Federated Learning

Distributed, Collaborative Learning

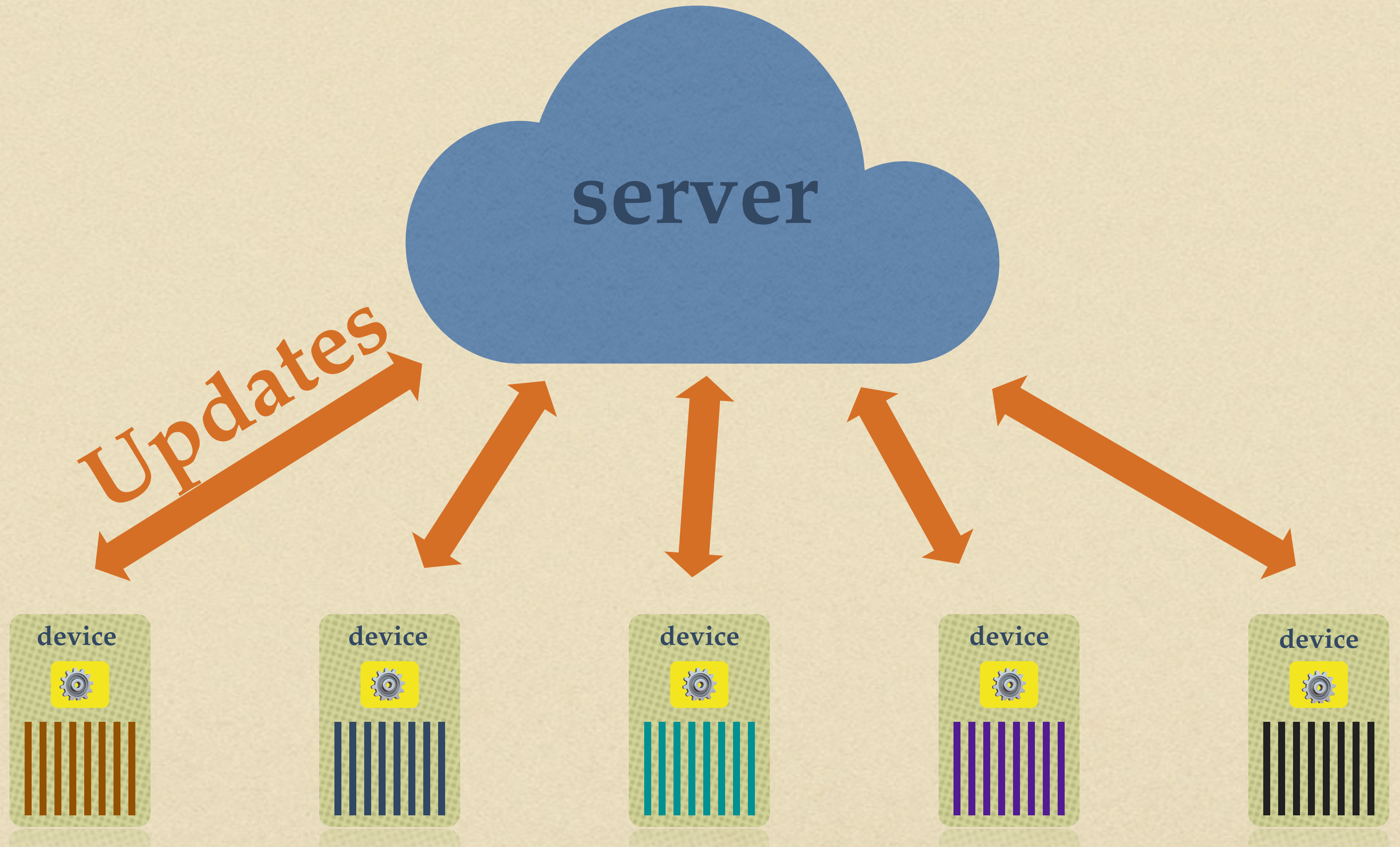
centralized

traditional, sharing data



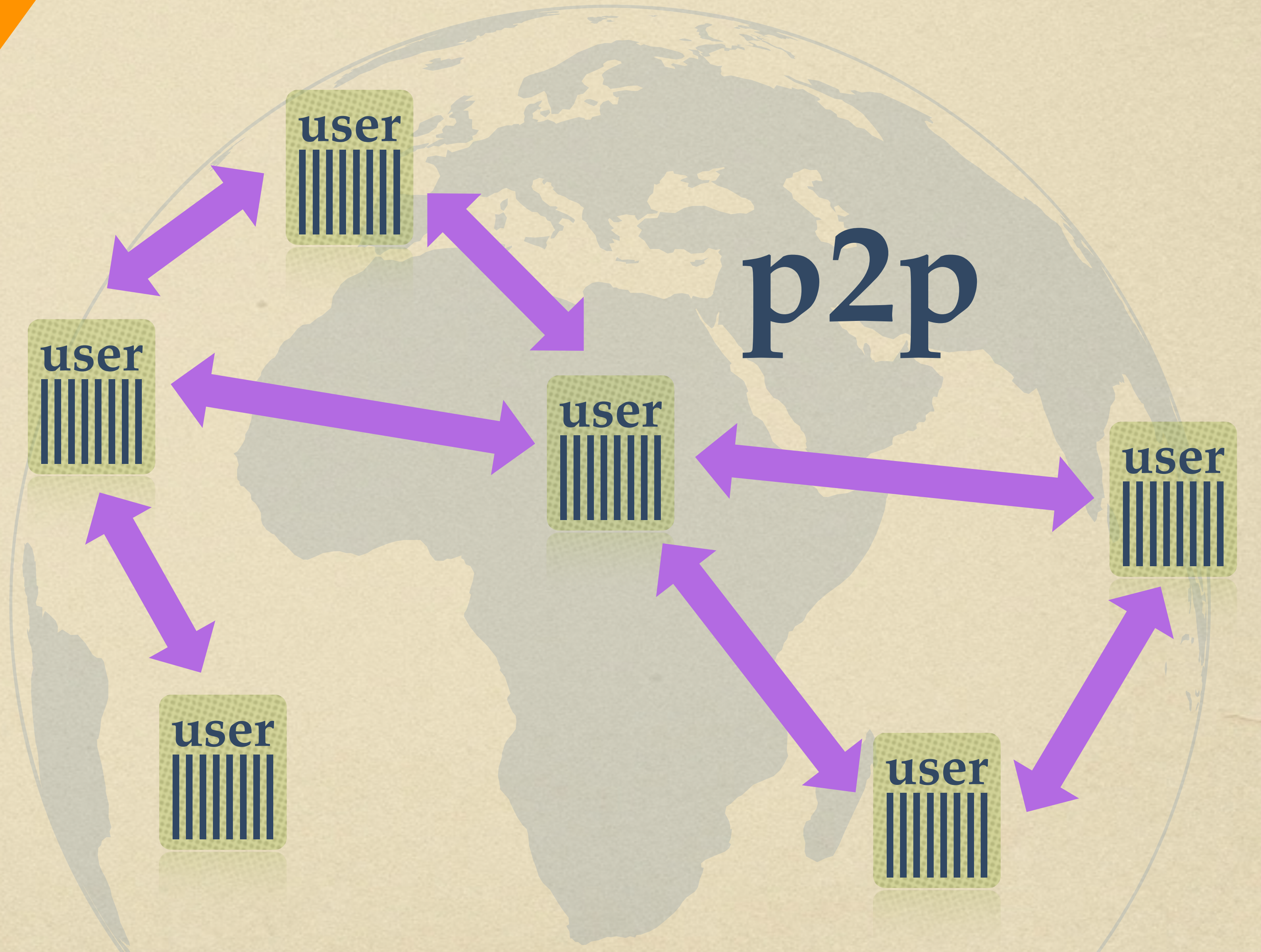
federated

sharing model updates



decentralized

learning



Thanks!

mlo.epfl.ch
tml.epfl.ch