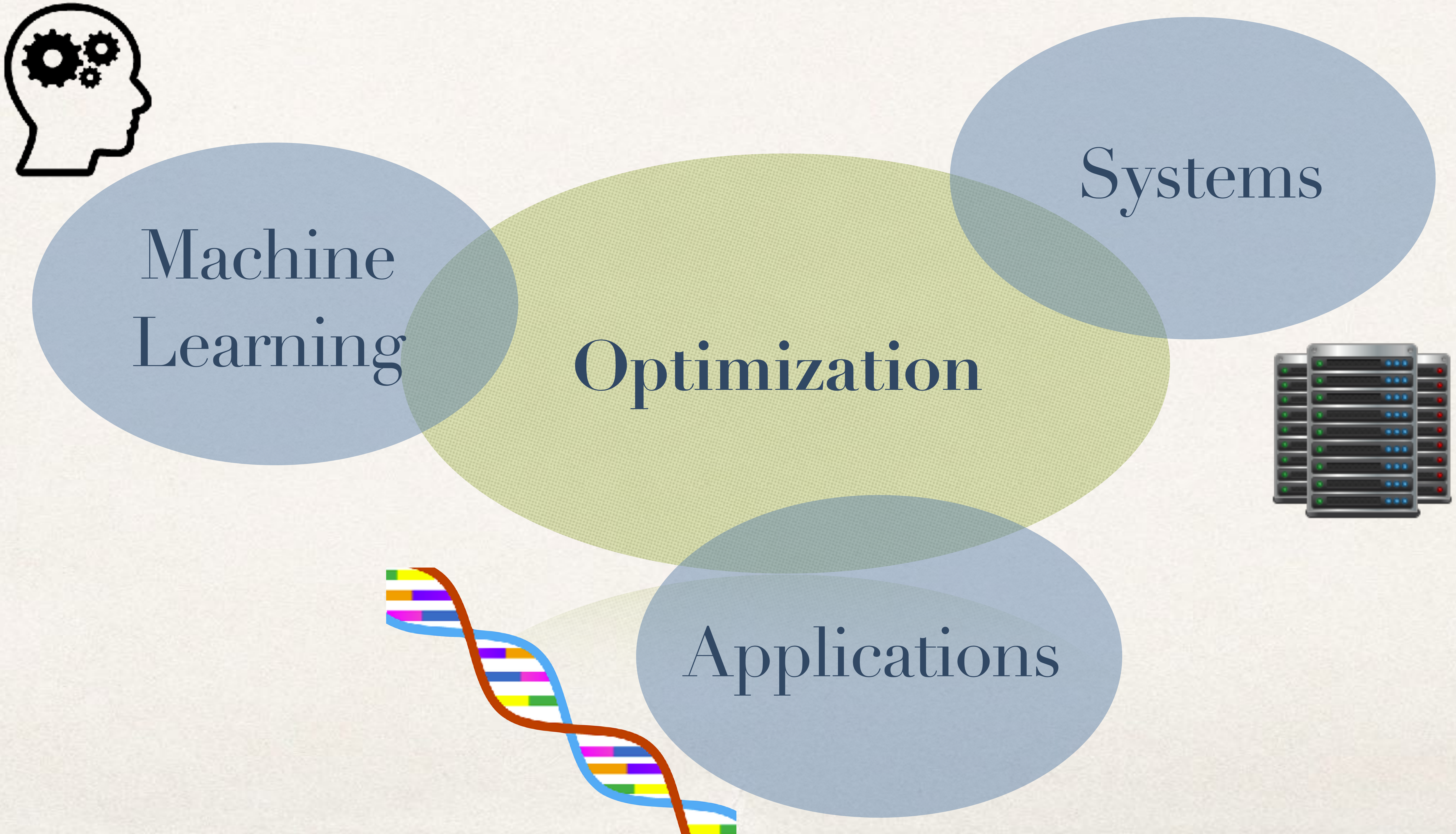
Optimization

for Machine Learning in Practice

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

Where are we?



Trends - General

- **privacy in ML**
- **decentralized training**
- **new hardware & systems**
- **trust in ML** (e.g. robust & secure against adversarial attacks)

Adversarial Attacks

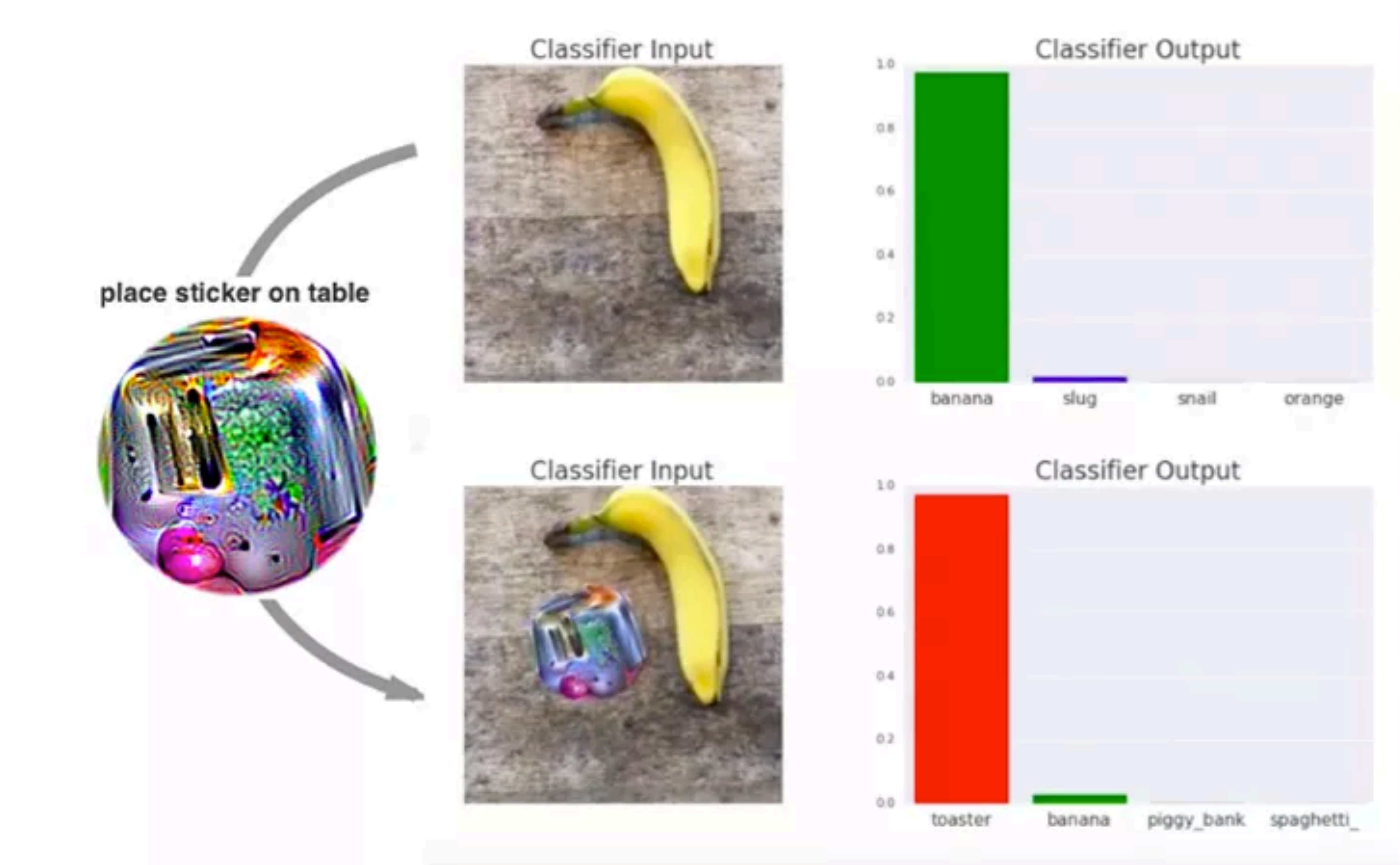


Image: [Tom B. Brown/Dandelion Mané]

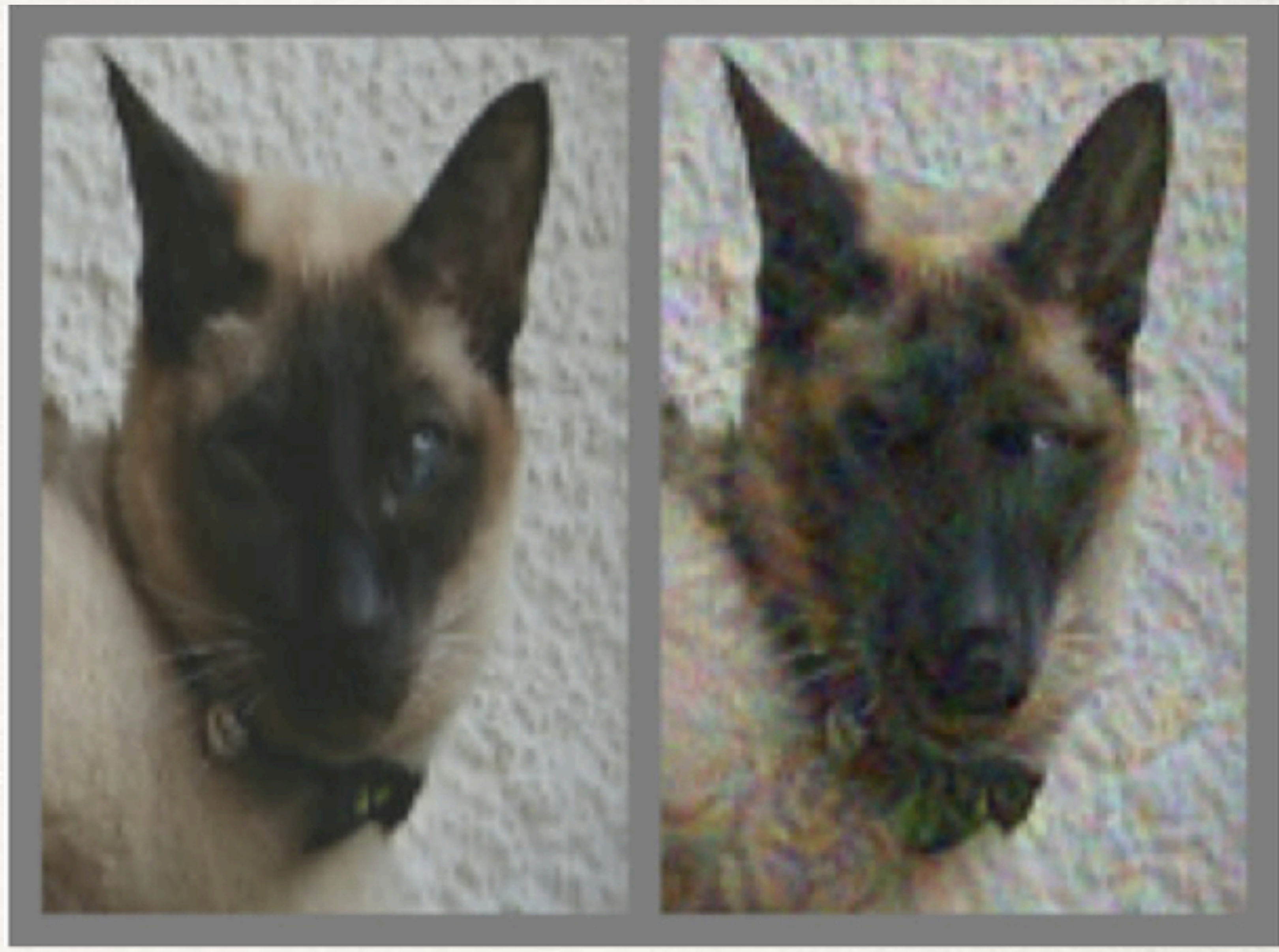


Image: Elsayed ,Papernot et al 2018

Adversarial Attacks

- **training**

$$\min_{\boldsymbol{w}} \left(f_{\boldsymbol{w}}(\boldsymbol{x}_i) - y_i\right)^2$$

$\nabla_{\boldsymbol{w}} f$

change model

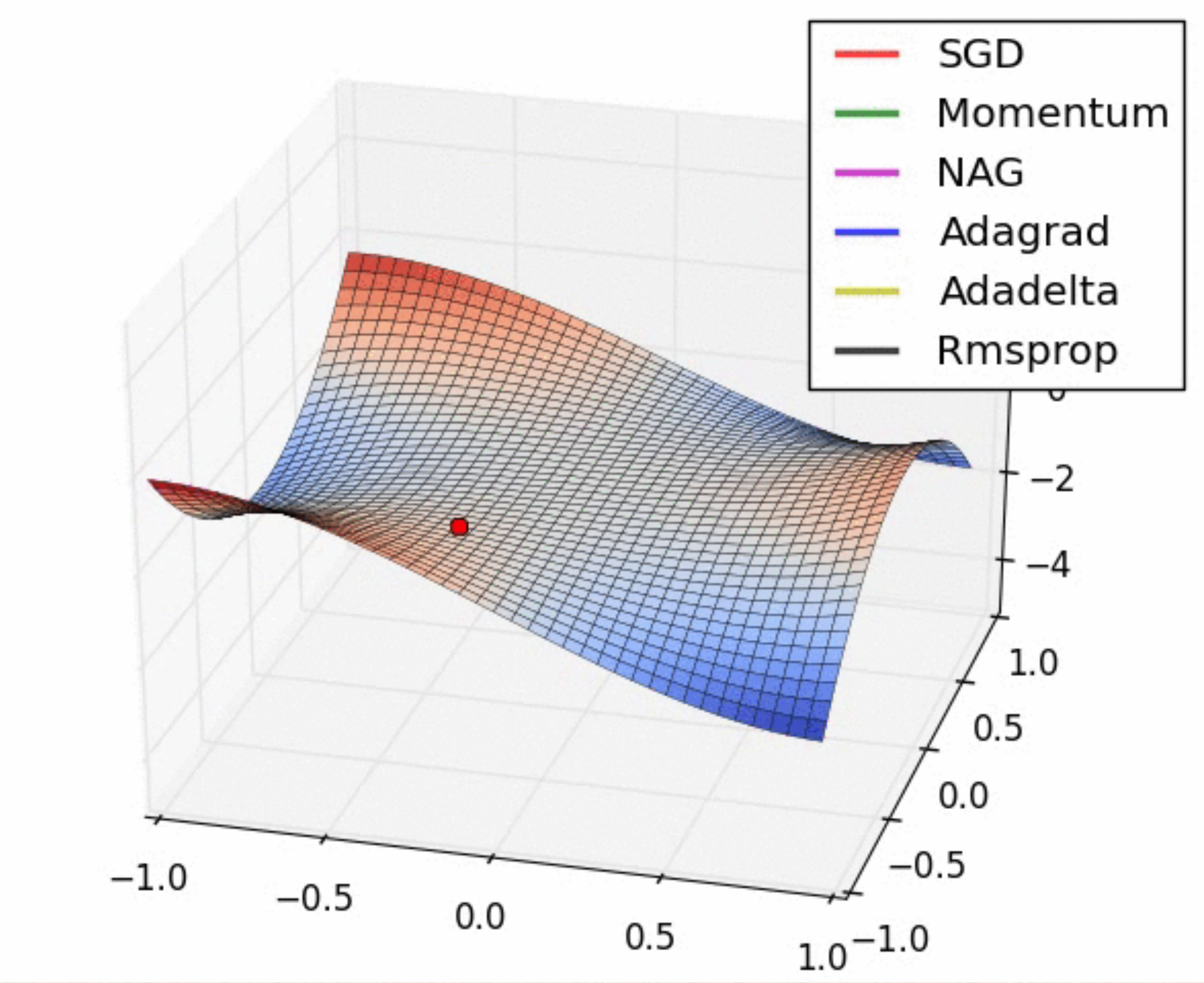
- **attacking**

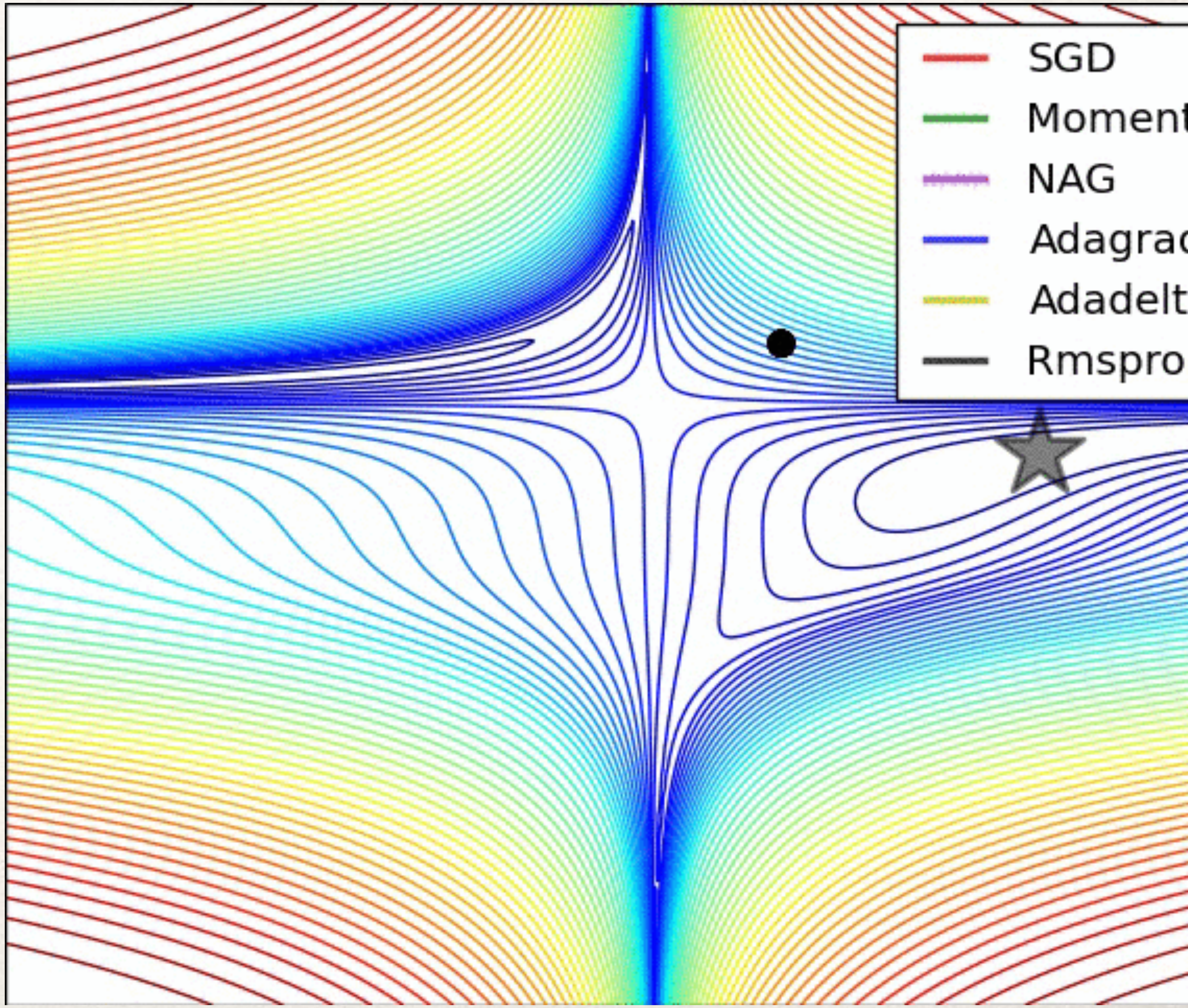
$$\min_{\boldsymbol{x}_i} \left(f_{\boldsymbol{w}}(\boldsymbol{x}_i) - y_i\right)^2$$

$\nabla_{\boldsymbol{x}_i} f$

change data

Practical comparison of algorithms

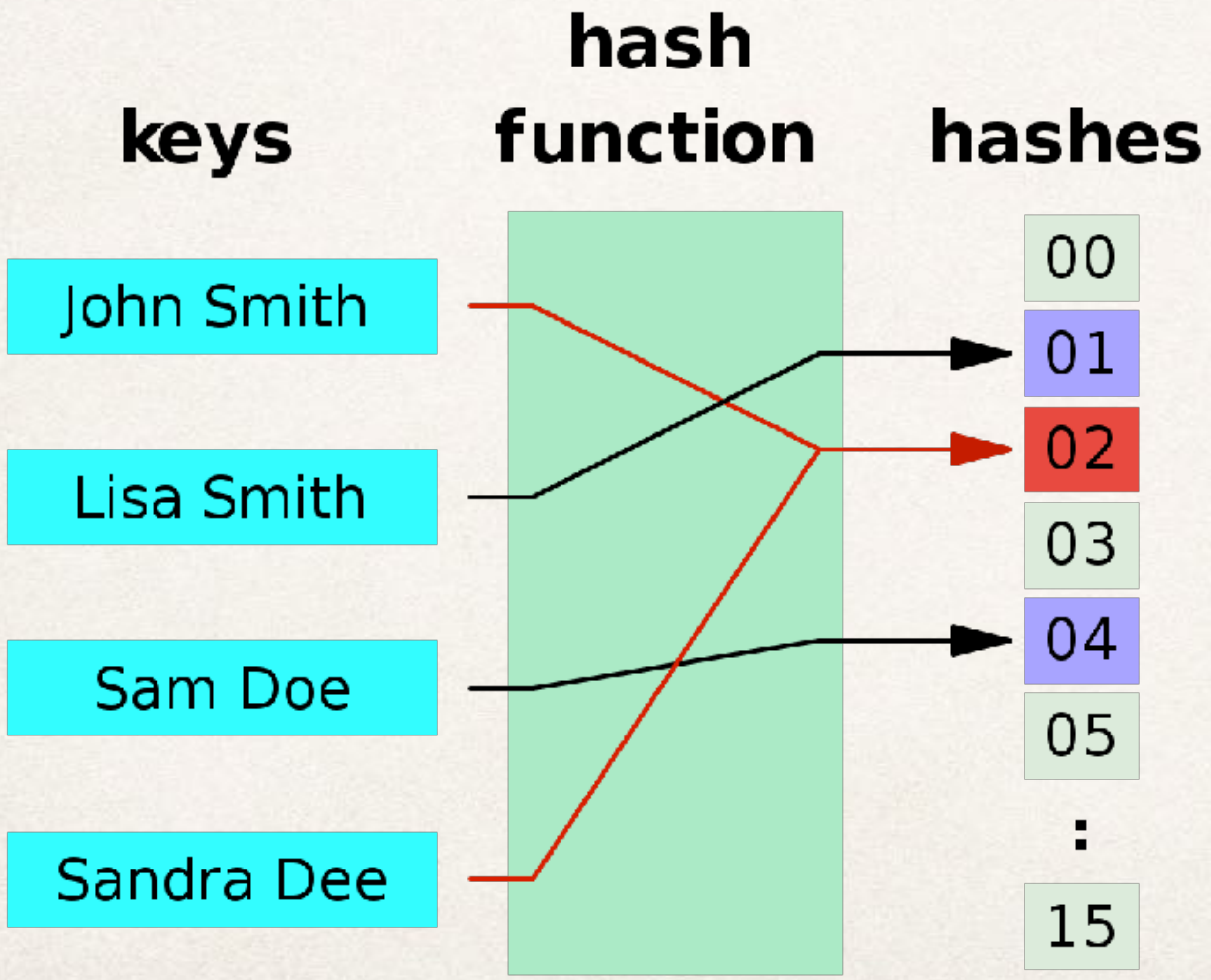




https://imgur.com/a/Hqolp#2dKCQHh

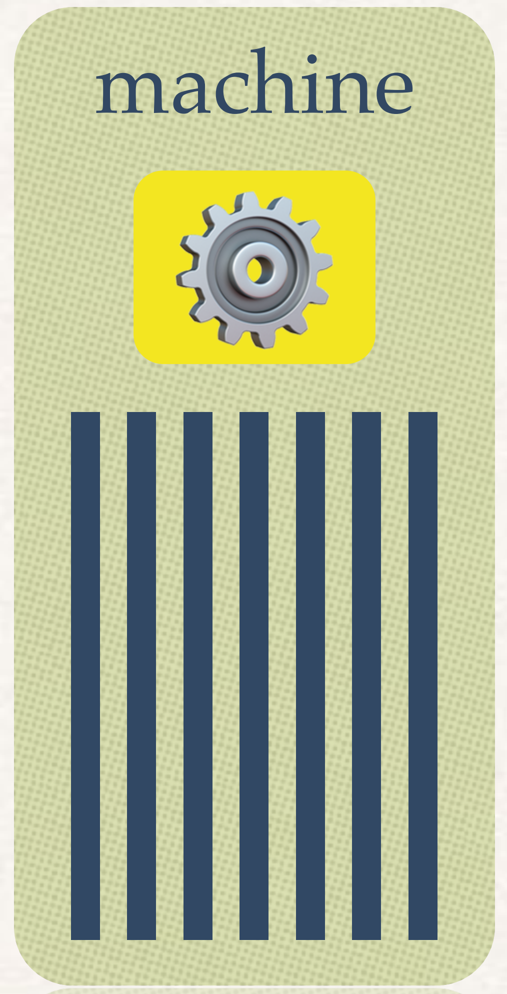
Practical tricks

- **feature hashing**

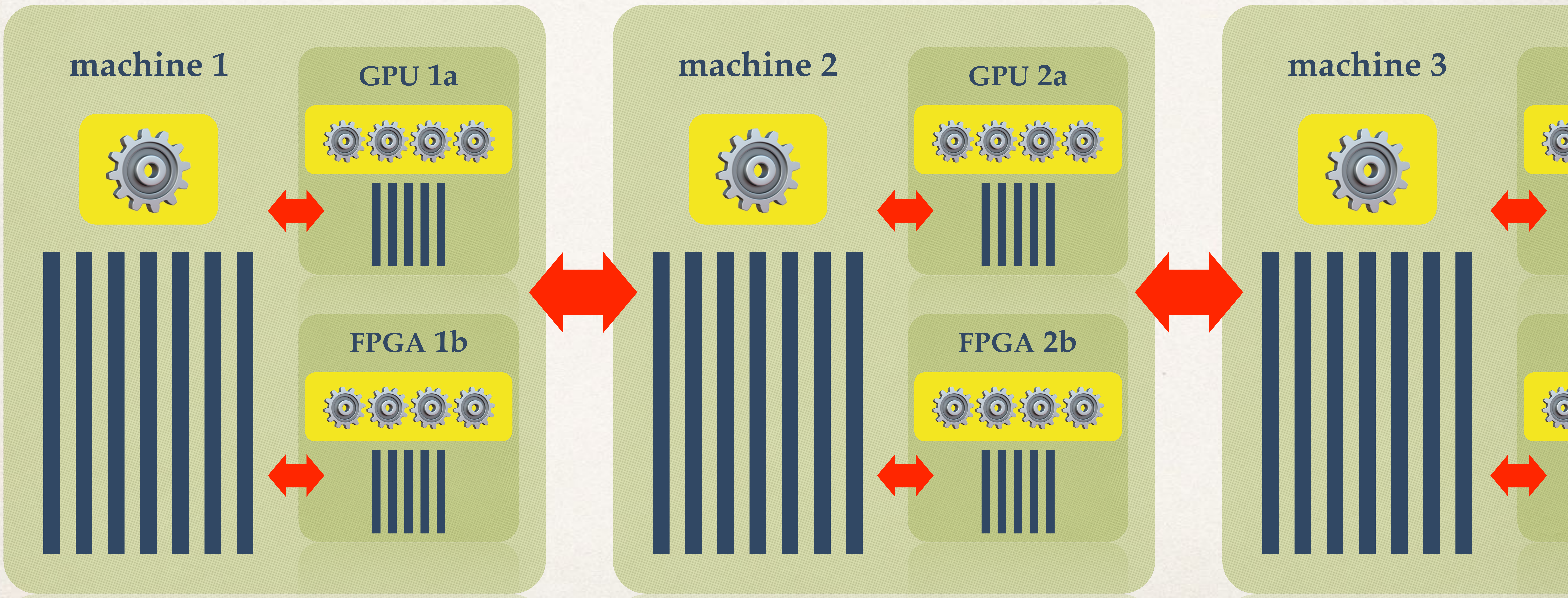


- **limited precision operations**

Systems ...then

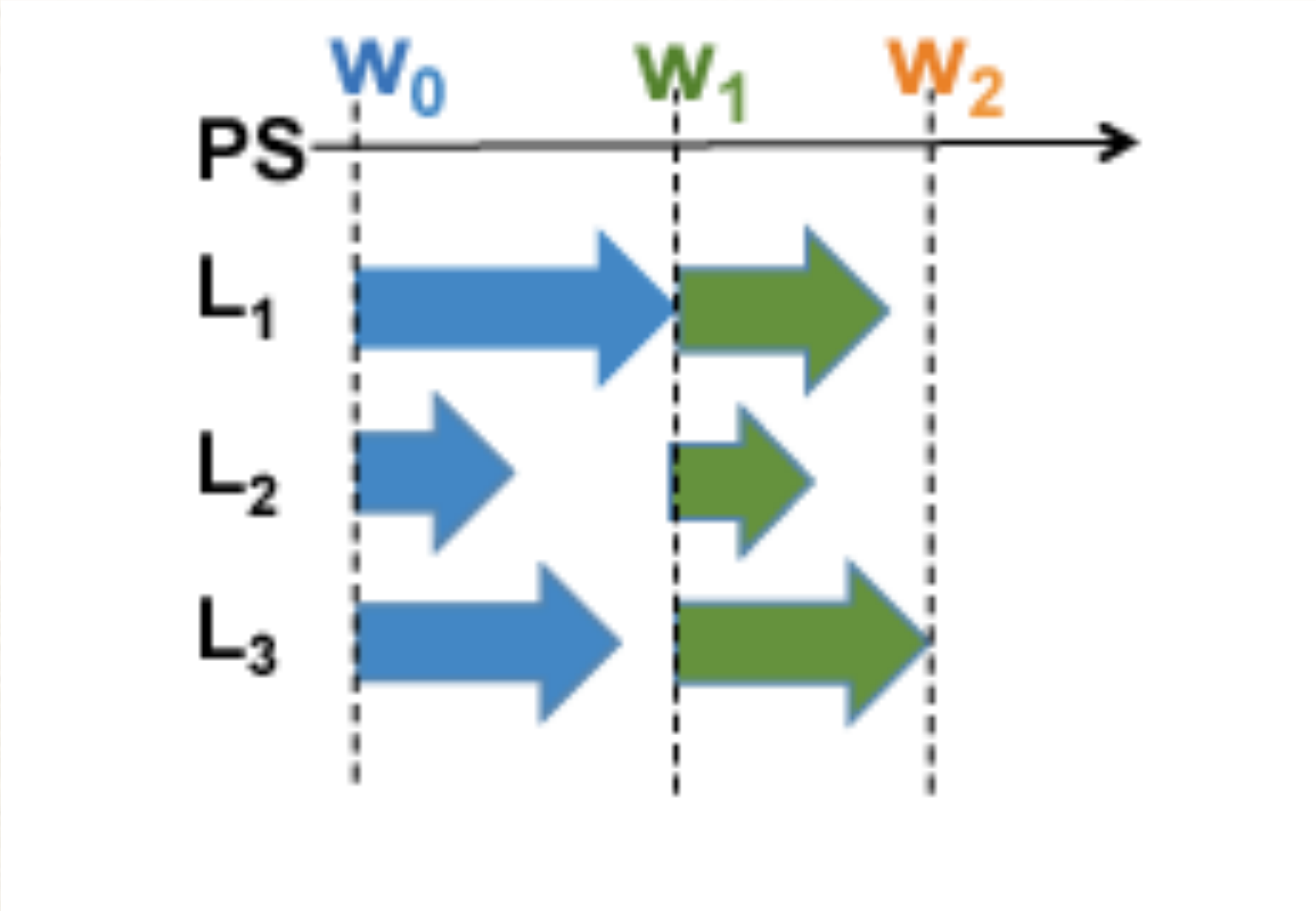


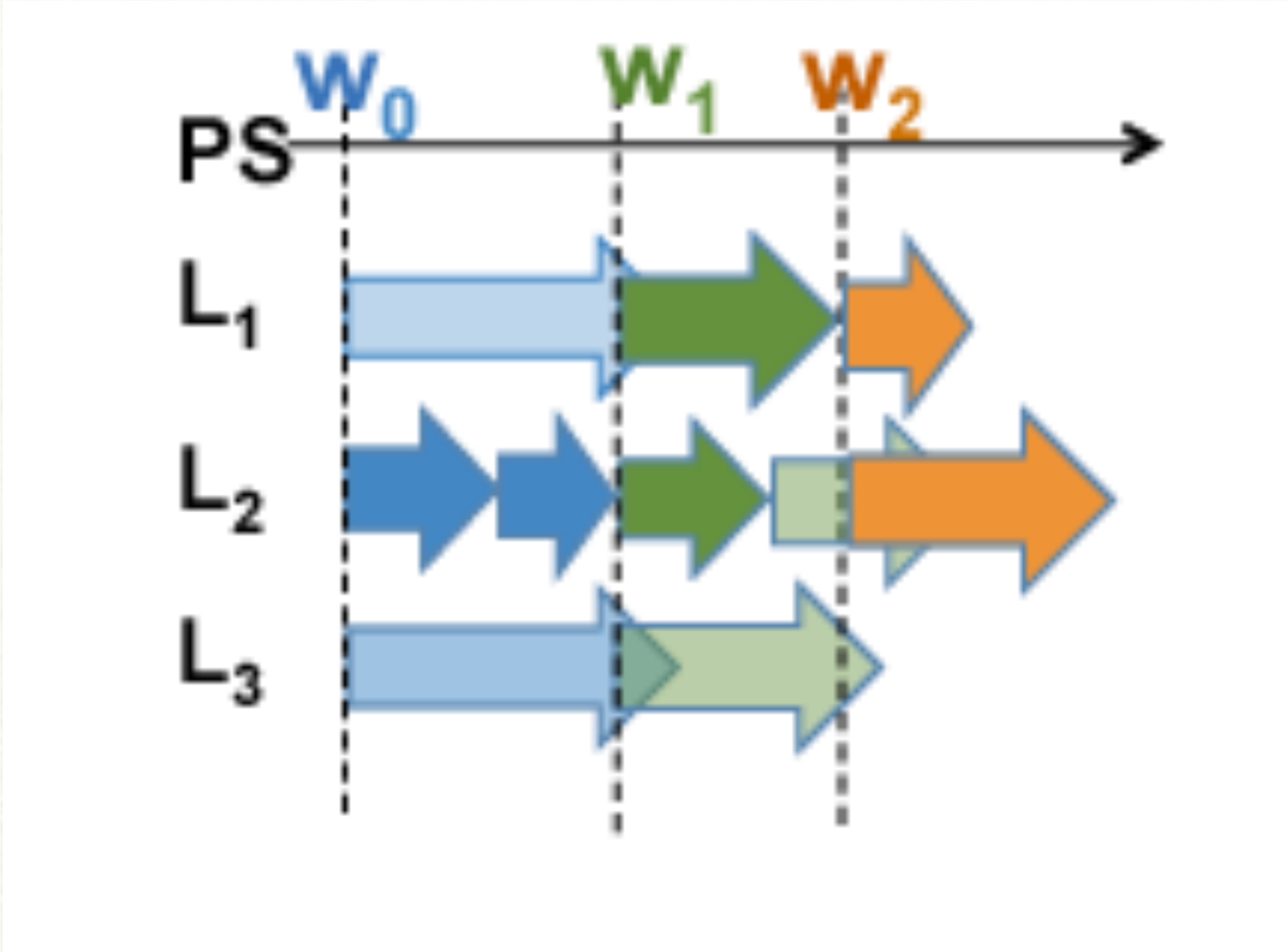
Systems …now



Parallel SGD

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



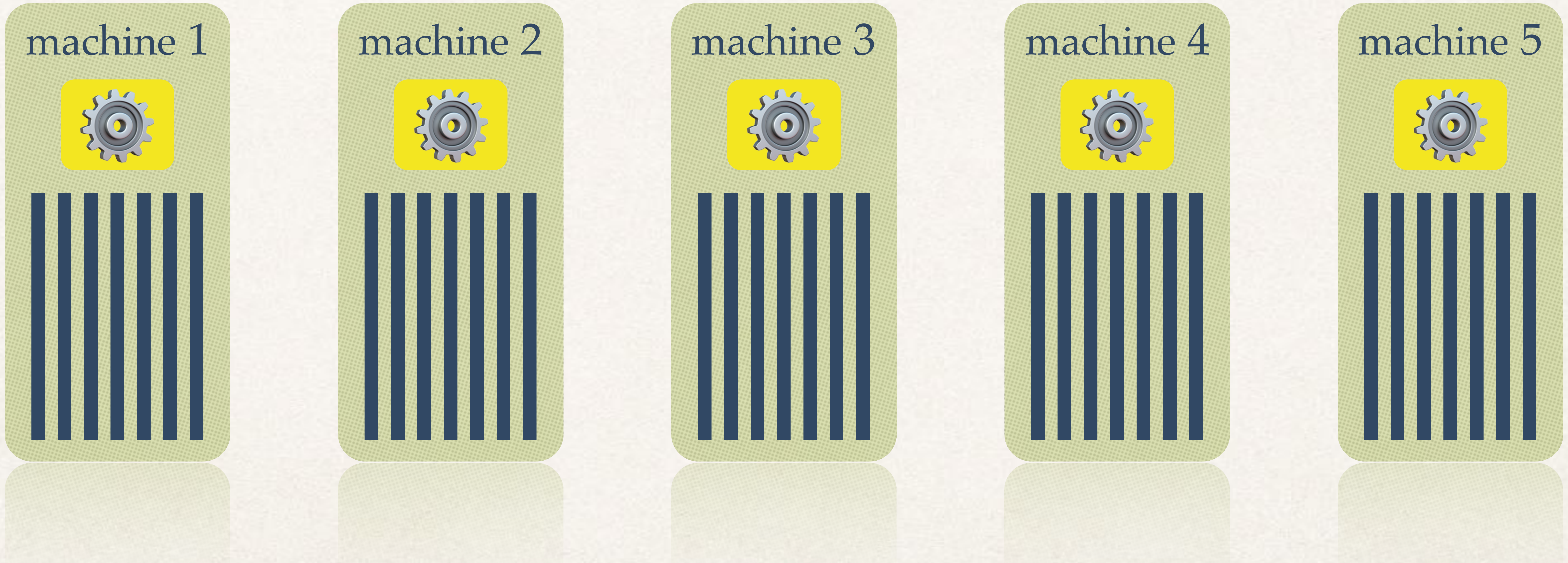


Mini-Batch!

1

Distributed

What if the data does not fit onto one device anymore?



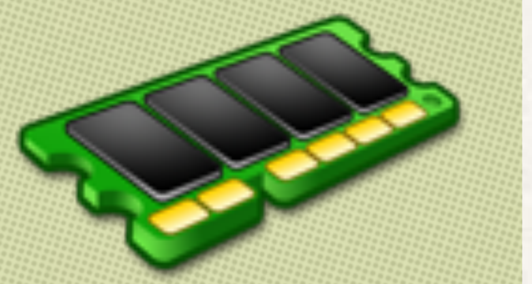
Challenge

The Cost of Communication

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

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

  100 ns



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

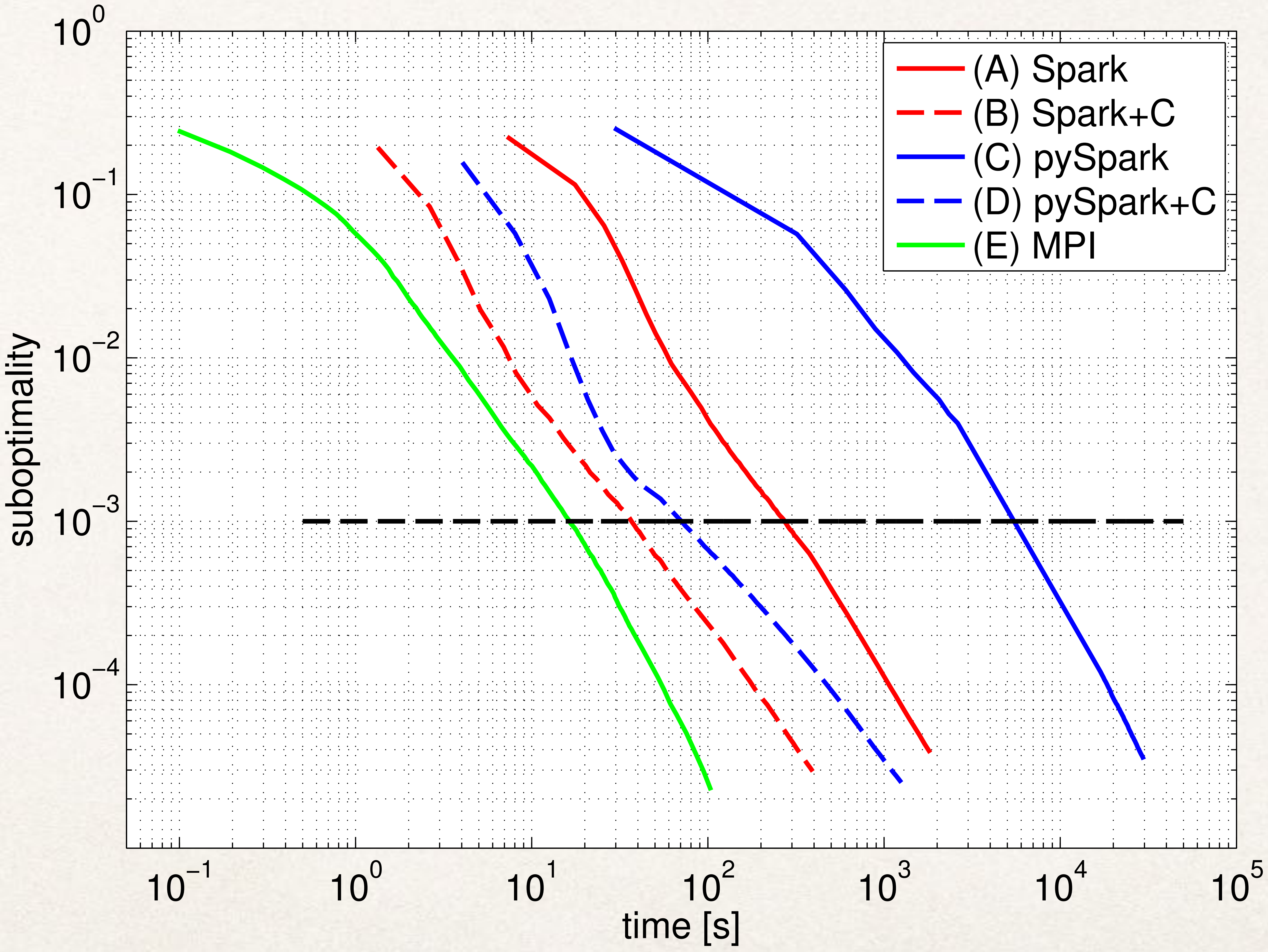
  500’000 ns

- Typical Map-Reduce iteration

  10’000’000’000 ns

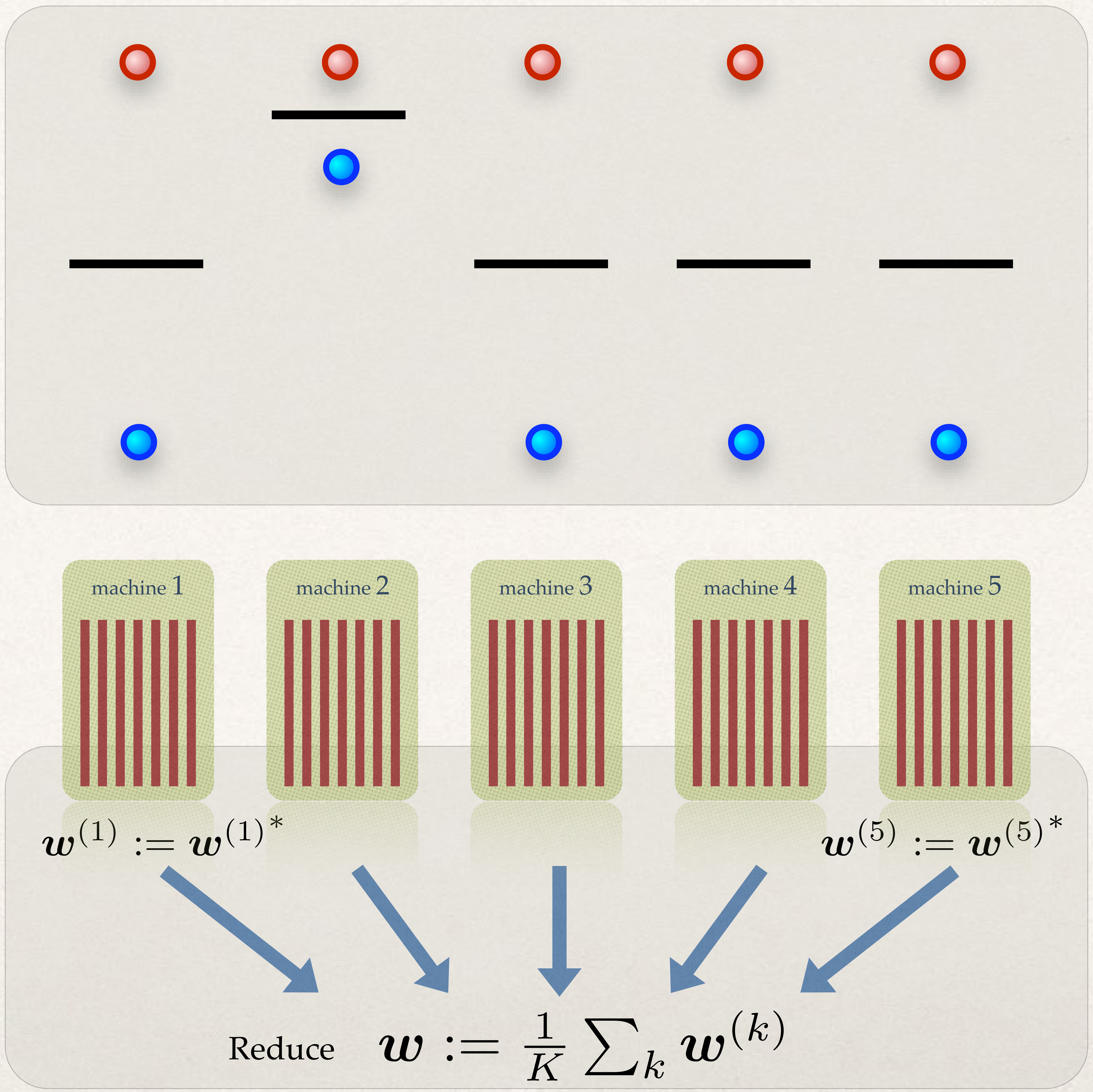
Challenge

The Cost of Communication

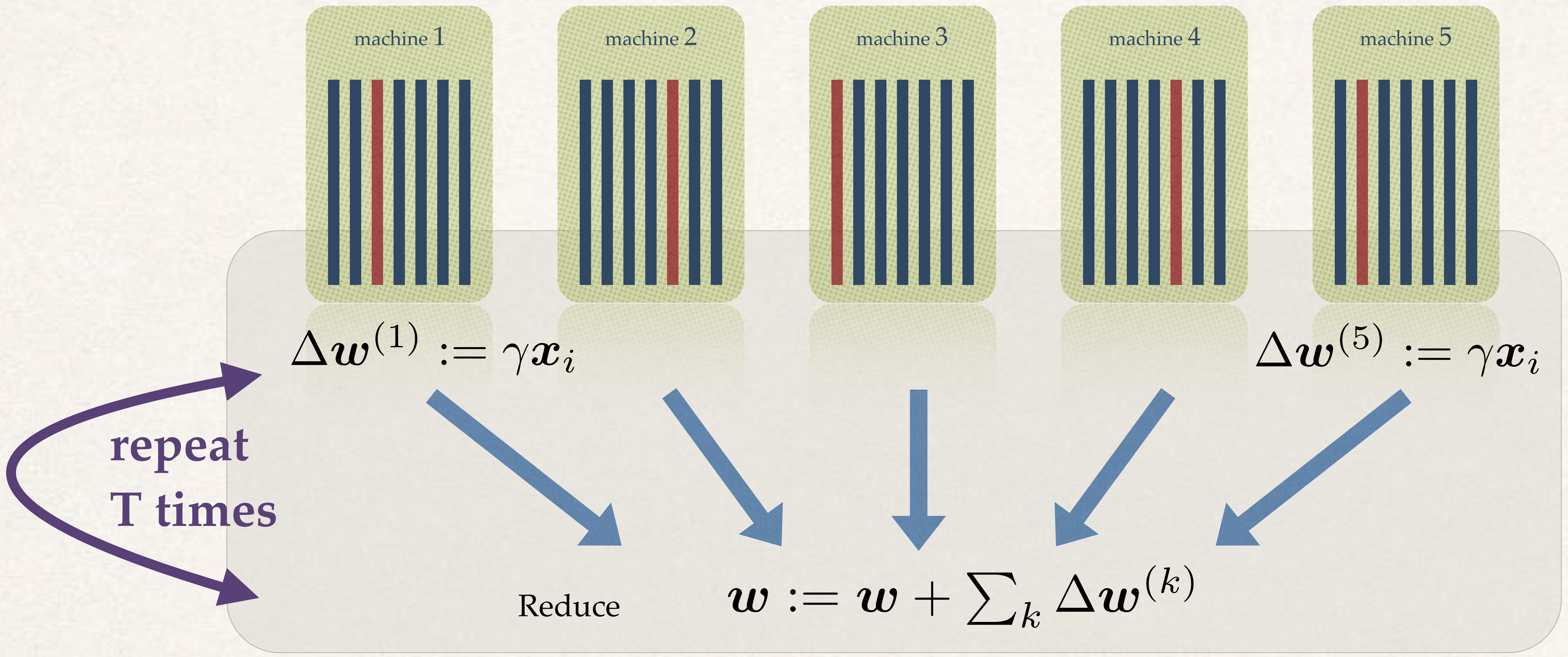


High-Performance Distributed Machine Learning using Apache Spark
Dünner et al. 2016, arxiv.org/abs/1612.01437

One-Shot Averaging Does Not Work



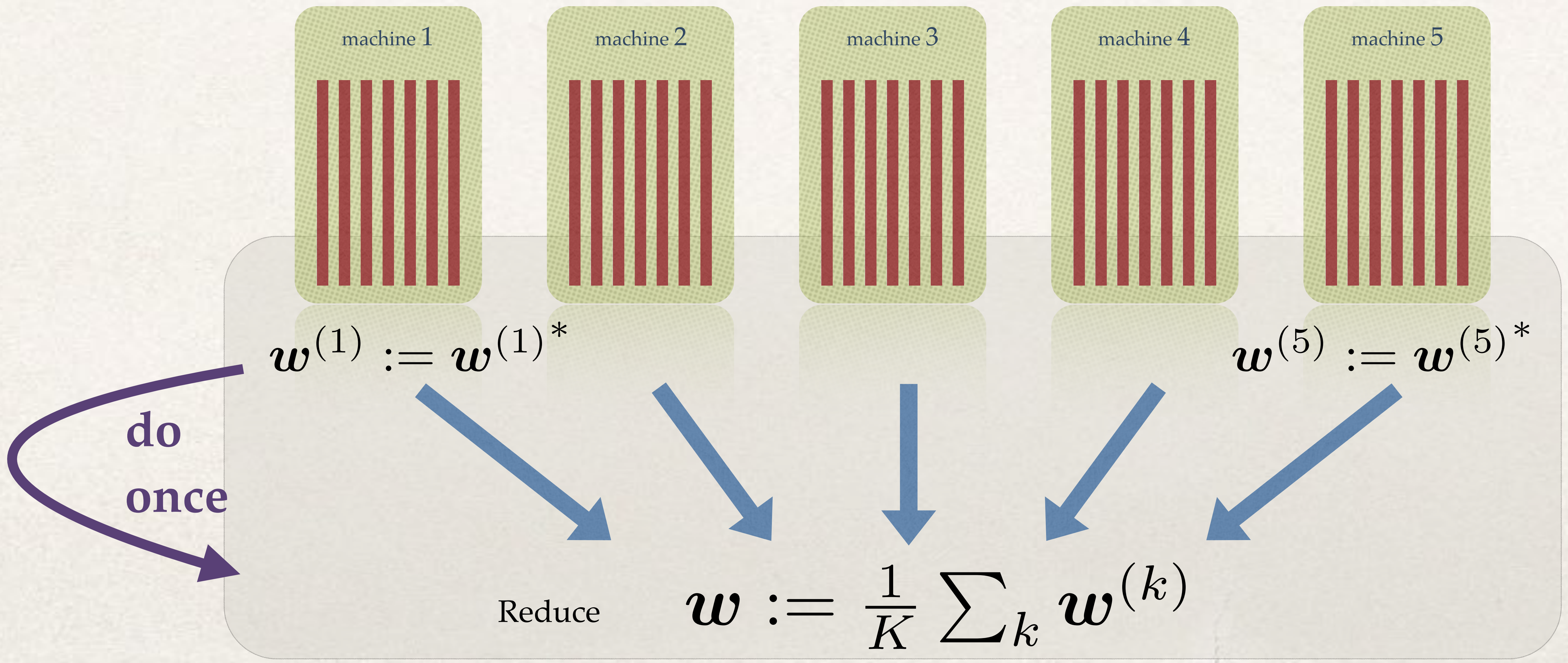
Communication: Always / Never



Naive Distributed SGD

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

"always communicate"



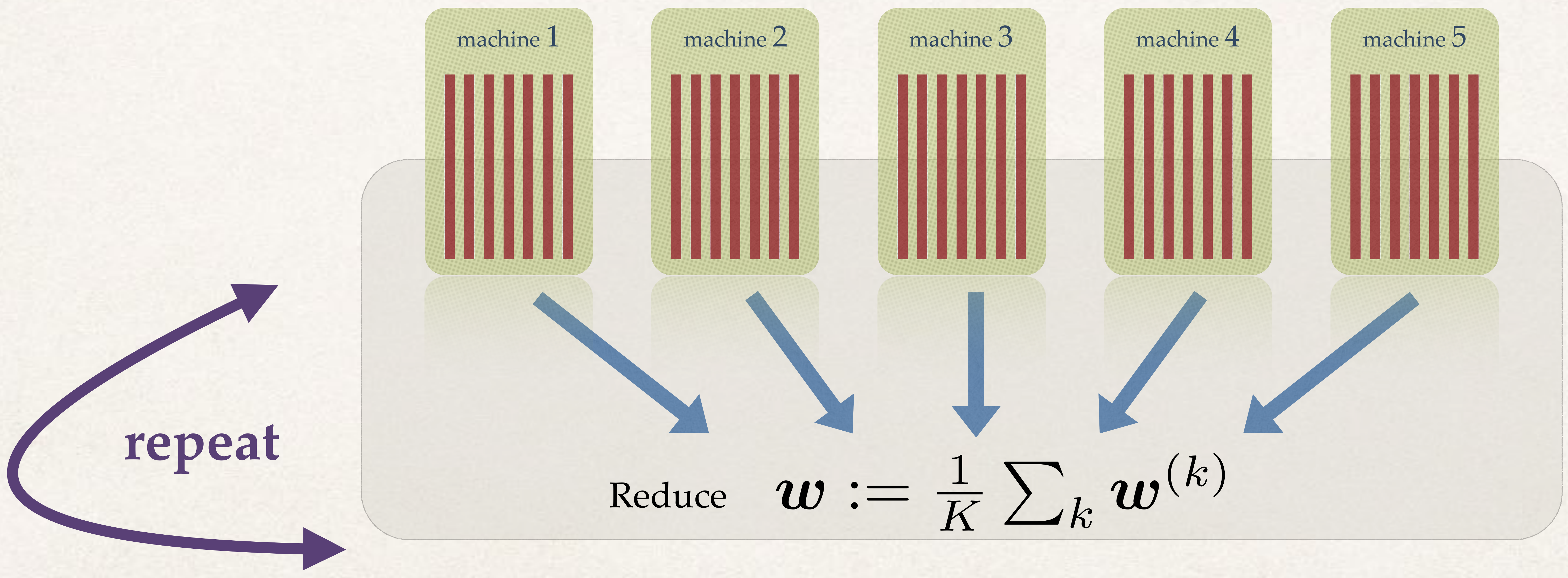
One-Shot Averaged Distributed Optimization

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

"never communicate"

Distributed Full Gradient, L-BFGS

(just distribute the full gradient computation)



Problem class

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \quad f(A\boldsymbol{\alpha}) \; + \; g(\boldsymbol{\alpha})$$

Optimization: Primal-Dual Context

$$A_{\text{loc}} \Delta \boldsymbol{\alpha}_{[k]} + \boldsymbol{w}$$

$$\min_{\boldsymbol{\alpha} \in \mathbb{R}^n} \quad \Big[\; \mathcal{O}_A(\boldsymbol{\alpha}) := f(A\boldsymbol{\alpha}) + g(\boldsymbol{\alpha}) \; \Big]$$

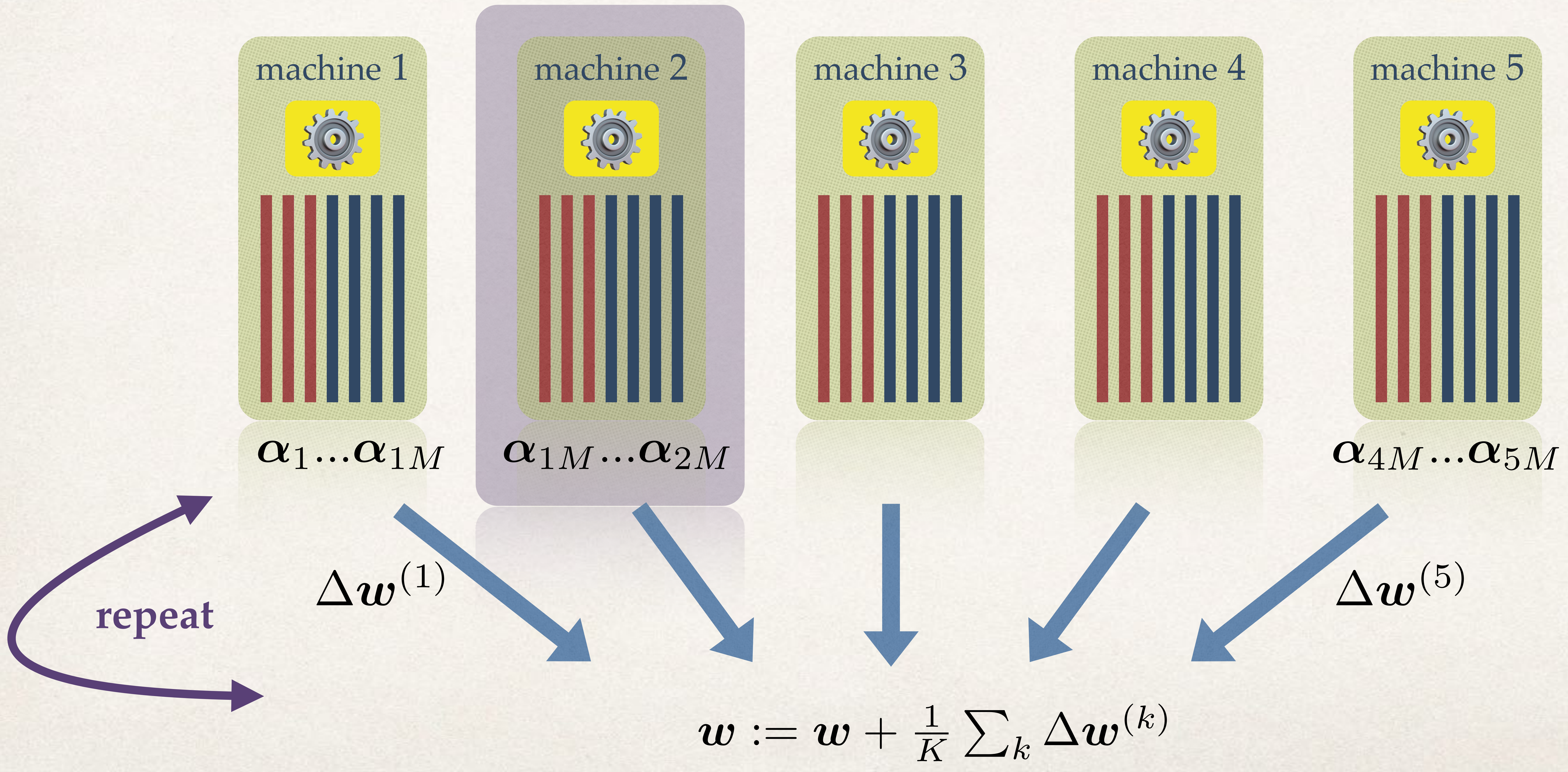
primal Lasso
dual L2-reg SVM/Log-Regr
primal L1-reg SVM/Log-Reg

correspondence

$$\boldsymbol{w} := \nabla f(A\boldsymbol{\alpha})$$

$$\min_{\mathbf{w} \in \mathbb{R}^d} \quad \Big[\; \mathcal{O}_B(\mathbf{w}) := g^*(-A^\top \mathbf{w}) + f^*(\mathbf{w}) \; \Big]$$

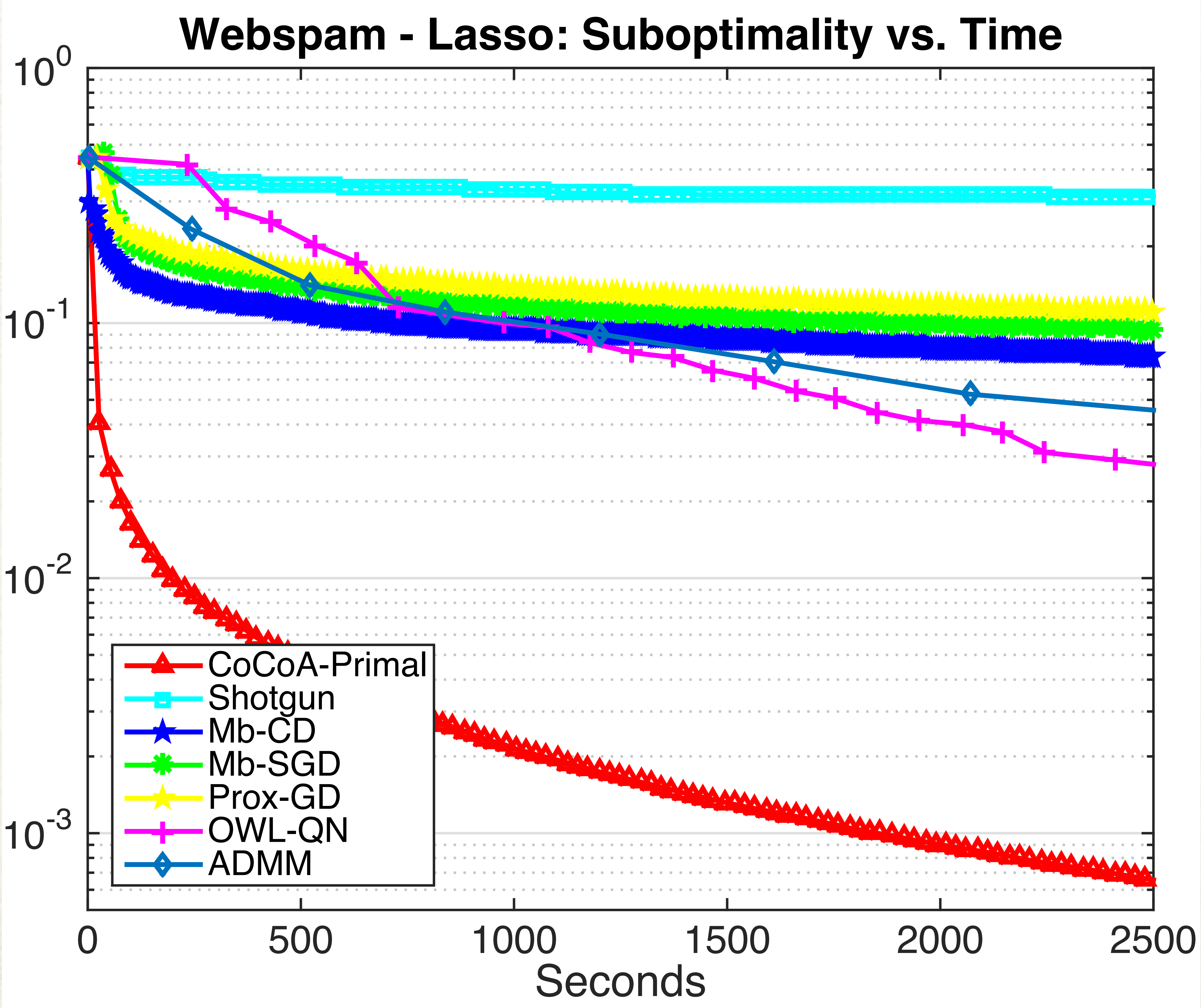
CoCoA - Communication Efficient Distributed Optimization



Distributed Experiments

Dataset	Training	Features	Sparsity
url	2,396,130	3,231,961	3.5e-3%
epsilon	400,000	2,000	100%
kddb	19,264,097	29,890,095	9.8e-5%
webspam	350,000	16,609,143	0.02%

L1-Regularized Linear Regression



NIPS 2014, ICML 2015, JMLR 2018
arxiv.org/abs/1611.02189

- part of TensorFlow core (L2)
- *code in pytorch, TF, spark, C (also L1)*

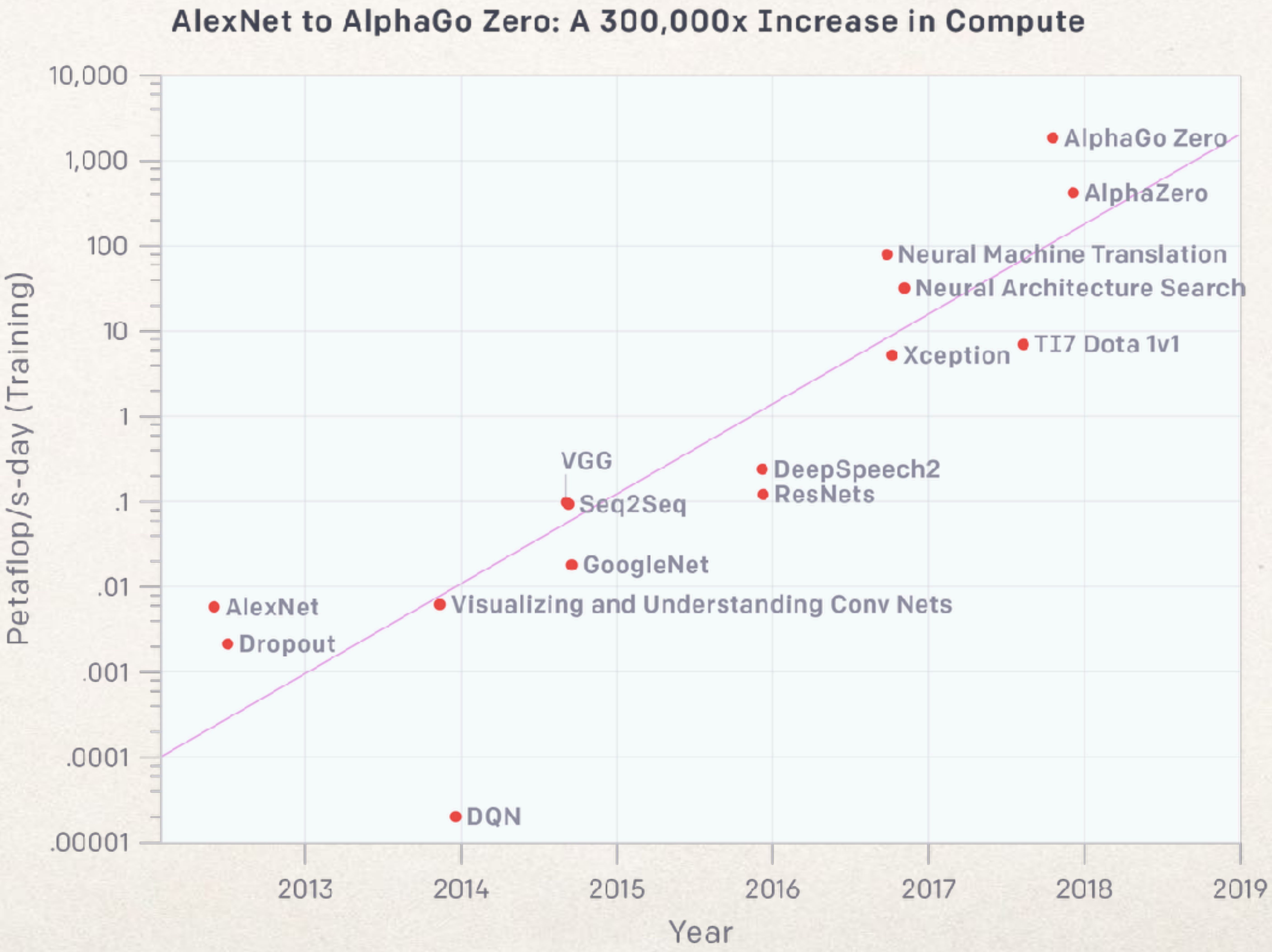
Summary

- **adaptivity** to the communication cost
- **re-usability** of good existing solvers
- **accuracy** certificates
- **second-order** and **trust-region** version (local Hessian)

Next Steps

- **adaptivity** to the degree of separability
- generalization to **deep learning, SGD**
- **decentralized version** (communication on graph)
- **benchmarking** & code

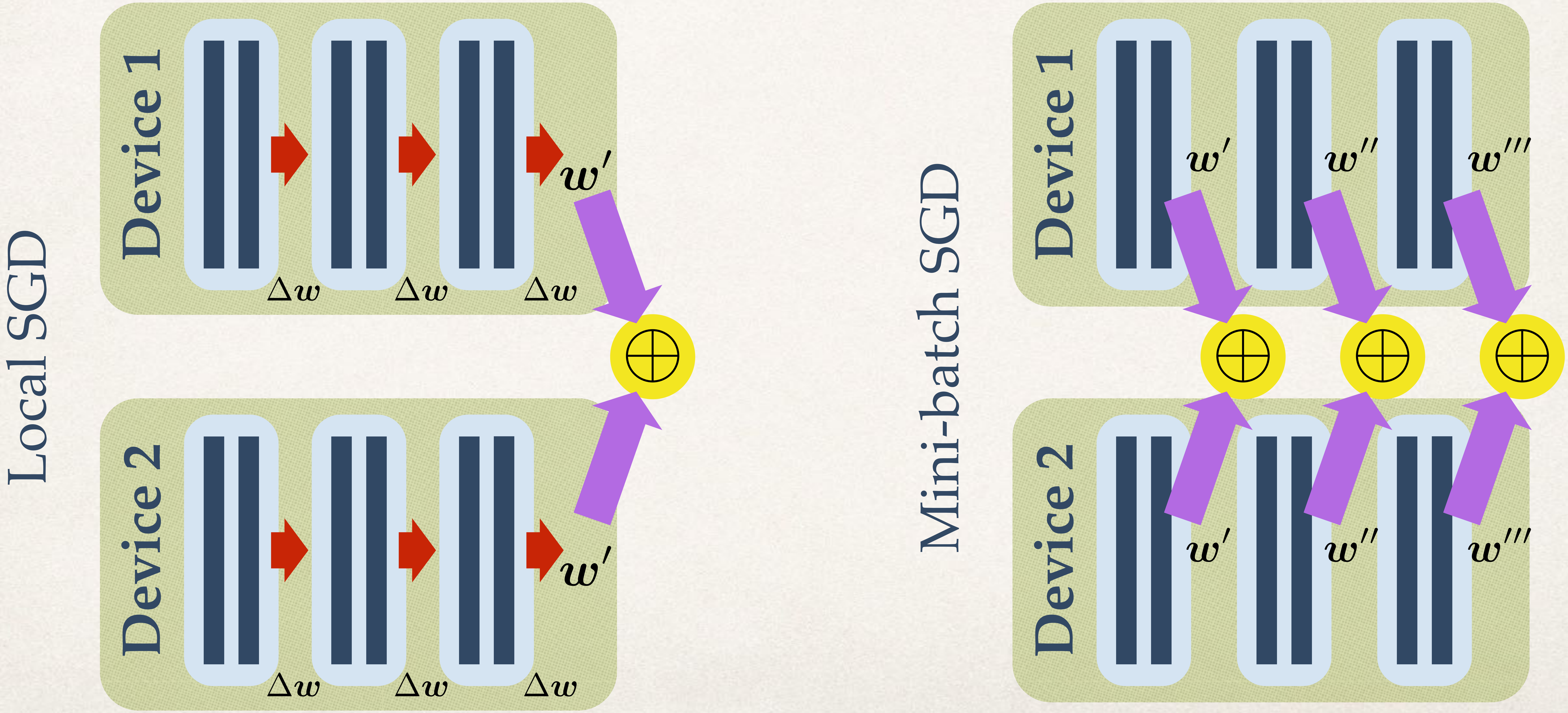
Deep Learning



https://blog.openai.com/ai-and-compute/

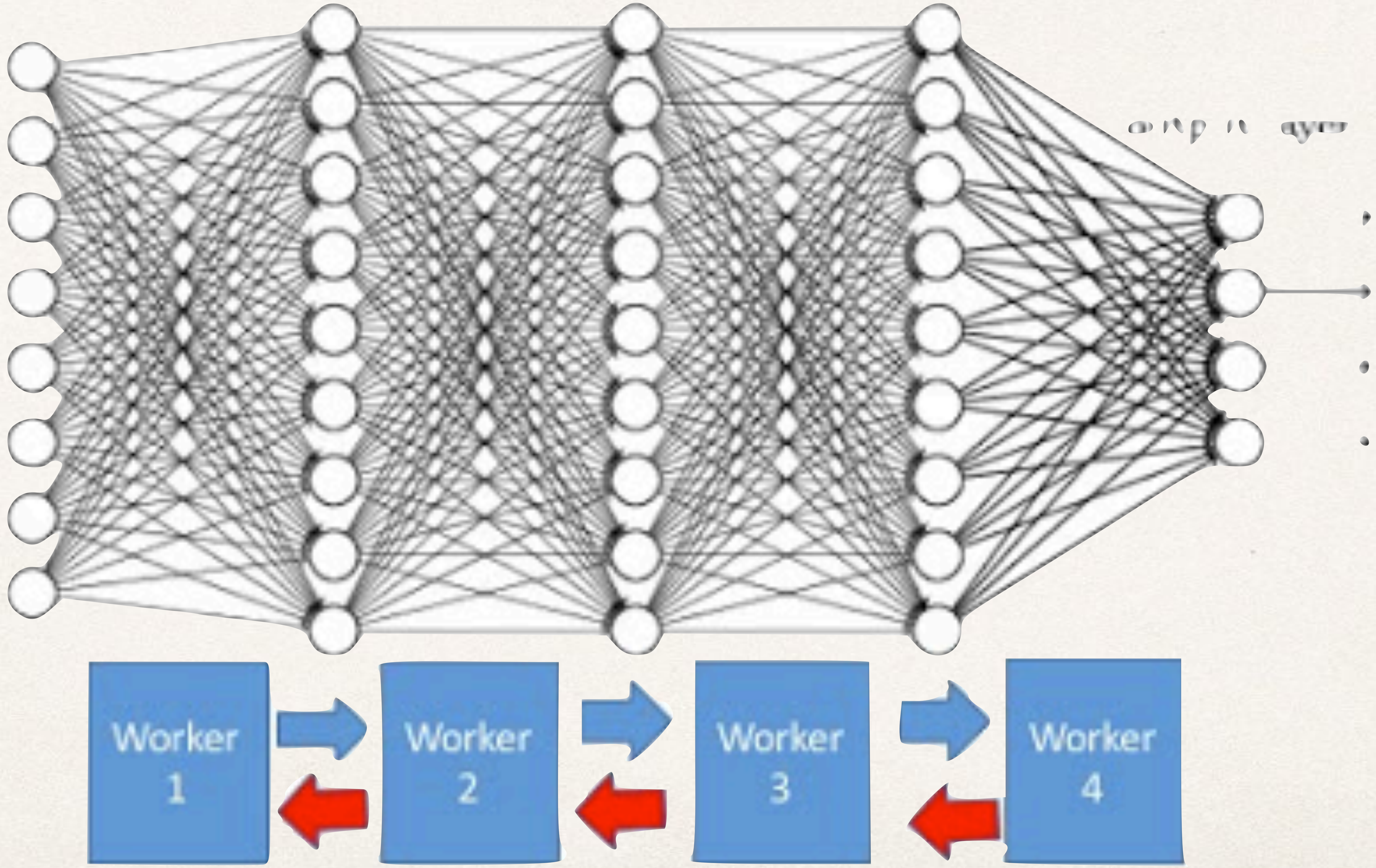
(Data Parallel)

Distributed DL



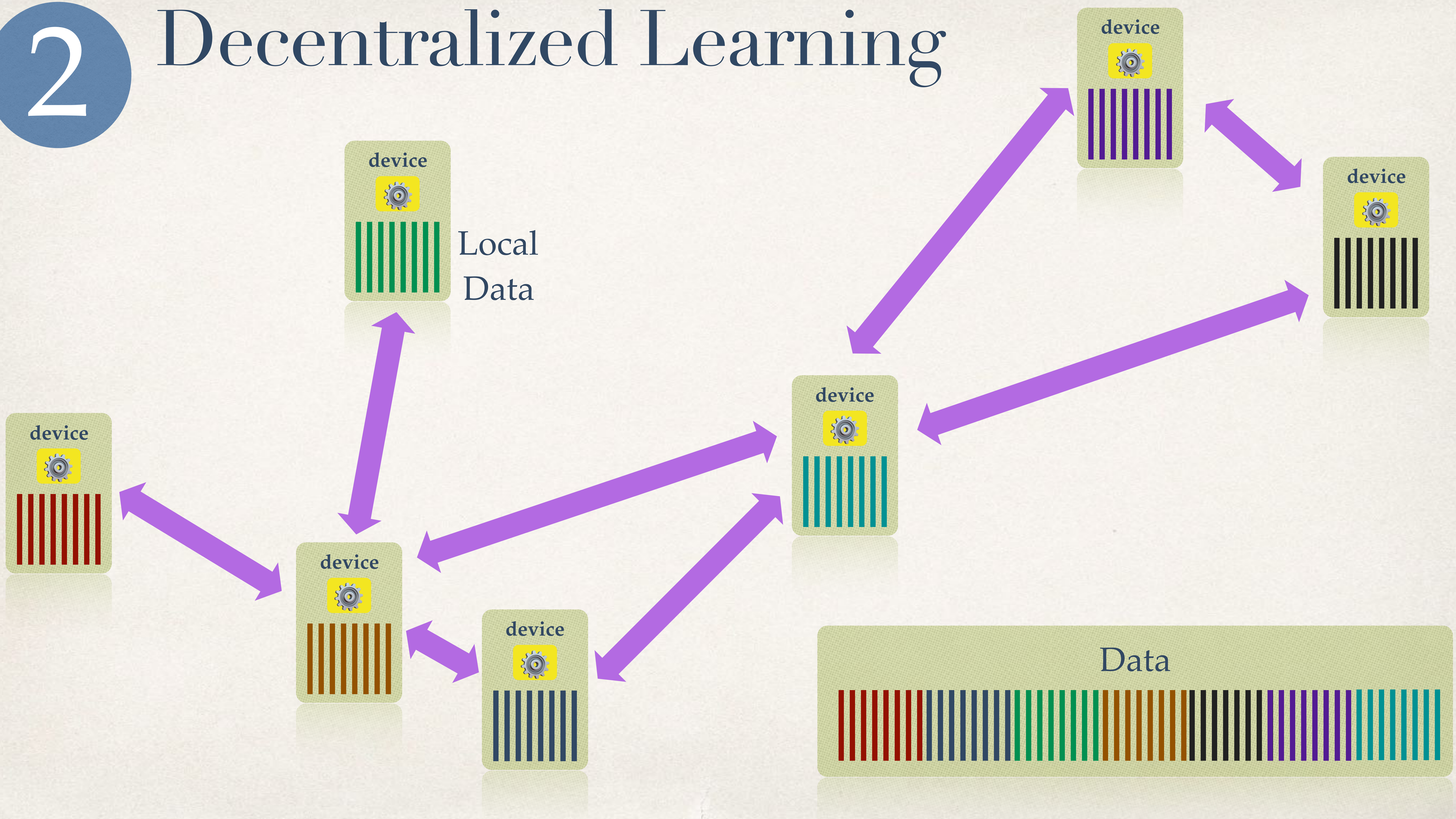
(Model Parallel)

Distributed DL



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

2 Decentralized Learning



Motivation

image source

- Medical data is very sensitive
- Data cannot be sent outside of the hospital
- People could use the data to prevent diseases

Motivation

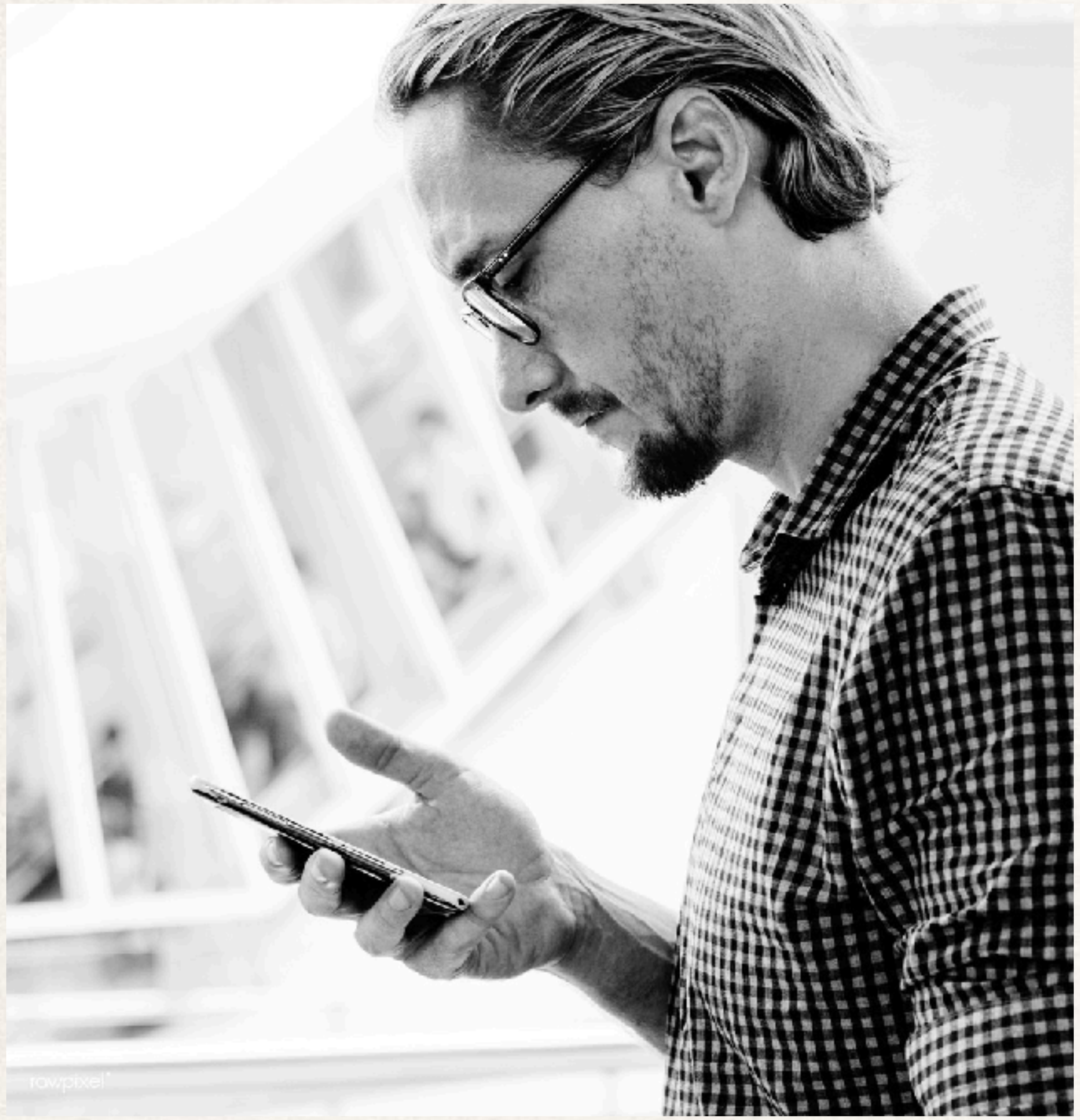
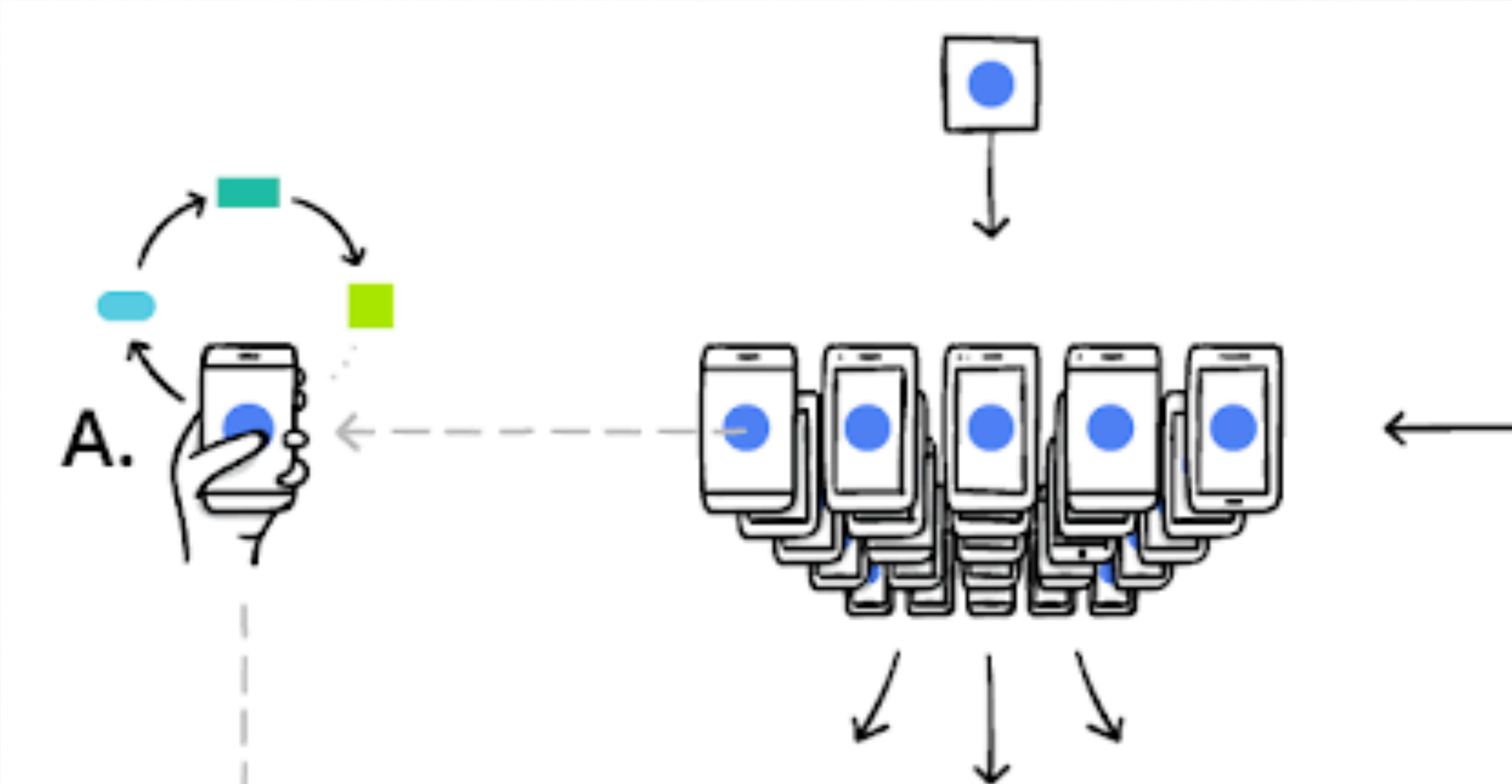


image source

- learn from users writing on smartphones
- data is very sensitive

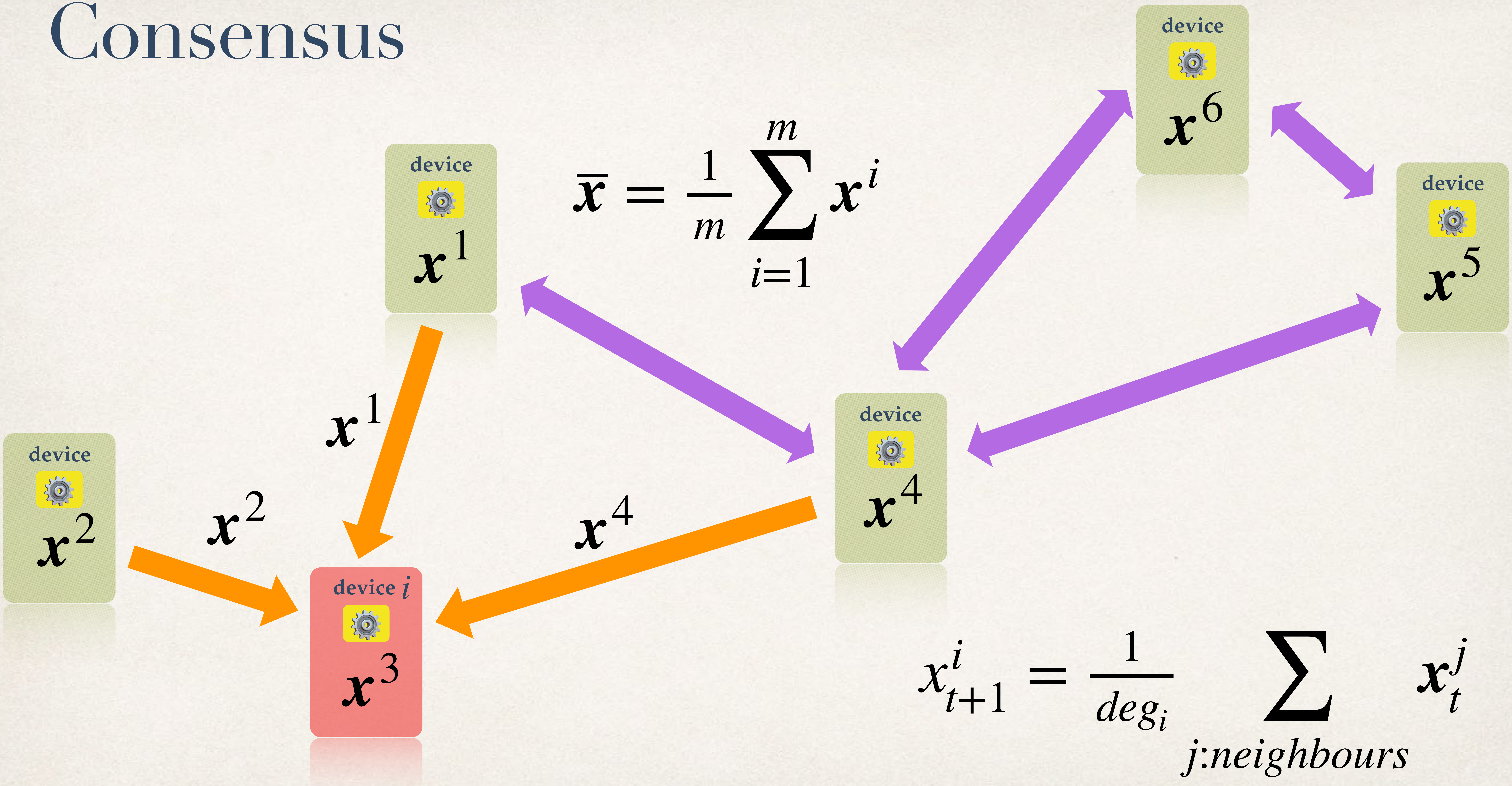


Motivation

image source

- Decentralized learning can speed up training time in datacenters

Consensus



$$\bar{\boldsymbol{x}} = \frac{1}{m}\sum_{i=1}^{m} \boldsymbol{x}^i$$

$$x^i_{t+1} = \frac{1}{deg_i} \sum_{j:neighbours} \boldsymbol{x}^j_t$$

Communication Compression

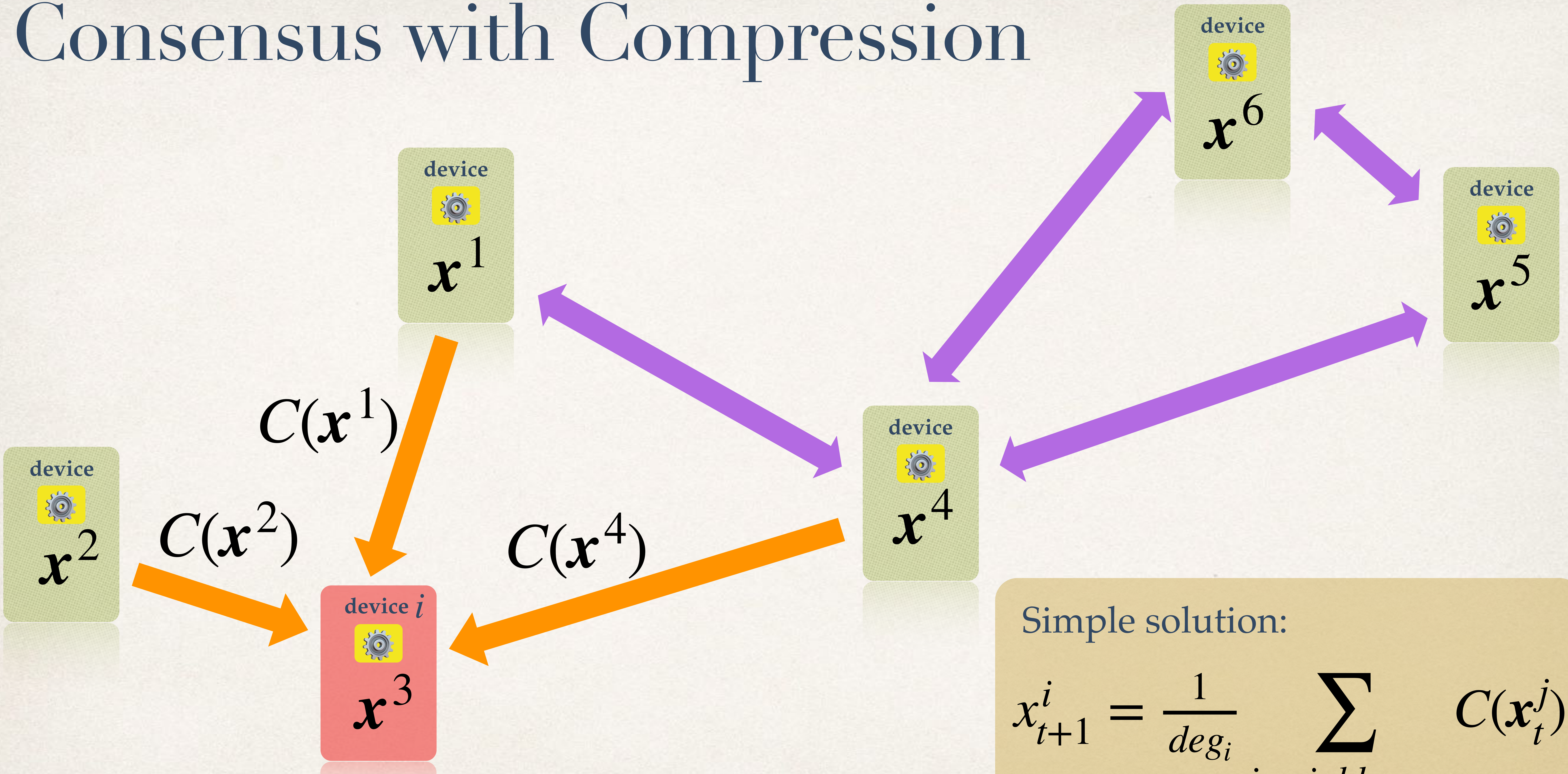
- limited-bit precision vector

e.g. 1-bit per entry reduces communication 32 times

- random/top k% of all the entries

e.g. k=0.1% reduces communication 1000 times

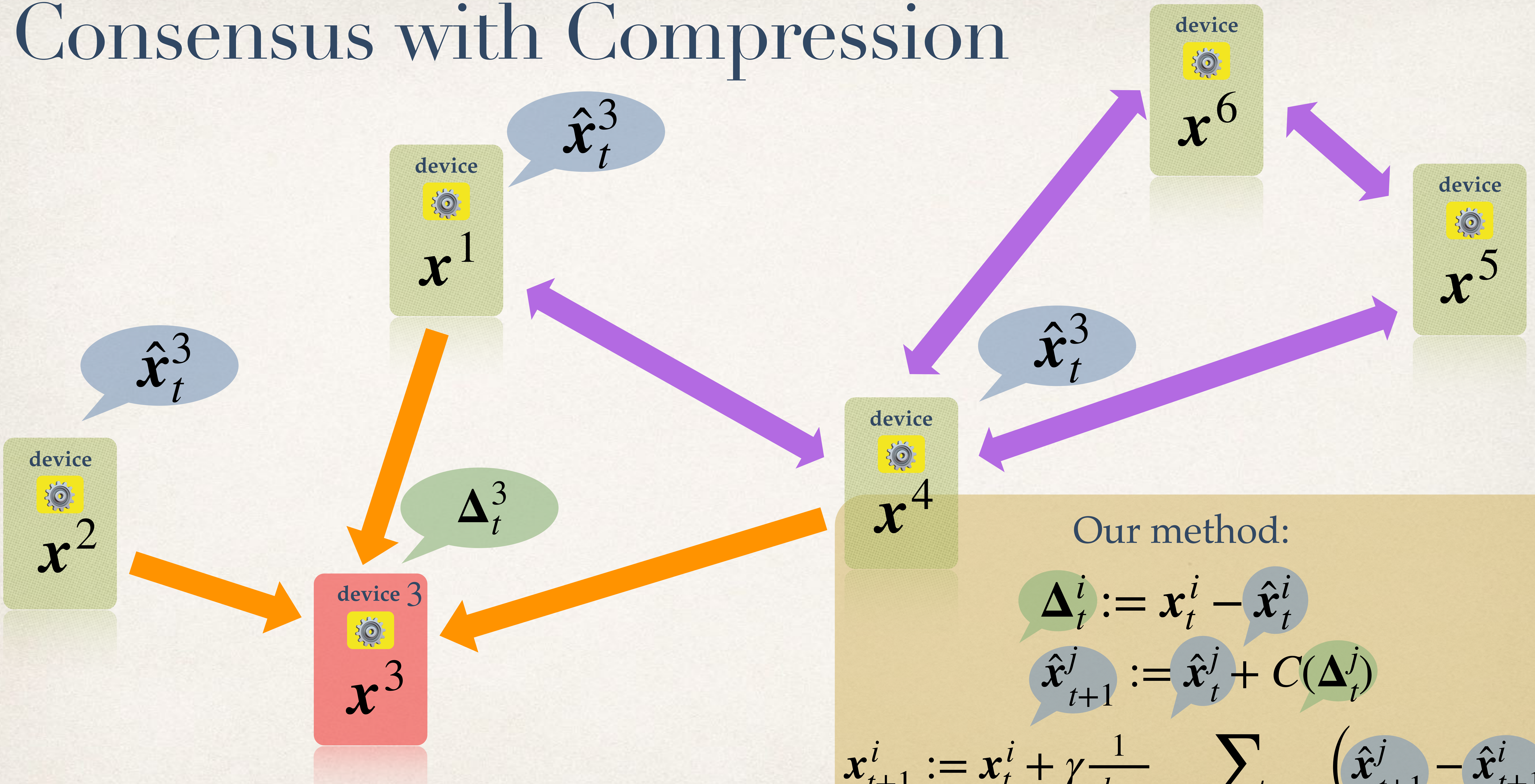
Consensus with Compression



Simple solution:

$$x^i_{t+1} = \frac{1}{deg_i} \sum_{j:neighbours} C(x^j_t)$$

Consensus with Compression



Our method:

$$\boldsymbol{\Delta}_t^i := \boldsymbol{x}_t^i - \hat{\boldsymbol{x}}_t^i$$

$$\hat{\boldsymbol{x}}_{t+1}^j := \hat{\boldsymbol{x}}_t^j + C(\boldsymbol{\Delta}_t^j)$$

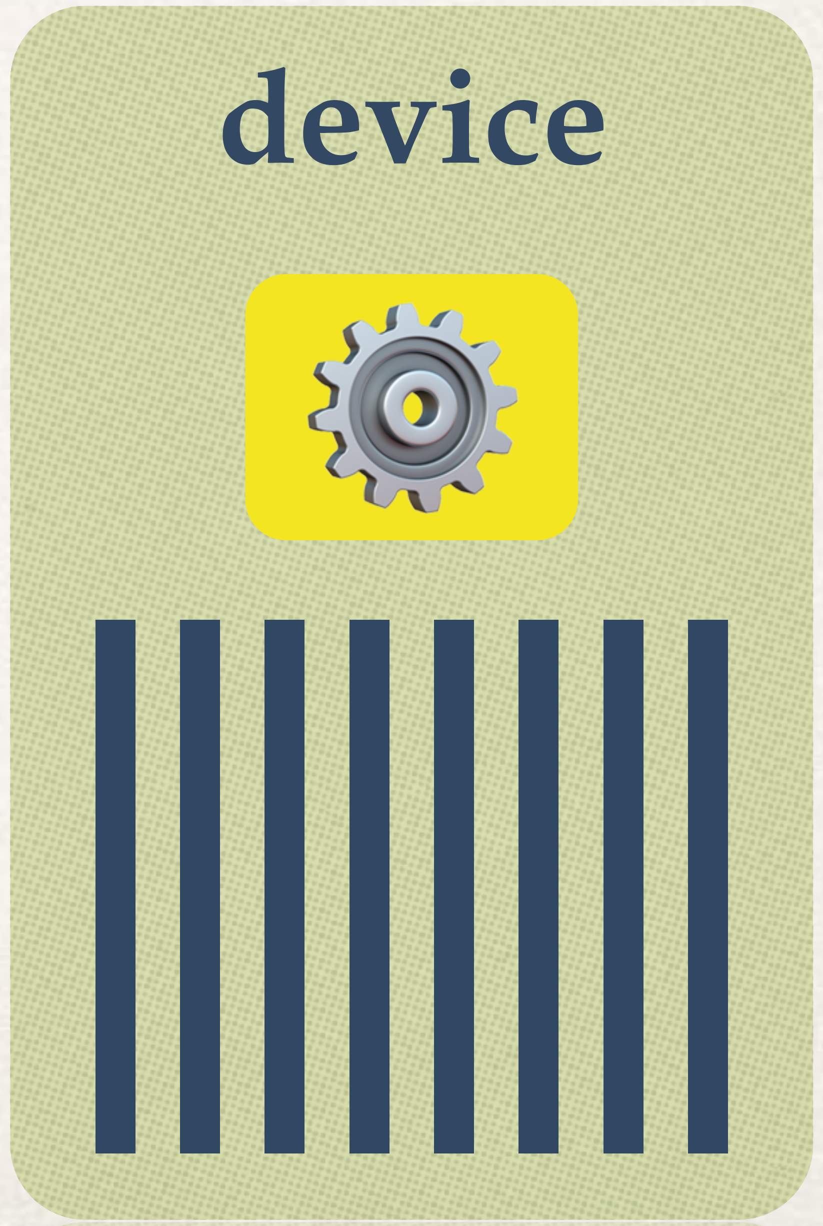
$$\boldsymbol{x}_{t+1}^i := \boldsymbol{x}_t^i + \gamma \frac{1}{deg_i} \sum_{j:neighbours} \left(\hat{\boldsymbol{x}}_{t+1}^j - \hat{\boldsymbol{x}}_{t+1}^i \right)$$

Stochastic Gradient Descent (SGD)

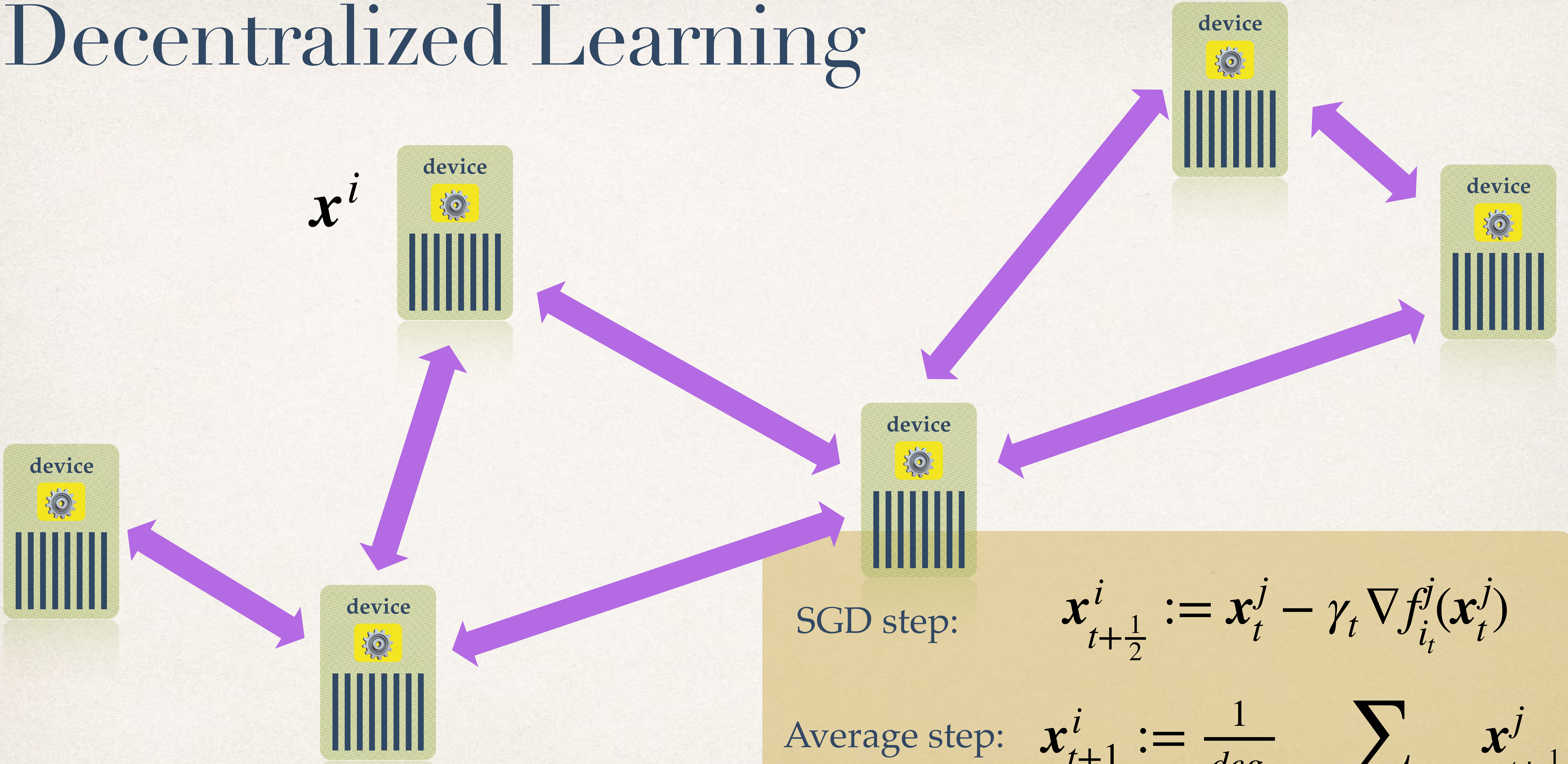
$$\min_{\boldsymbol{x}} \; f(\boldsymbol{x}) = \frac{1}{|data|} \sum_{i \in data} f_i(\boldsymbol{x})$$

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

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



Decentralized Learning



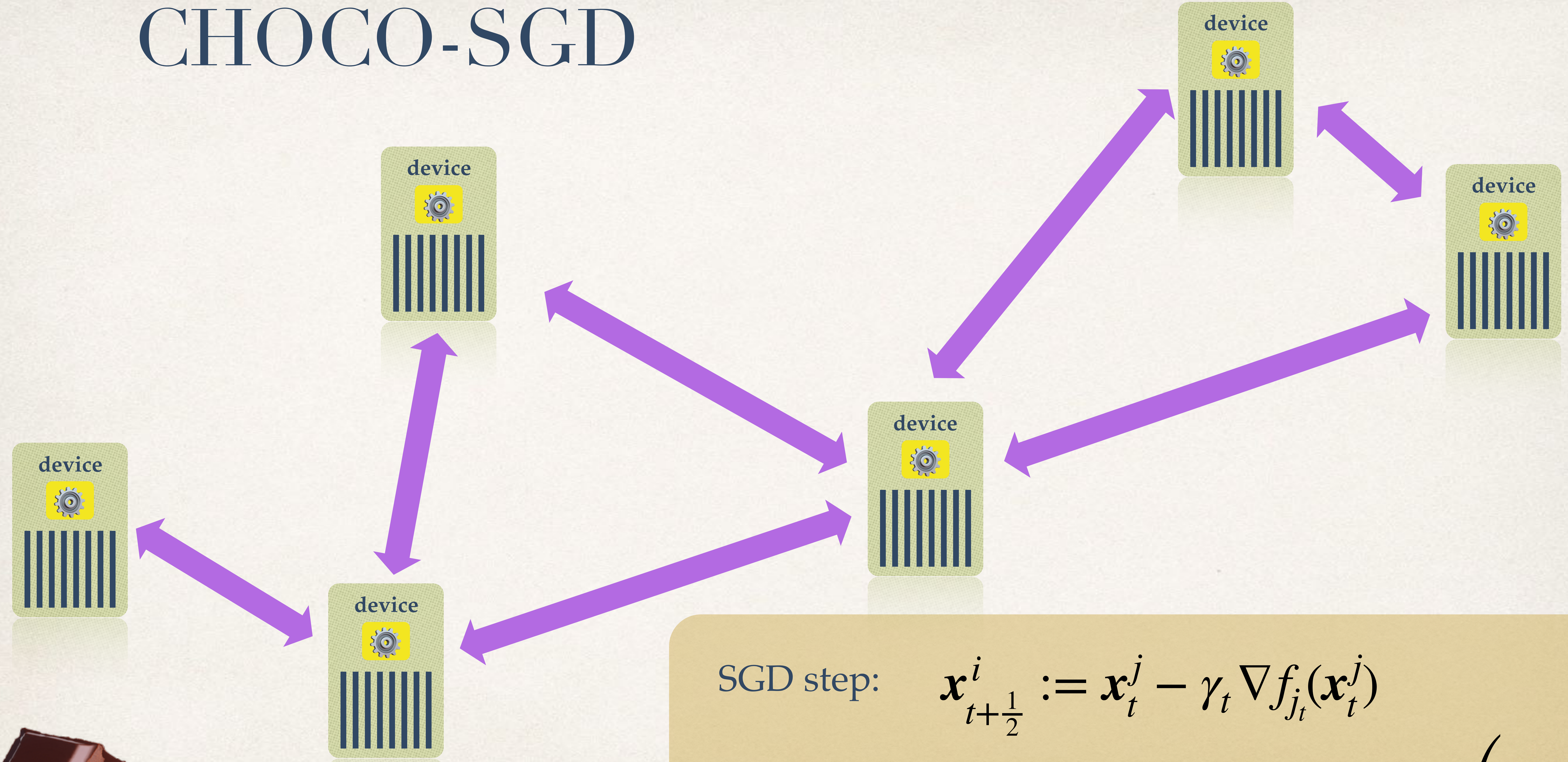
SGD step: $\boldsymbol{x}^i_{t+\frac{1}{2}} := \boldsymbol{x}^j_t - \gamma_t \nabla f^j_{i_t}(\boldsymbol{x}^j_t)$

Average step: $\boldsymbol{x}^i_{t+1} := \frac{1}{deg_i} \sum_{j:neighbours} \boldsymbol{x}^j_{t+\frac{1}{2}}$

CHOCO-SGD

image source

CHOCO-SGD



SGD step: $$\boldsymbol{x}^i_{t+\frac{1}{2}} := \boldsymbol{x}^j_t - \gamma_t \nabla f_{j_t}(\boldsymbol{x}^j_t)$$

$$\boldsymbol{x}^i_{t+1} := consensus\_with\_compression\left(\boldsymbol{x}^j_{t+\frac{1}{2}}\right)$$

Theoretical Result

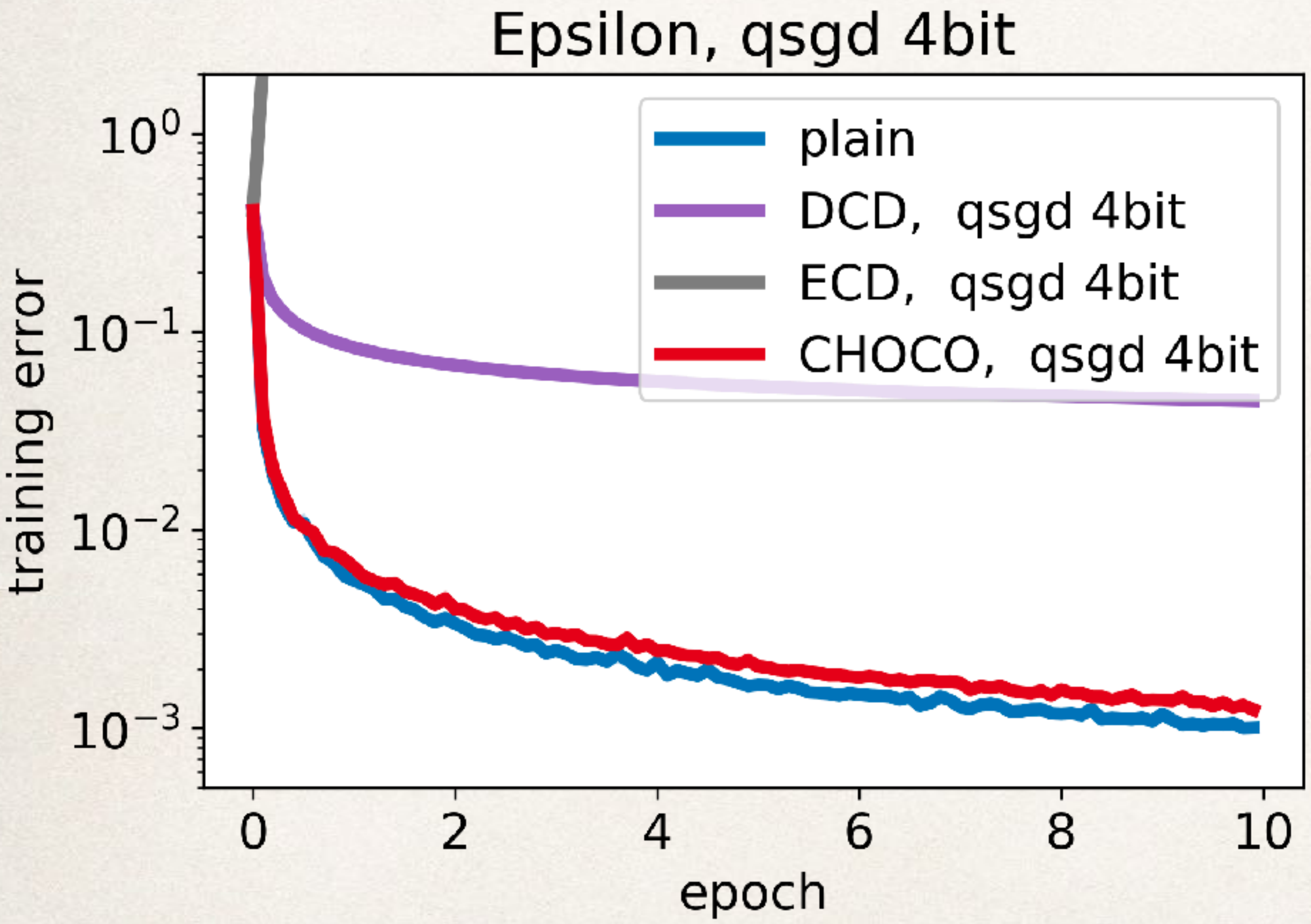
$$f(\boldsymbol{x}_{avg}^{(T)}) - f^\star = \mathcal{O}\left(\frac{1}{nT}\right) + \mathcal{O}\left(\frac{1}{\delta^2 T^2}\right) + \mathcal{O}\left(\frac{1}{\delta^3 T^3}\right)$$

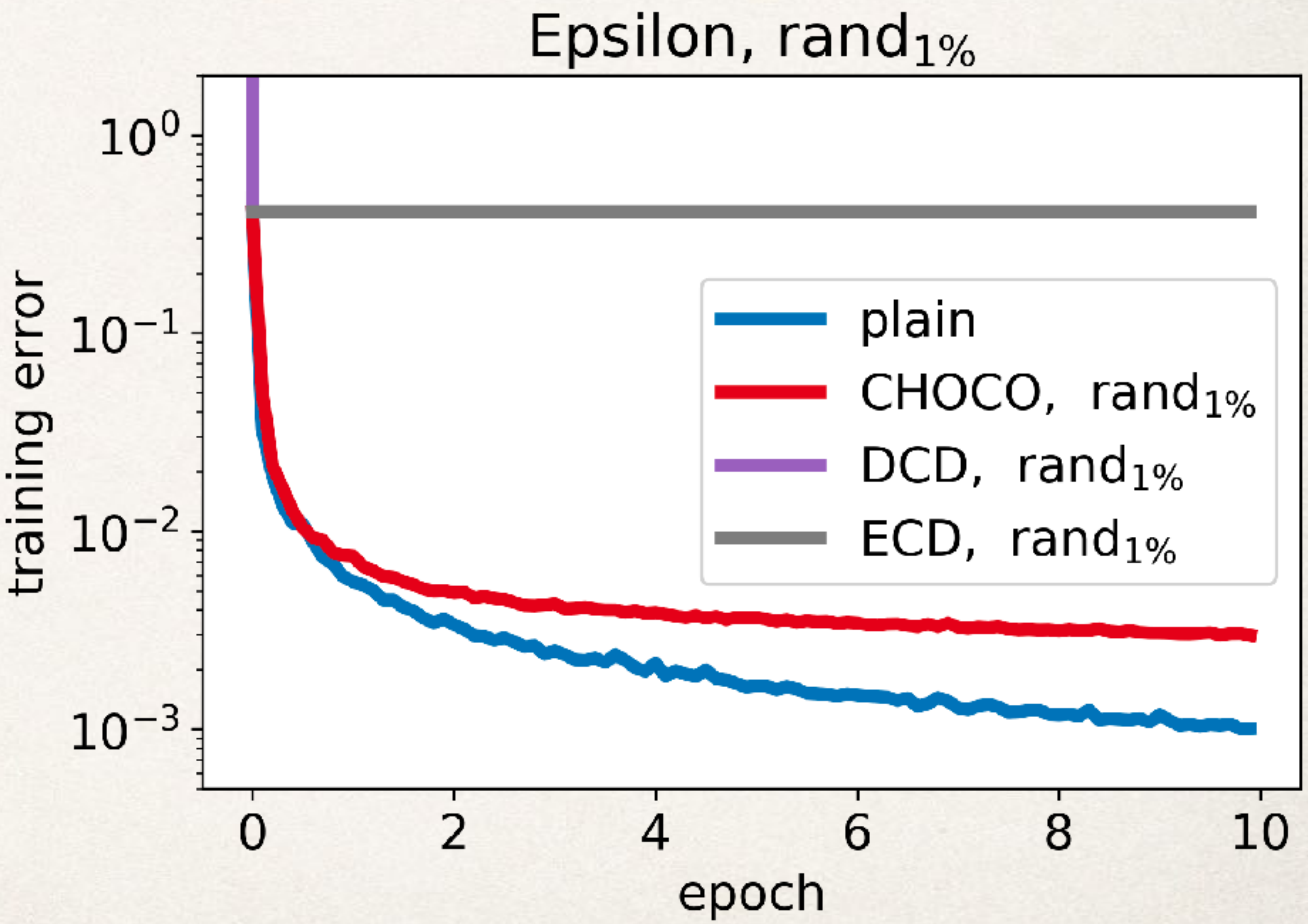
δ — compression ratio

$\delta \in [0,1], \quad \delta = 1$ for no compression

- (almost) same convergence rate as full communication

Experimental Results





CHOCO SGD

- First consensus algorithm that converges linearly with arbitrary compression
- First decentralized SGD algorithm that converges with arbitrary compression
- Practical performance

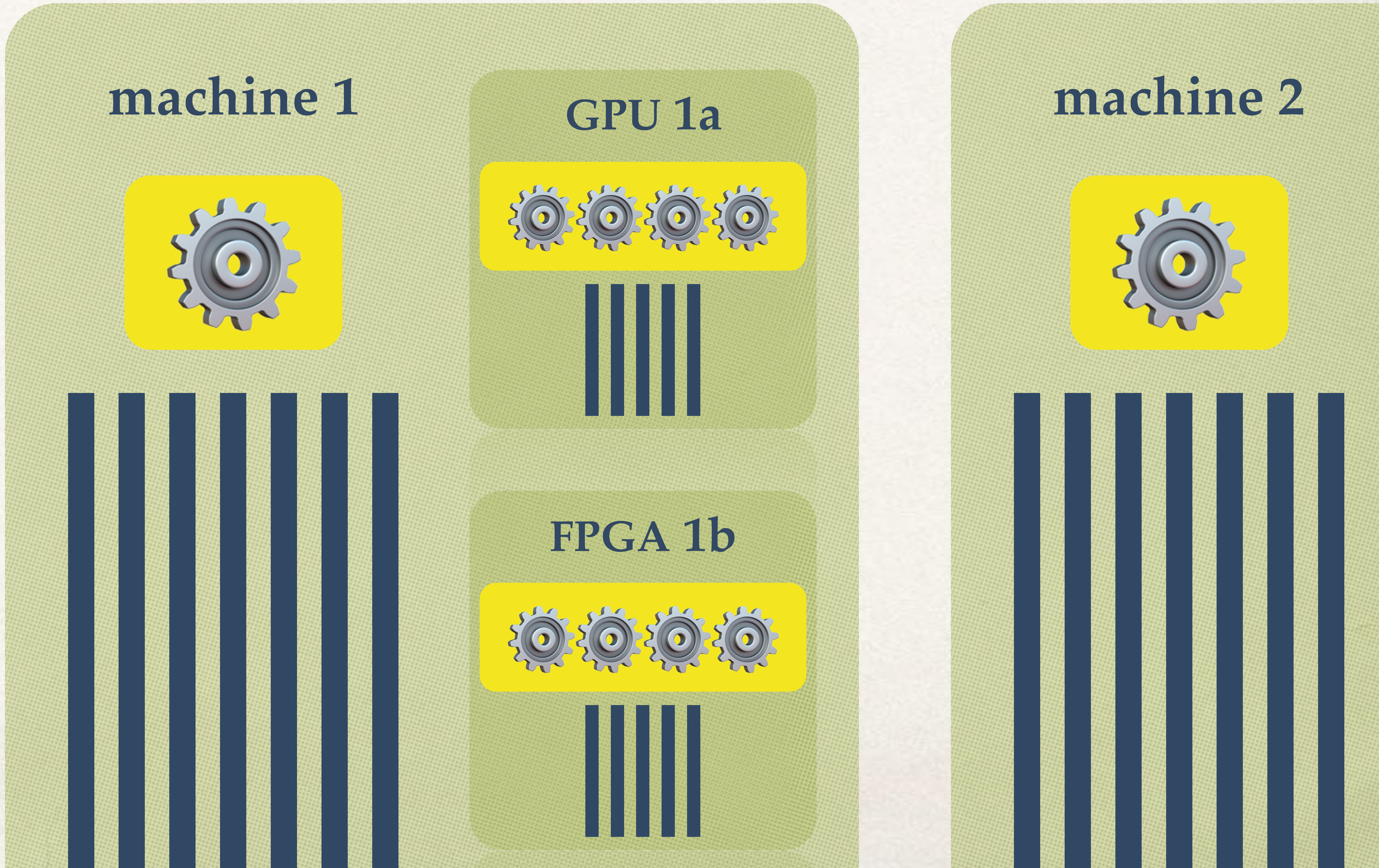
Future work

- Run it on deep learning experiments (multi GPU)
- Non-convex theory

3

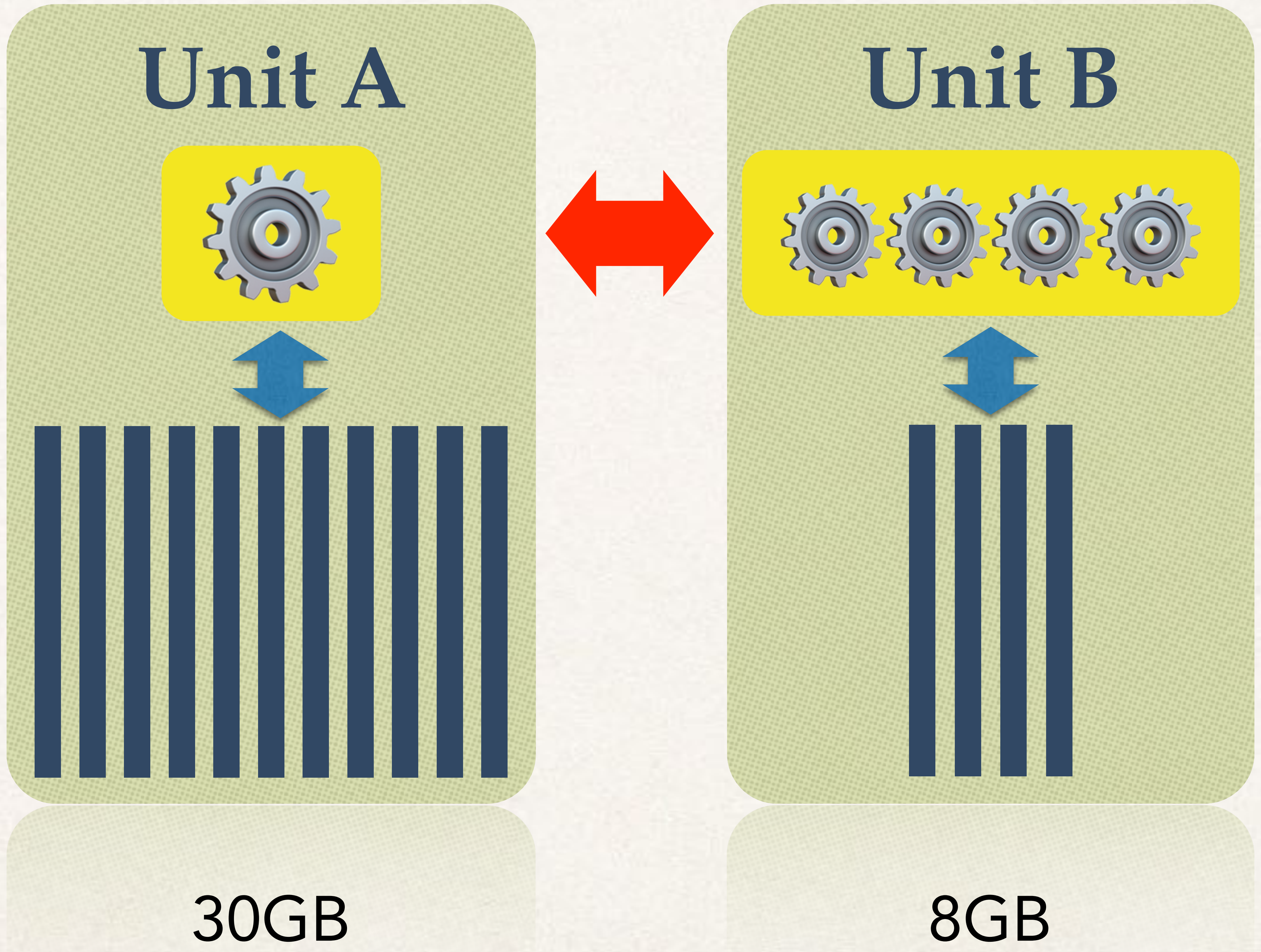
Leveraging Heterogenous Systems

Compute & Memory Hierarchy: Which data to put in which device?



Leveraging Heterogenous Systems

duality gap as selection criterion

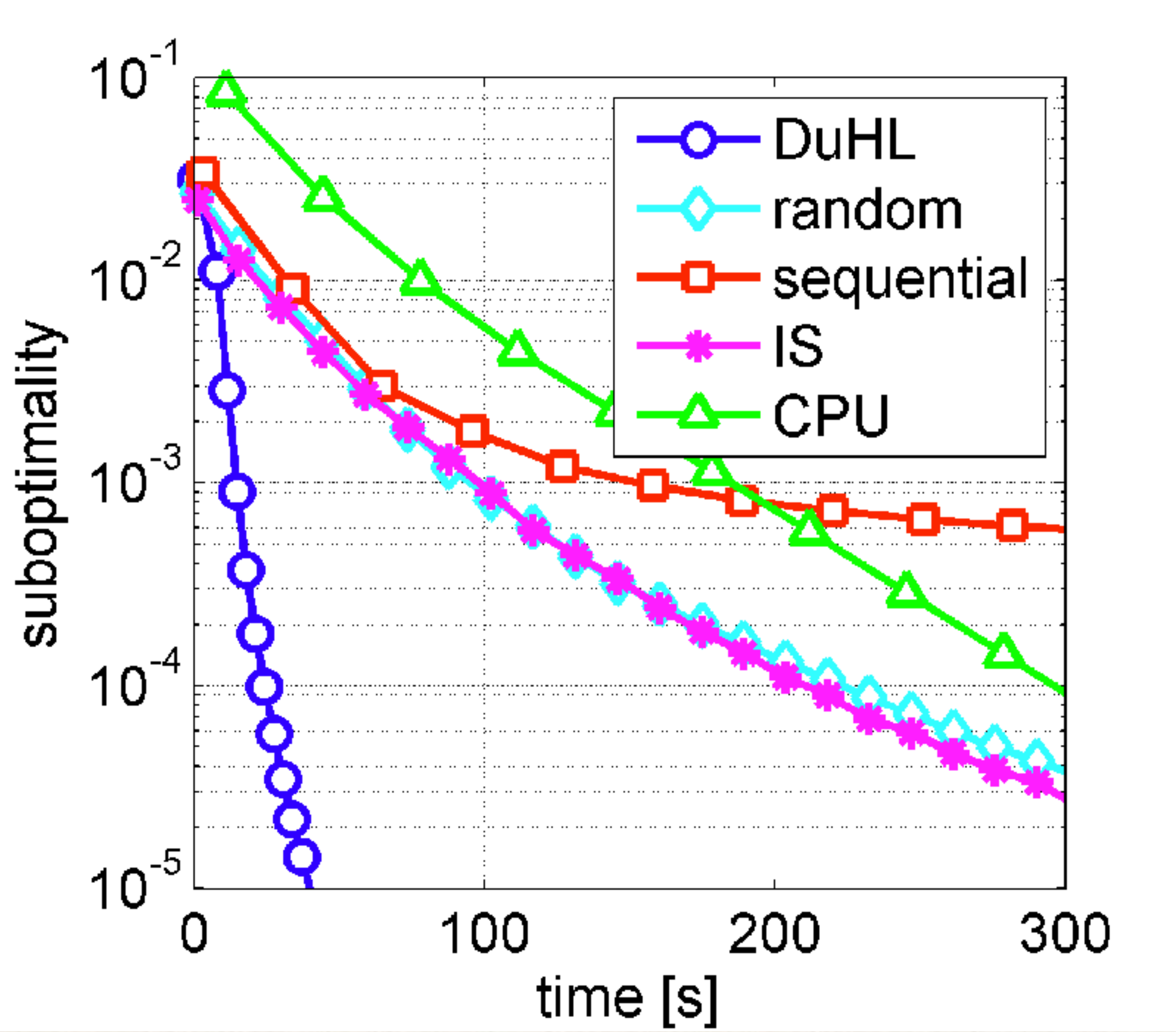


adaptive importance sampling

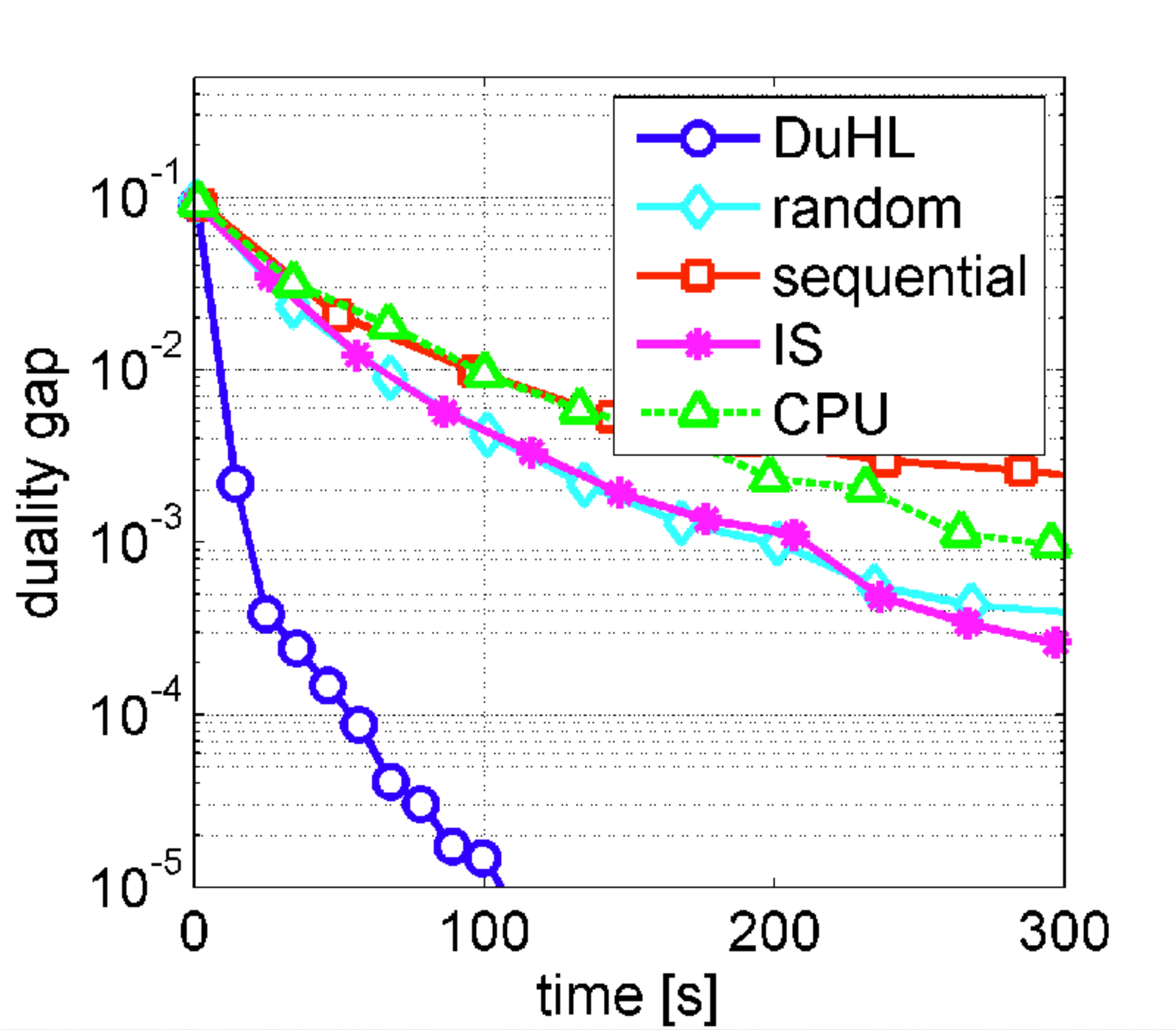
AISTATS 2017, 2018
NIPS 2017a,b

Experiments

RAM ⬌ GPU, 30GB dataset



Lasso



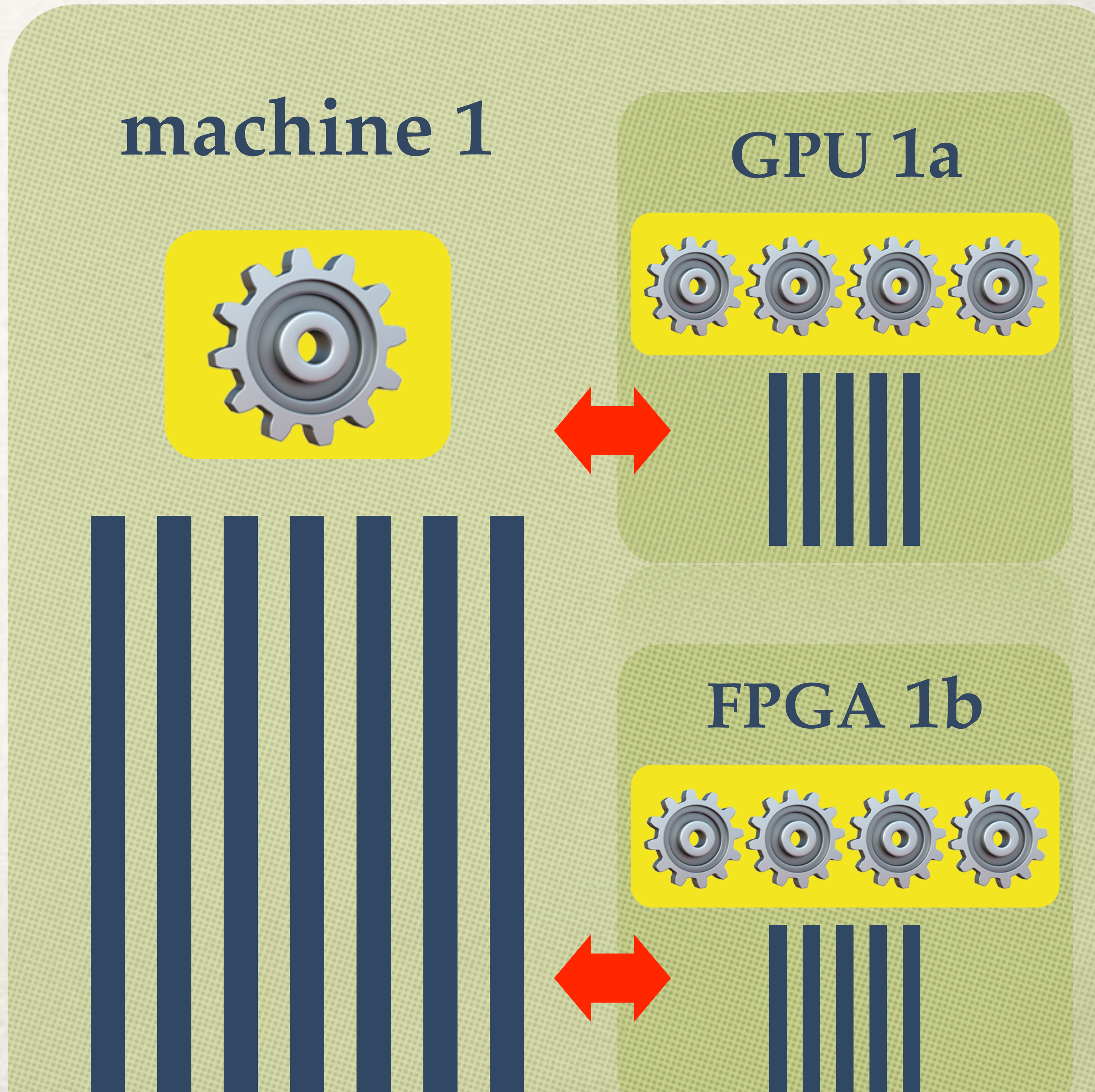
SVM

Experiments

terabyte click log dataset, IBM cloud implementation *[arXiv]*

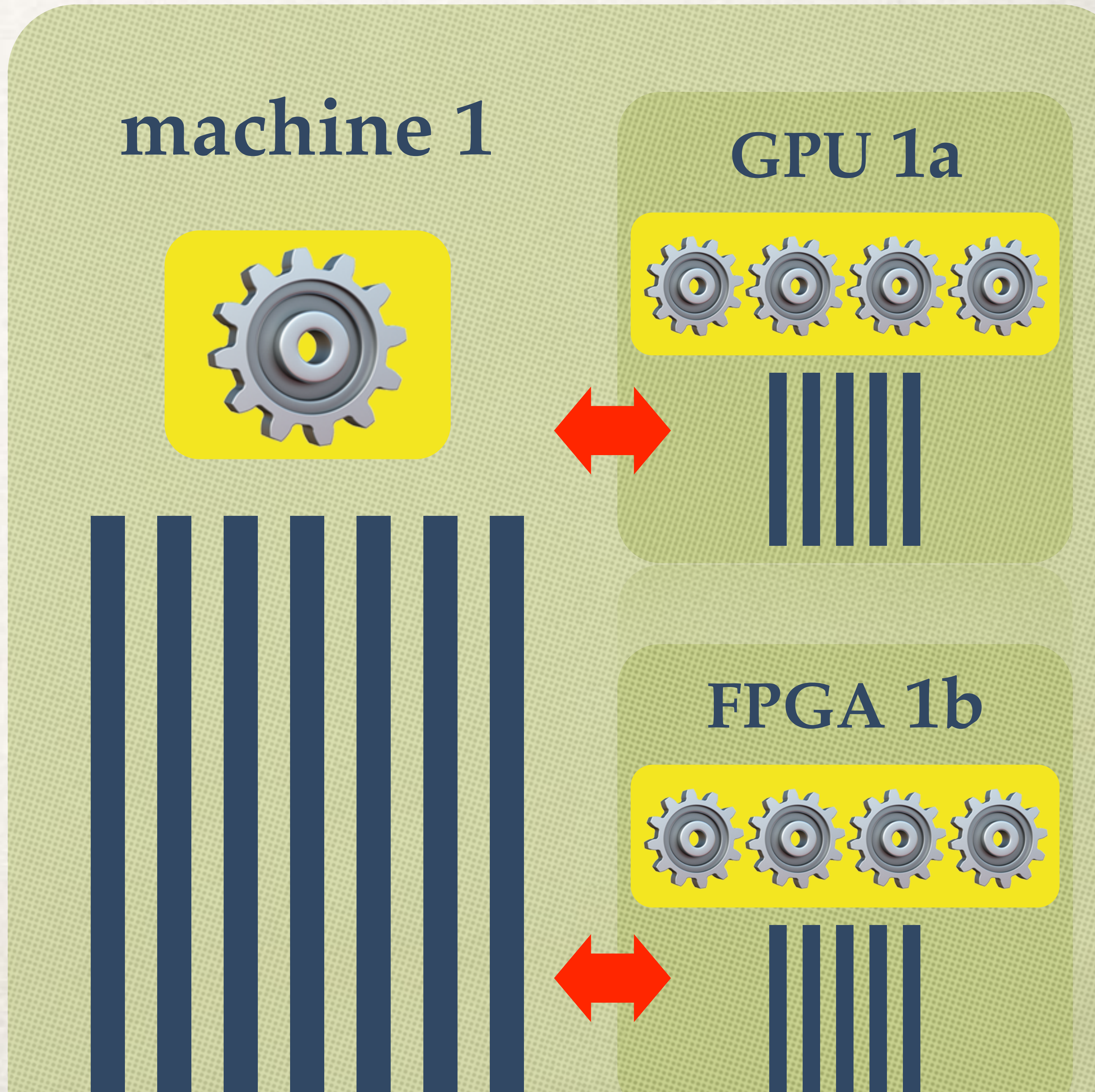
Open Research

- **limited precision operations** for efficiency of communication and computation
- **asynchronous** and **fault tolerant** algorithms
- heterogenous systems
- more **re-usable** algorithmic building blocks
  - for more systems and problems



Trends - Systems

- new **hardware**
  - TPU, GraphCore
  - sparse ops?
  - efficient numerics (limited precision), model compression
- **Software** frameworks
  - AutoGrad (Tensorflow, PyTorch, etc)
  - Communication?



Cloud ML


Microsoft Azure Machine Learning
Home
Studio
Gallery
Binary Classification: Direct marketing
In draft
Search experiment items
Saved Datasets
Data Format Conversions
Data Input and Output
Data Transformation
Feature Selection
Machine Learning
OpenCV Library Modules
Python Language Modules
R Language Modules
Statistical Functions
Text Analytics
Web Service
Deprecated
Reader
Metadata Editor
Project Columns
remove columns that are part
of the label
Split
Two-Class Boosted Decision T...
Two-Class Support Vector Ma...
Split
Split
Sweep Parameters
Sweep Parameters
Score Model
Score Model
Evaluate Model
1:1
Properties
Two-Class Boosted Decision Tree
Create trainer mode
Single Parameter
Maximum number of leav...
20
Minimum number of sam...
10
Learning rate
0.2
Number of trees construct...
100
Random number seed
0
Allow unknown categ...
Quick Help
Creates a binary classifier using a boosted
decision tree algorithm
(more help...)

Project:

Distributed Machine Learning Benchmark

Goal:

Public and Reproducible
Comparison of Distributed Solvers

github.com/mlbench/mlbench

PyTorch

Google

Apache

HPC

Auto ML

- **hyper-parameter optimization**
  zero-order methods
- **learning to learn**
  adaptive methods
- **neural architecture search**
  zero-order, warm-start

Thanks!

mlo.epfl.ch