Optimization for Machine Learning in Practice II

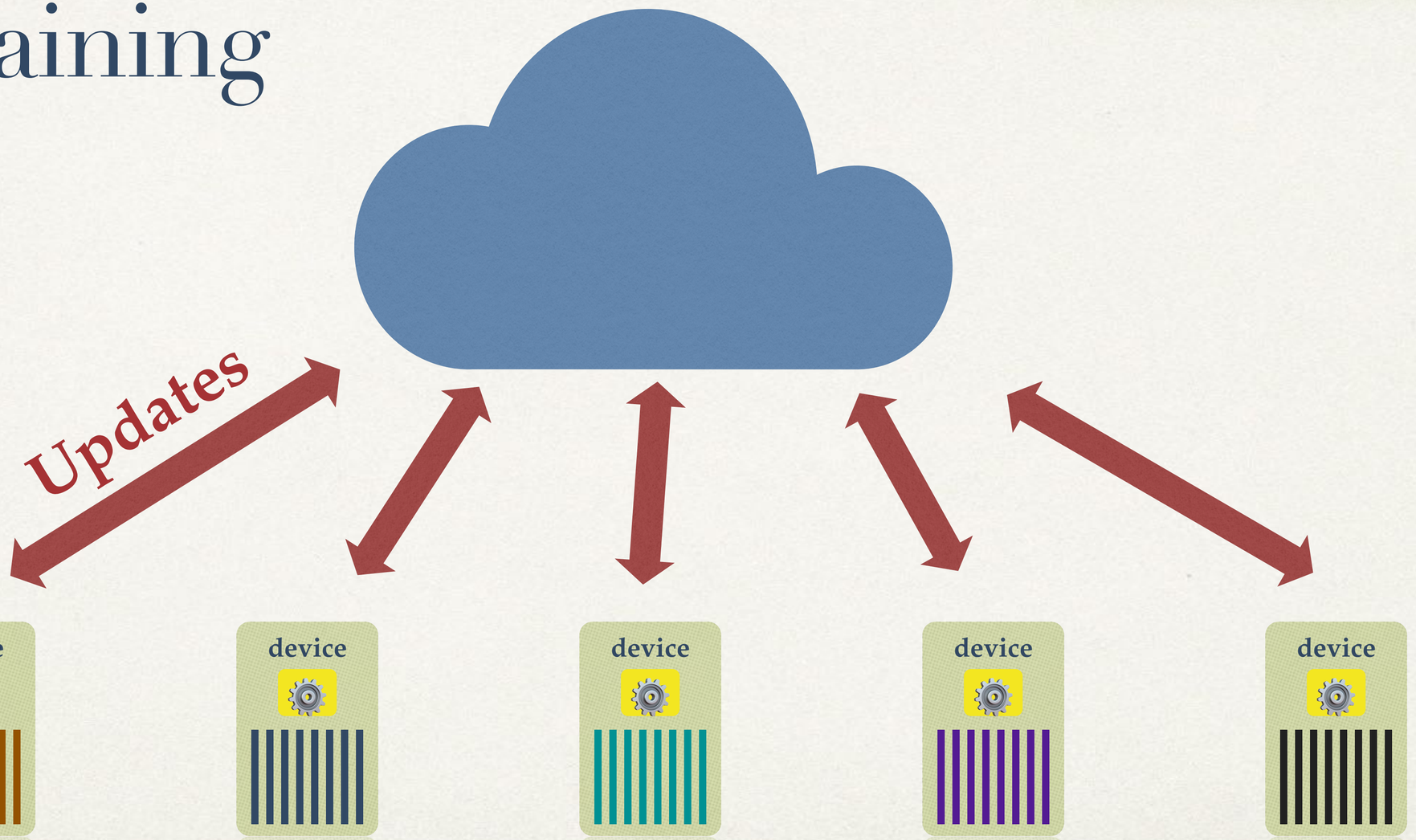
Martin Jaggi

EPFL

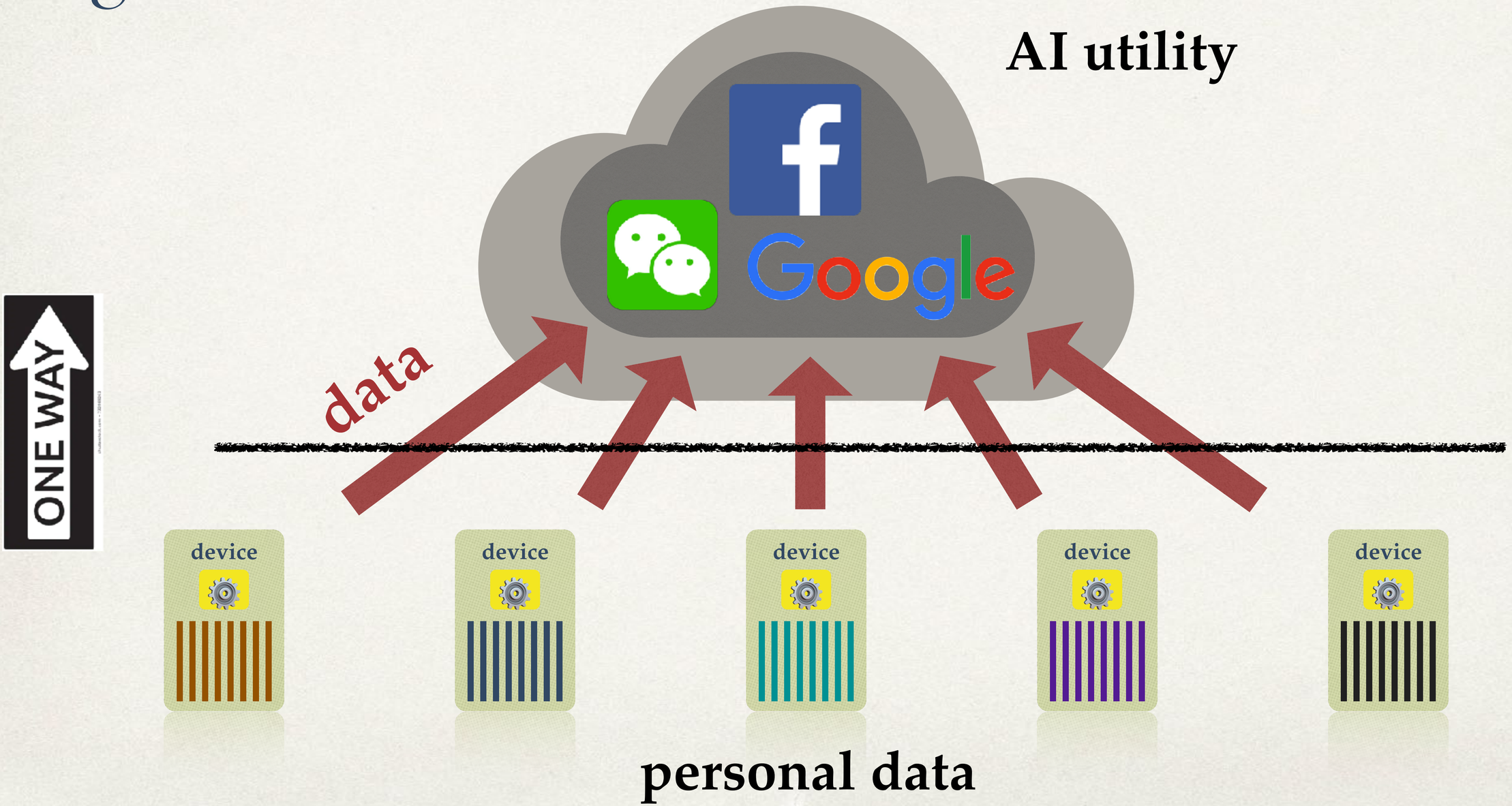
Machine Learning and Optimization Laboratory

mlo.epfl.ch

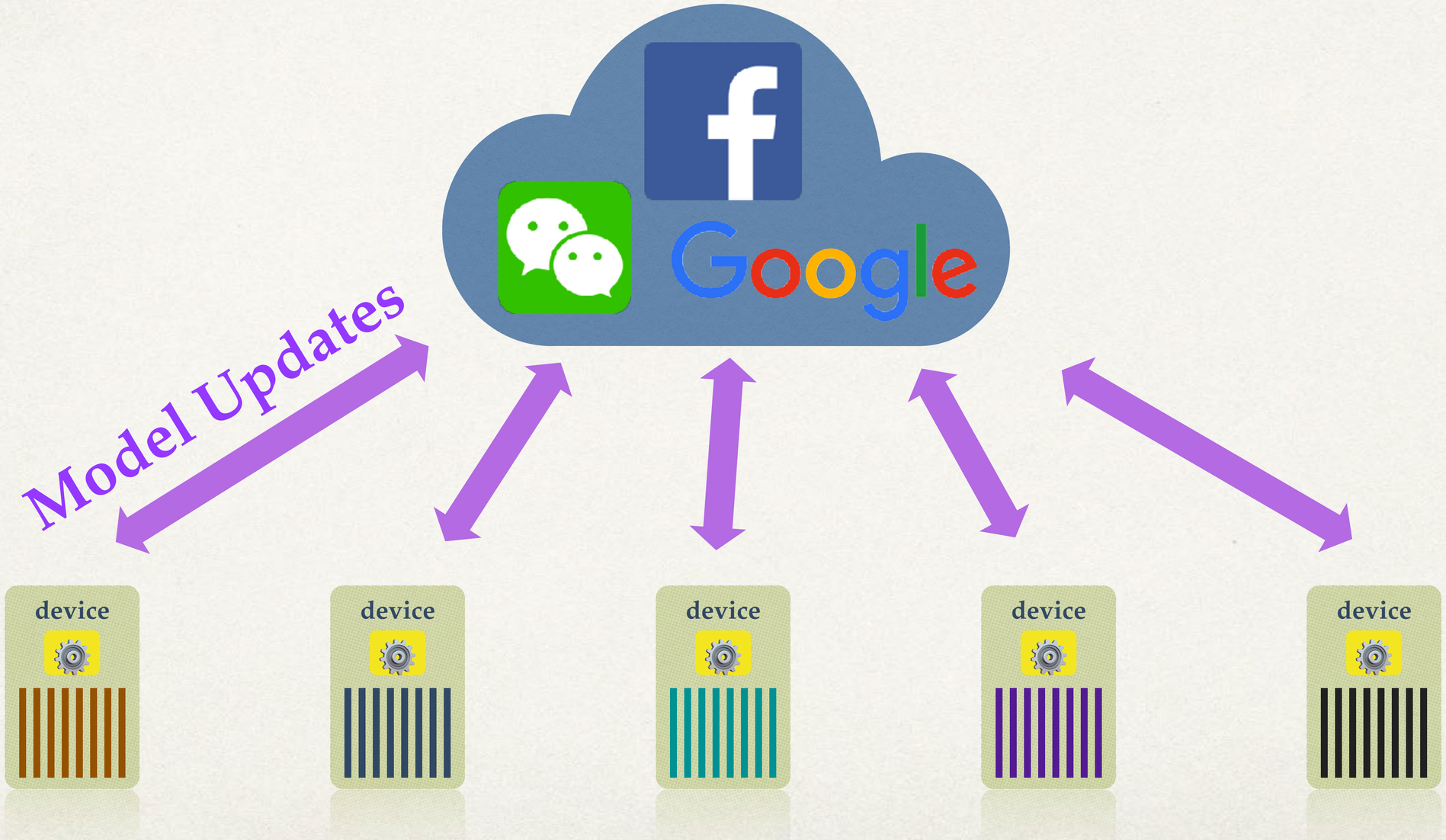
2 Collaborative Training



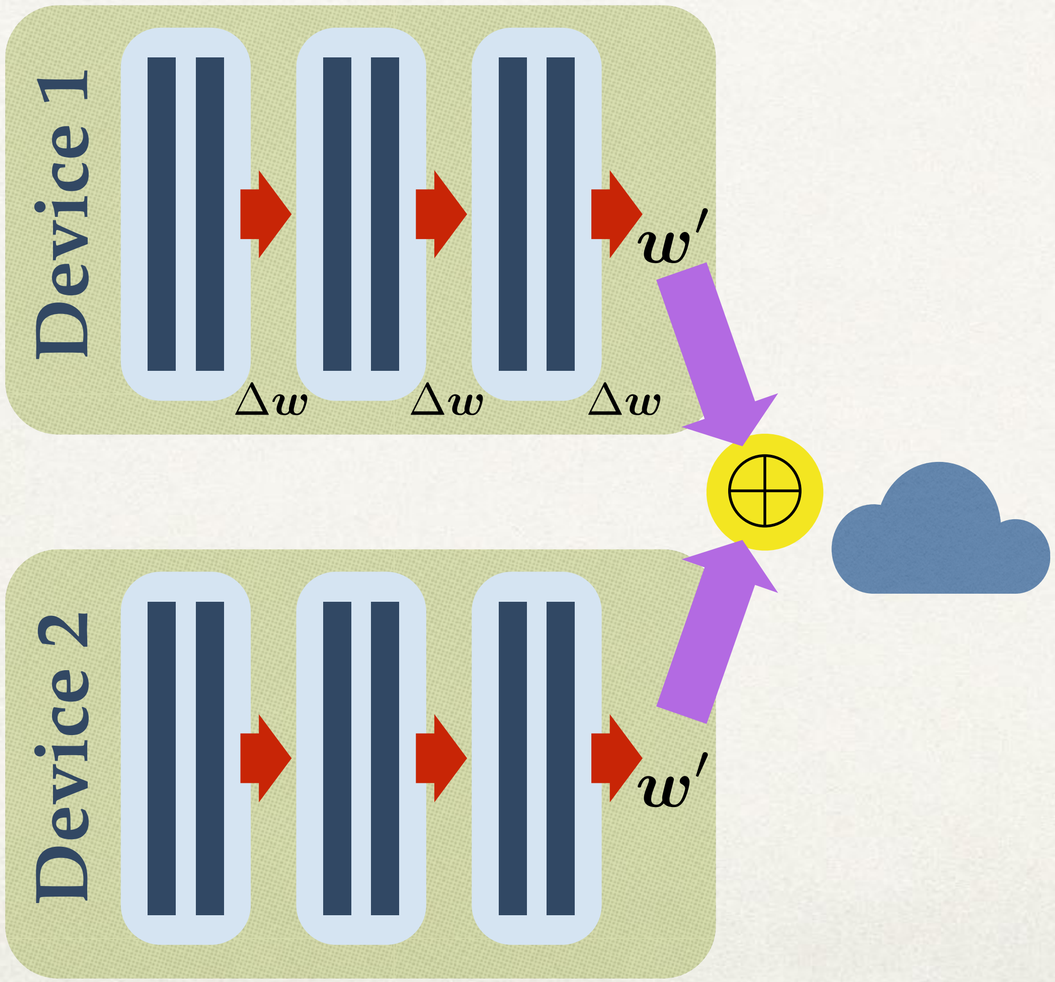
Big Picture


AI utility
Google
data
ONE WAY
device
device
device
device
device
personal data

2a Federated Learning

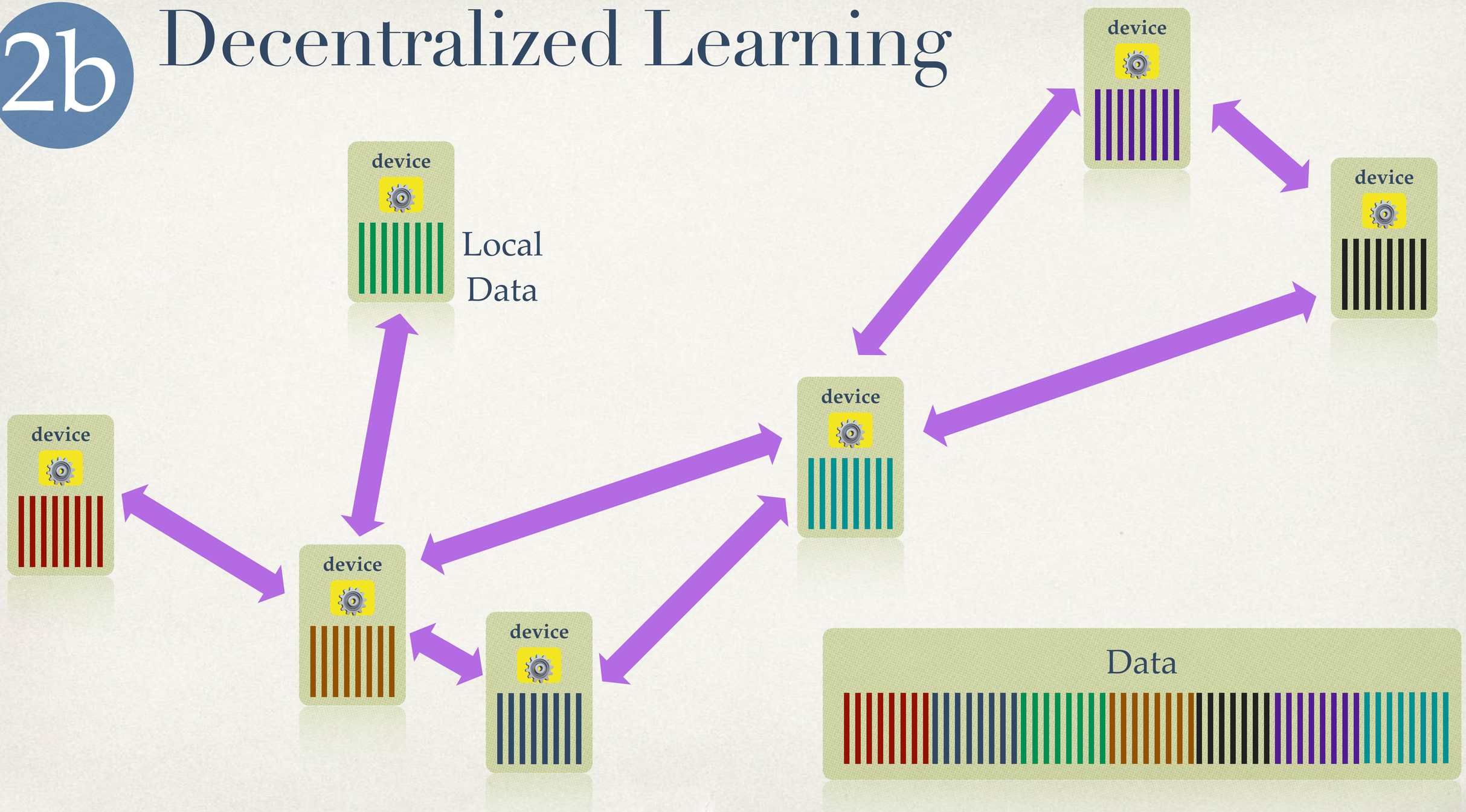


2a Federated Learning



- Local SGD steps = "Federated averaging"
- Google Android Keyboard

2b Decentralized Learning



Motivation

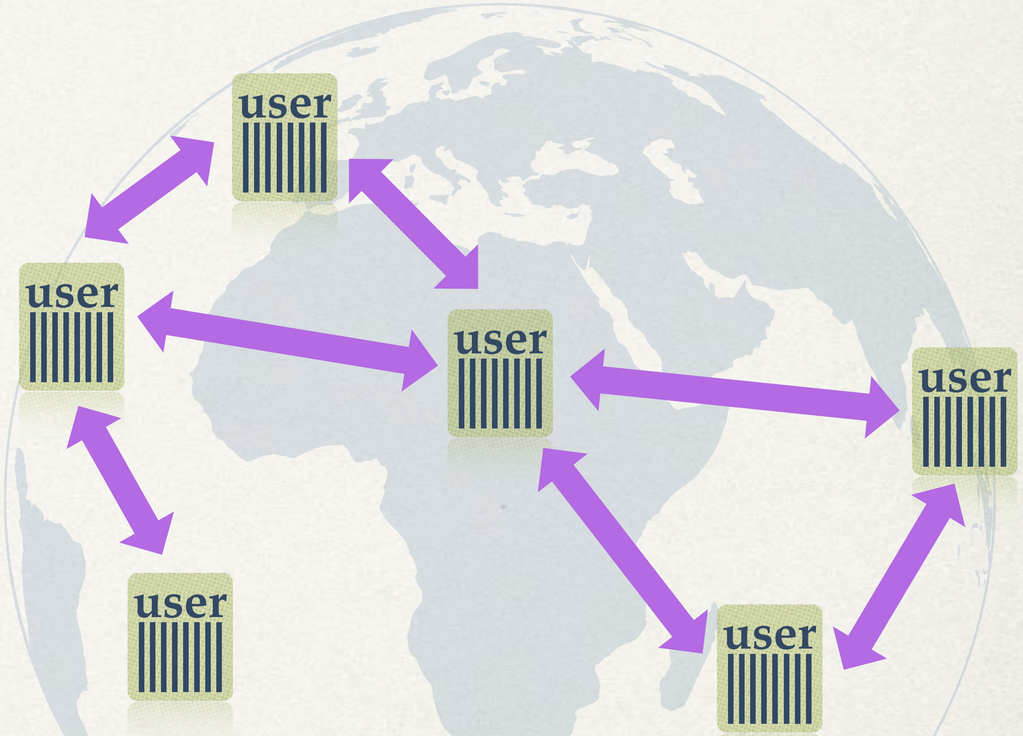
- **Applications:**

  any ML system with user data

  servers, devices, sensors, hospitals, ...

image source

- **Advantages:**



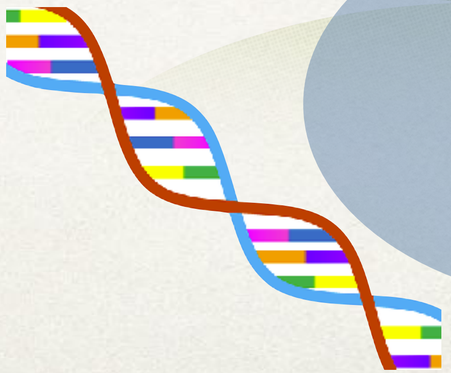
AI utility, control and privacy aligned with data ownership

Required Building Blocks

Robustness

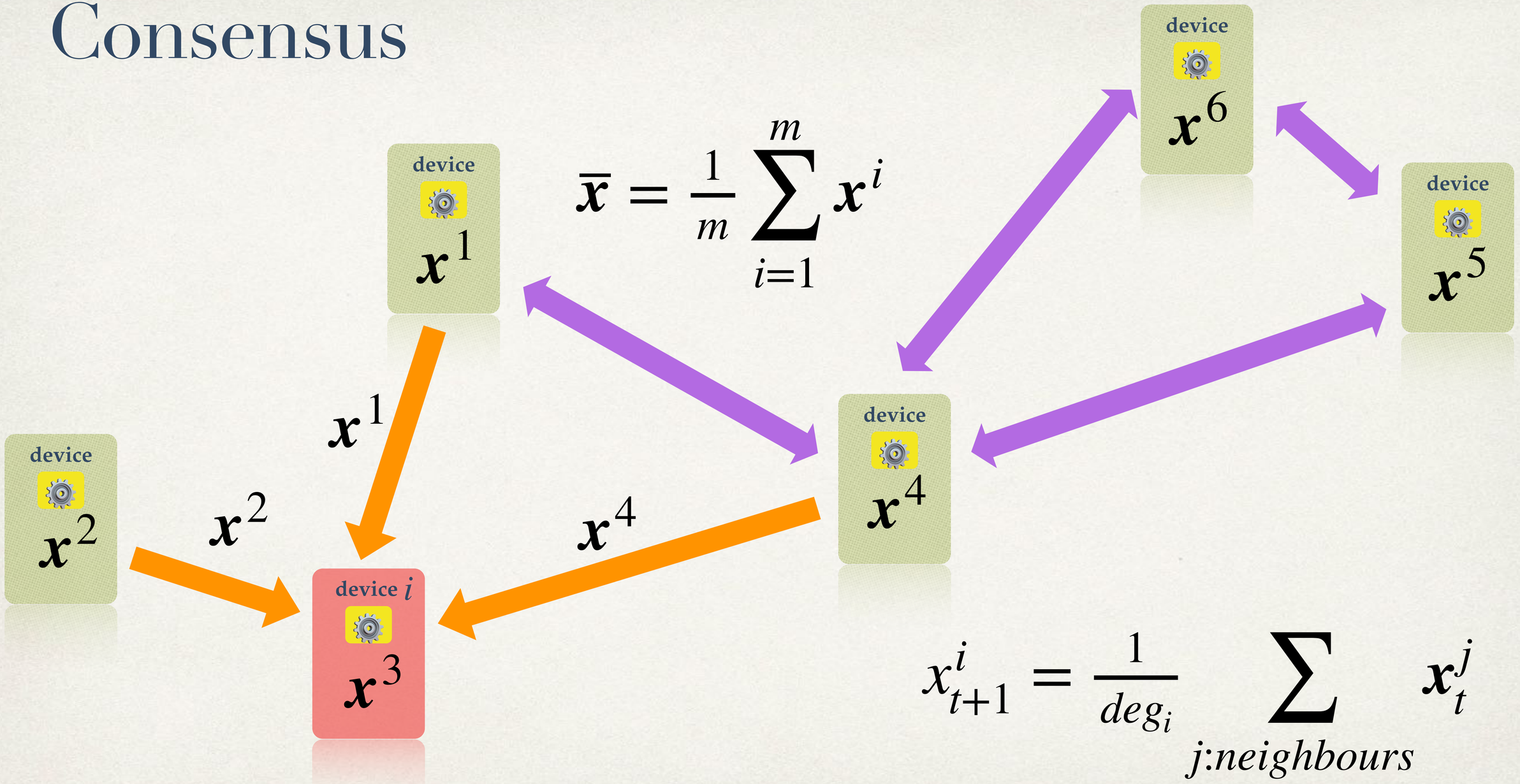
Efficiency

Decentralized ML



Privacy

Consensus



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

$$x_{t+1}^i = \frac{1}{deg_i} \sum_{j:neighbours} x_t^j$$

Communication Compression

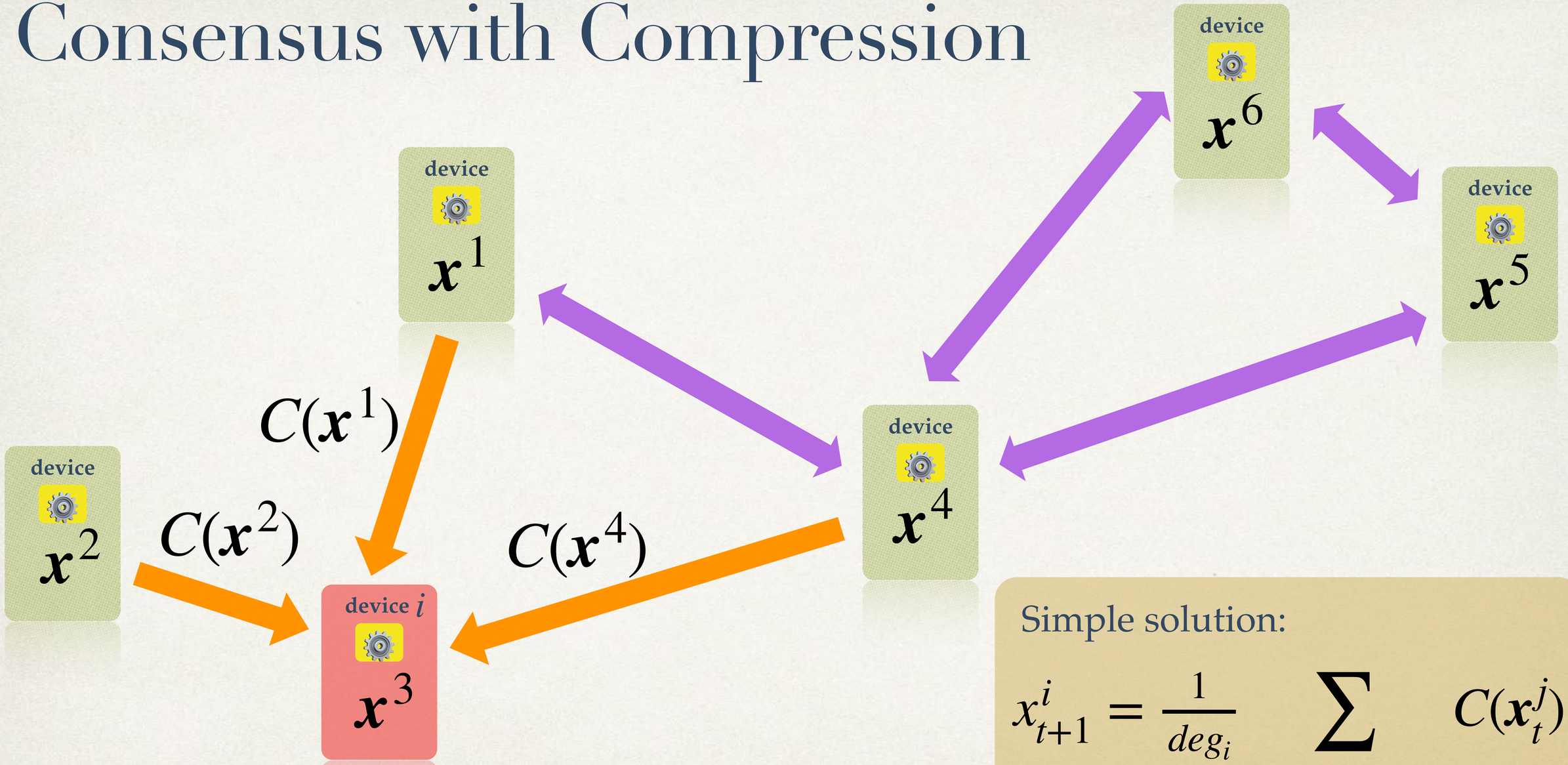
- limited-bit precision vector

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

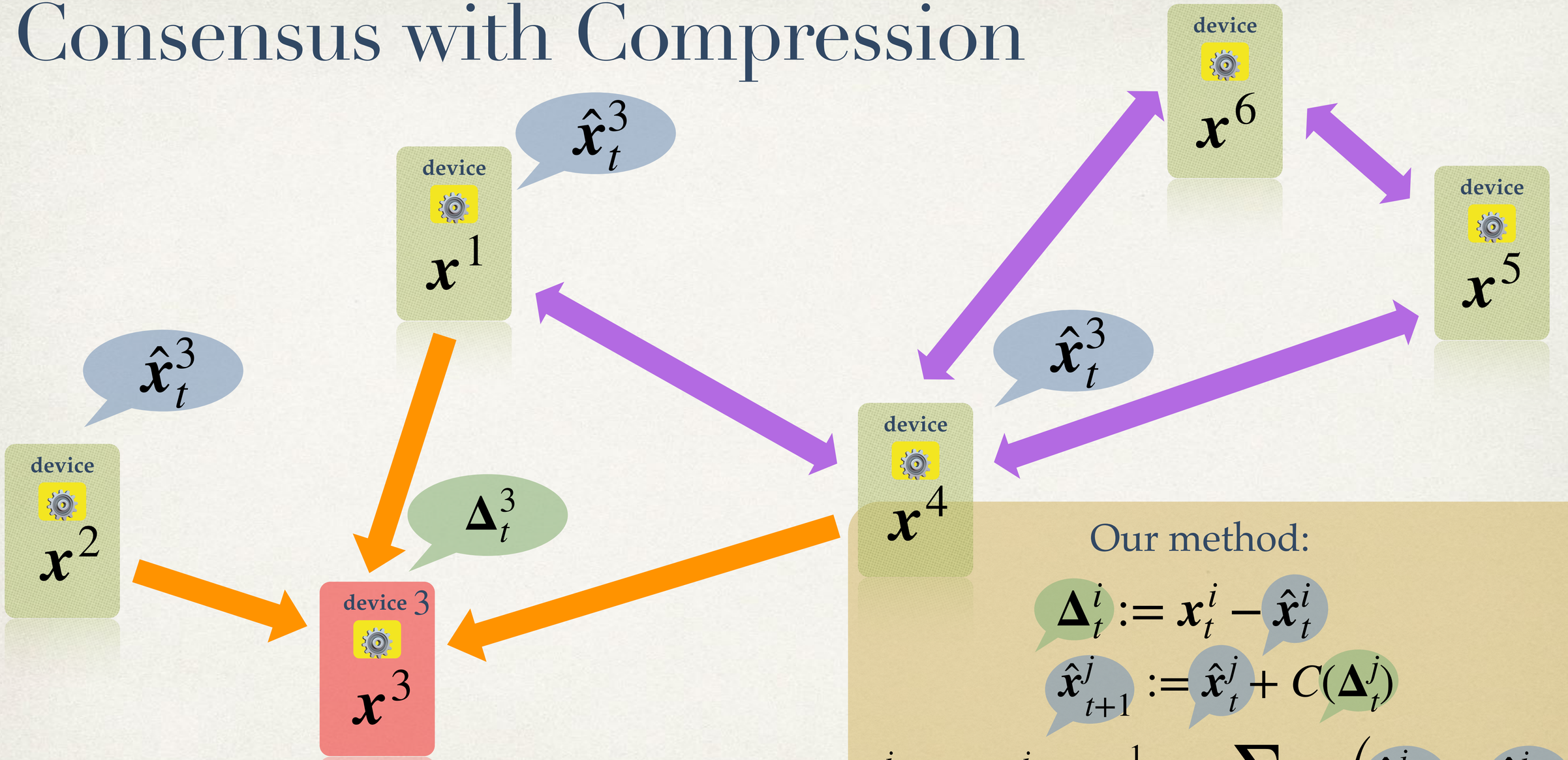
Consensus with Compression



Simple solution:

$$x_{t+1}^i = \frac{1}{deg_i} \sum_{j:neighbours} C(x_t^j)$$

Consensus with Compression



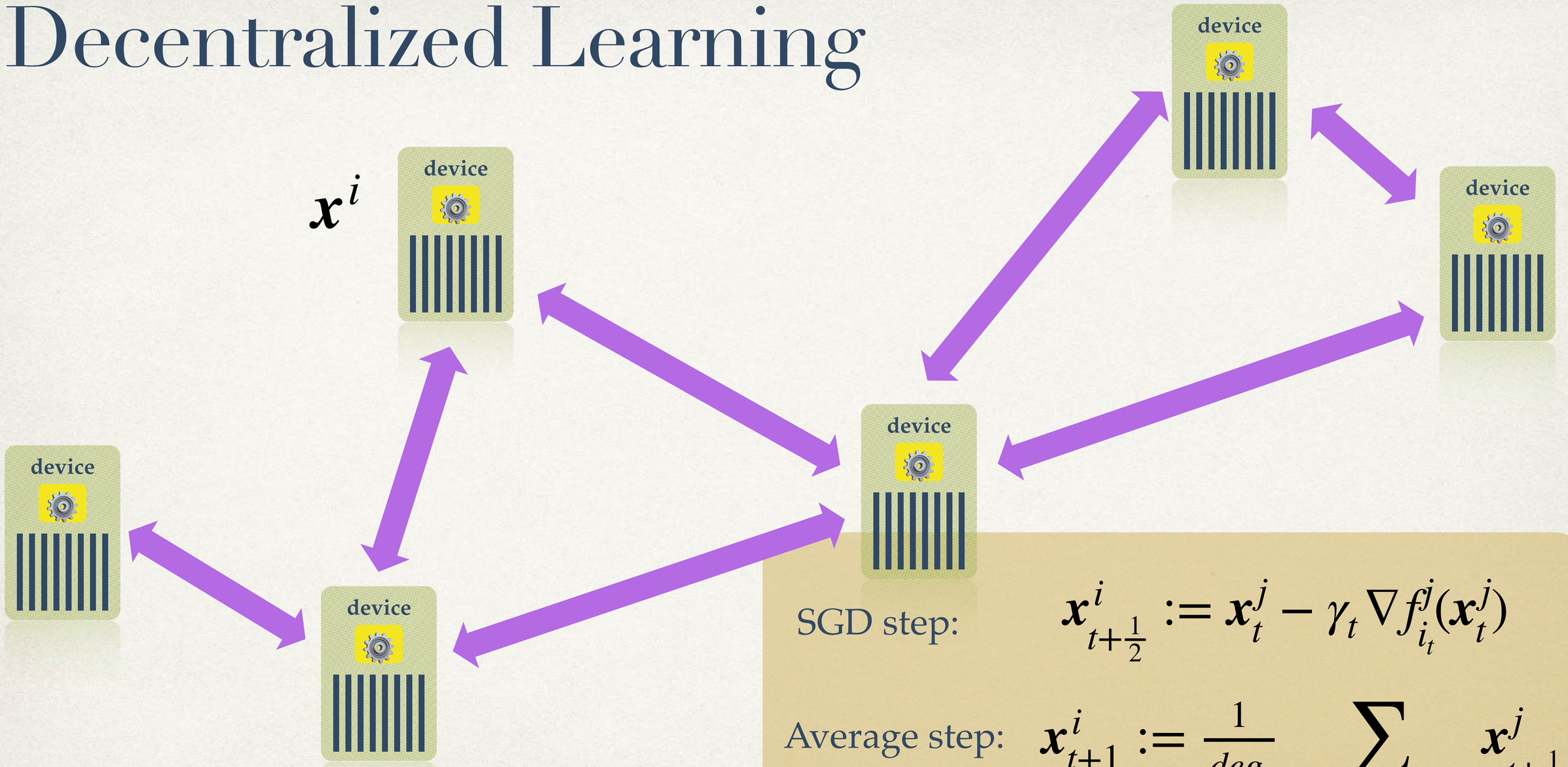
Our method:

$$\Delta_t^i := x_t^i - \hat{x}_t^i$$

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

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

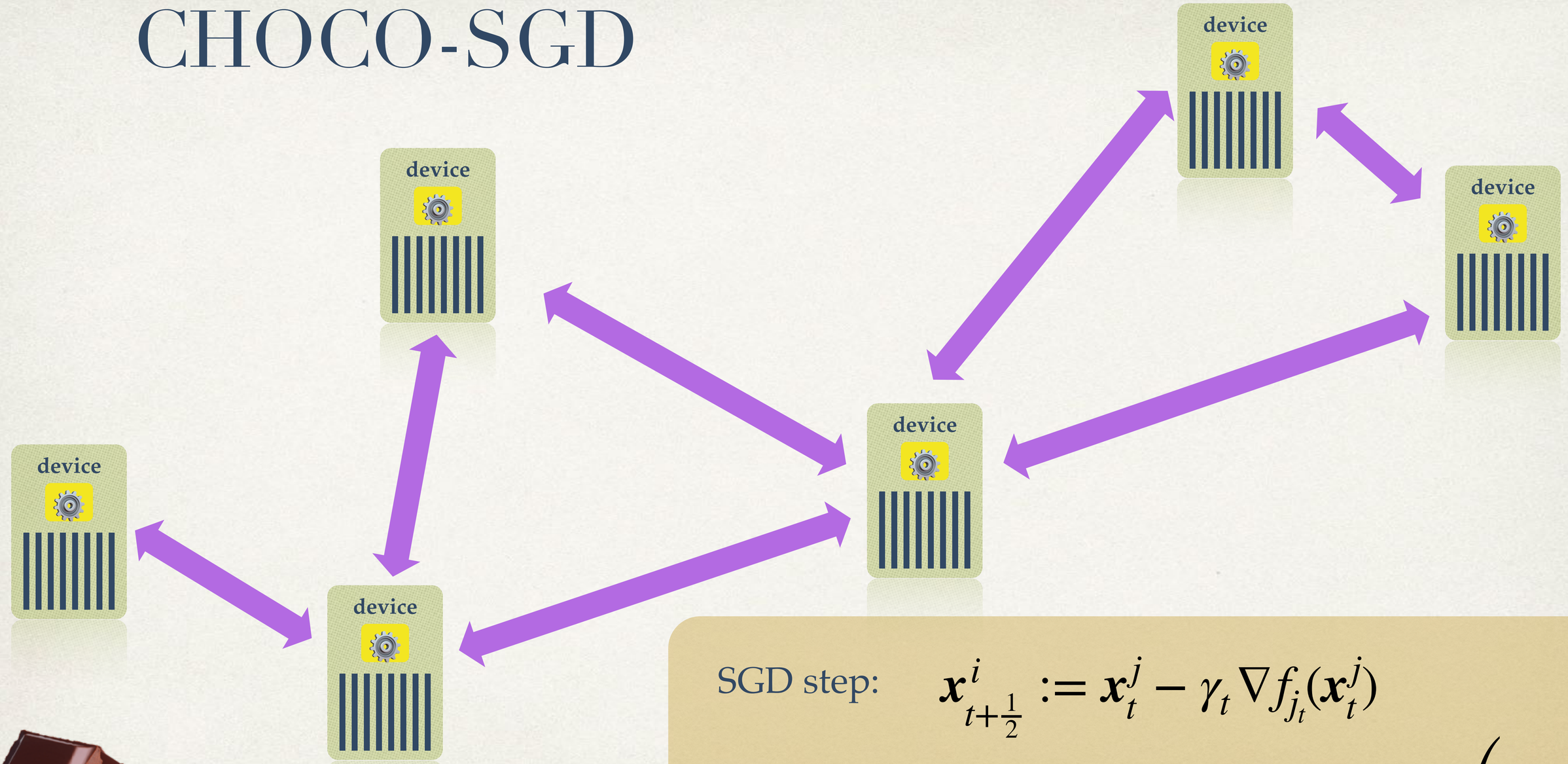
Decentralized Learning



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

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

CHOCO-SGD



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

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

Convergence (Non-Convex Case)

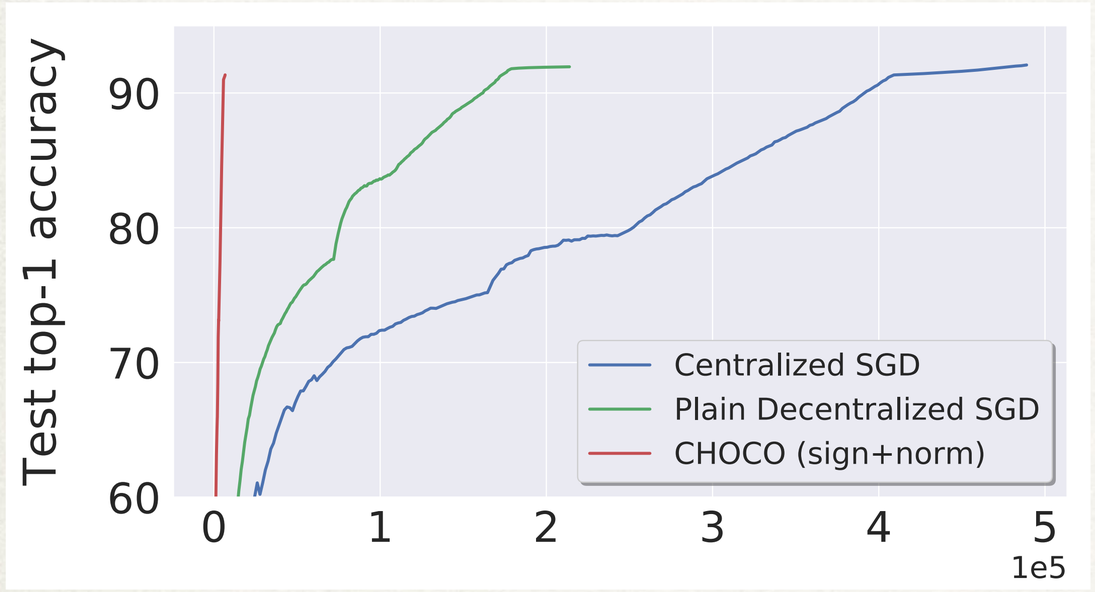
$$\frac{1}{T+1}\sum_{t=0}^{T}\|\nabla f(\bar{x}_t)\|^2 = \mathcal{O}\left(\frac{1}{\sqrt{nT}} + \frac{n}{\delta^2\rho^4 T}\right)$$

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

ρ — spectral gap of the graph topology

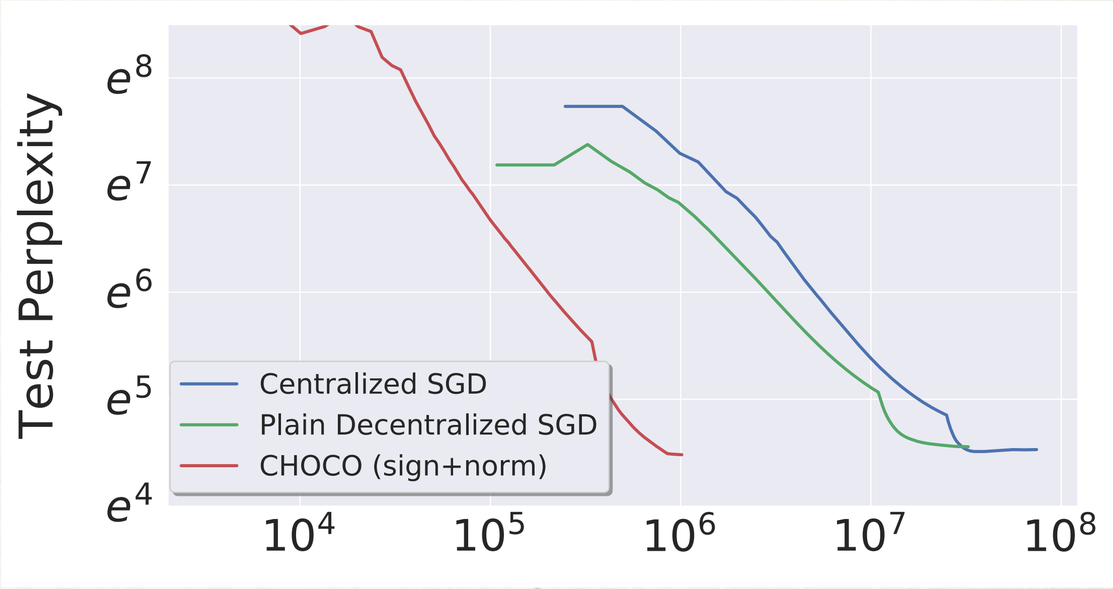
- linear speedup in the number of workers

Decentralized DL



data transmitted (MB)

Resnet20 on Cifar 10

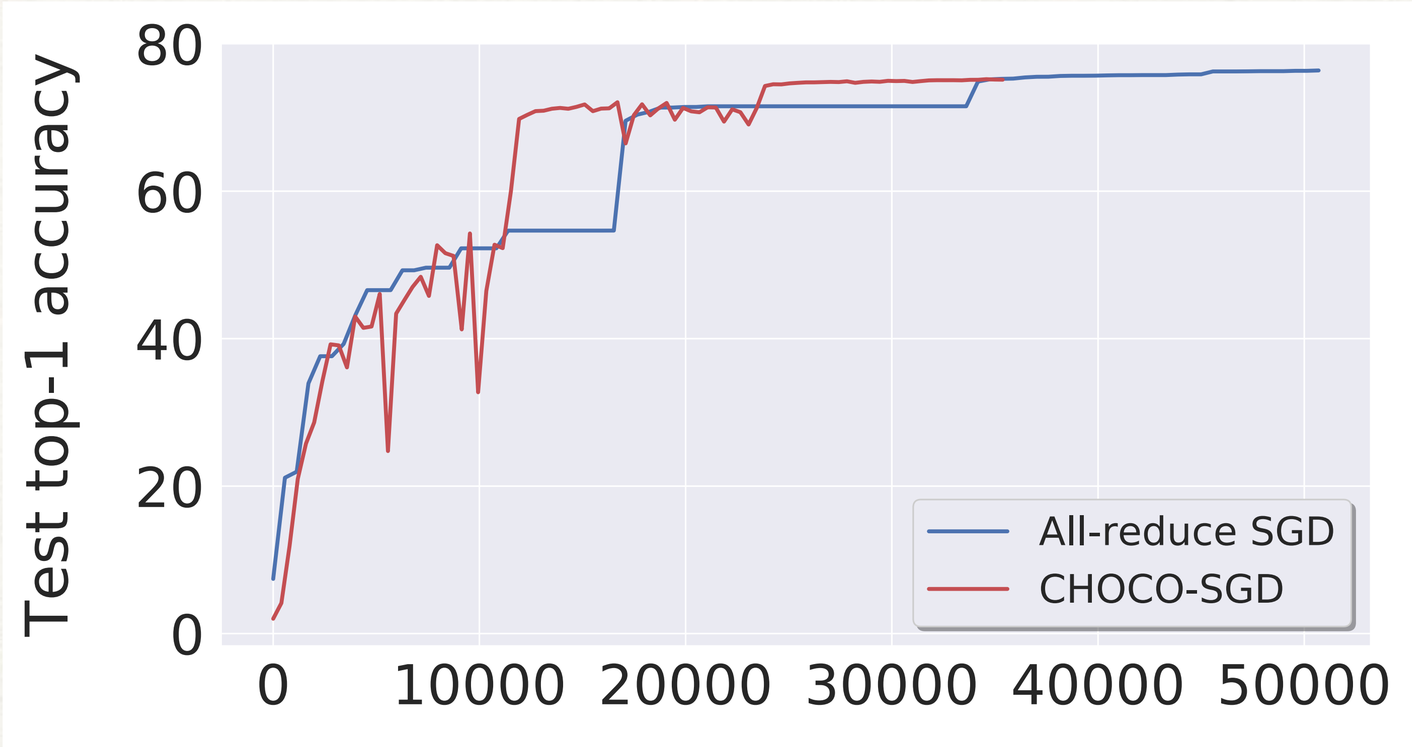


data transmitted (MB)

Language model (3-layer LSTM) on WikiText-2

Social Network Topology, 32 nodes of max deg 14
Sign quantization

DL in Datacenter



time

Resnet50 on ImageNet-1k
Ring of 8 nodes, each has 4 P100 GPUs

Conclusions - Choco

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

Building Blocks for Decentralized ML

- **Efficiency: Communication & Compute**

  on-device learning, Edge AI

  peer-to-peer communication

- **Privacy**

  data locality, leakage?, attacks?

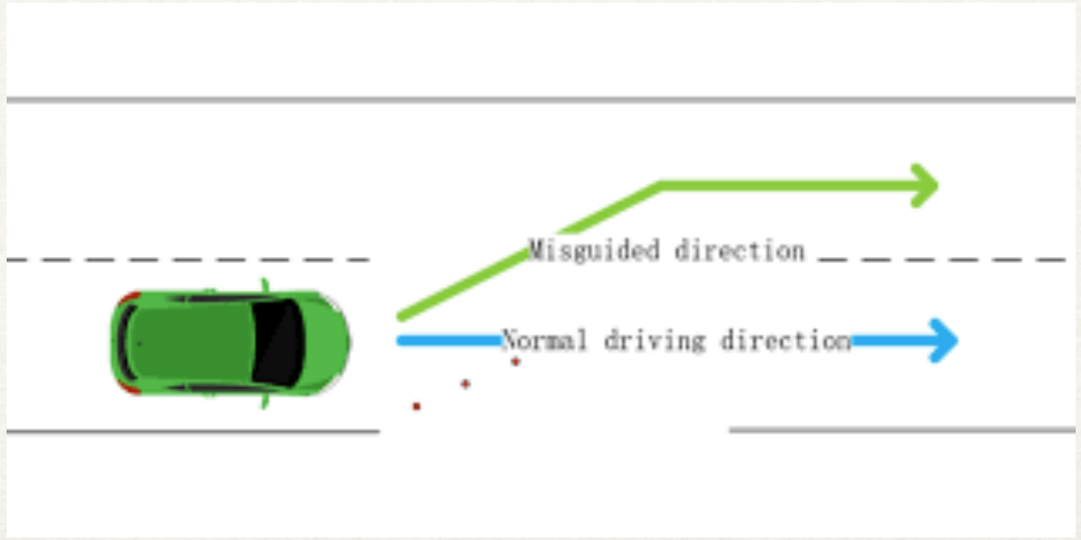
- **Robustness & Incentives**

  tolerate bad players, reward collaboration

3

Robustness

During Training and Inference



Byzantine-robust training

- **Mean vs median**

Zeno: Byzantine-suspicious stochastic gradient descent, 2018, Cong Xie et al.

Adversarial Attacks (at inference time)

Image: [Tom B. Brown/Dandelion Mané]

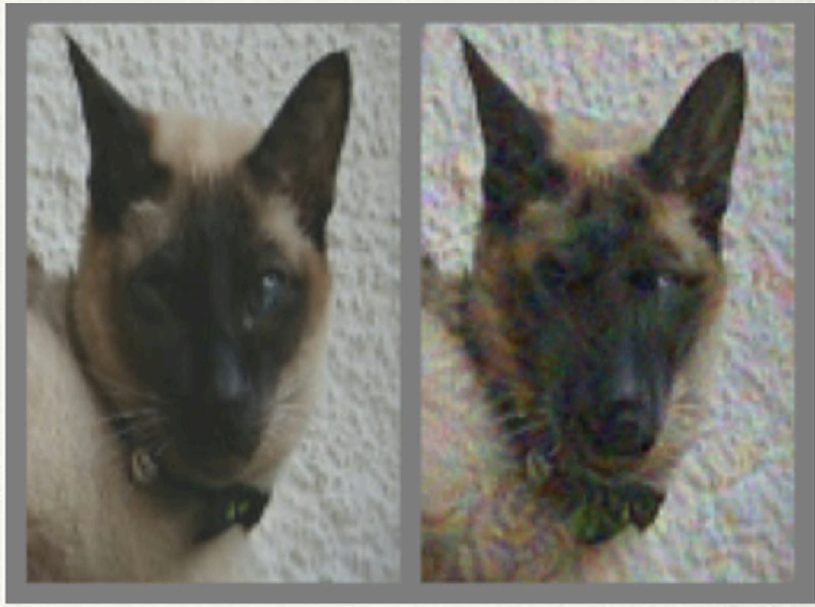


Image: Elsayed ,Papernot et al 2018

Adversarial Attacks (at inference time)

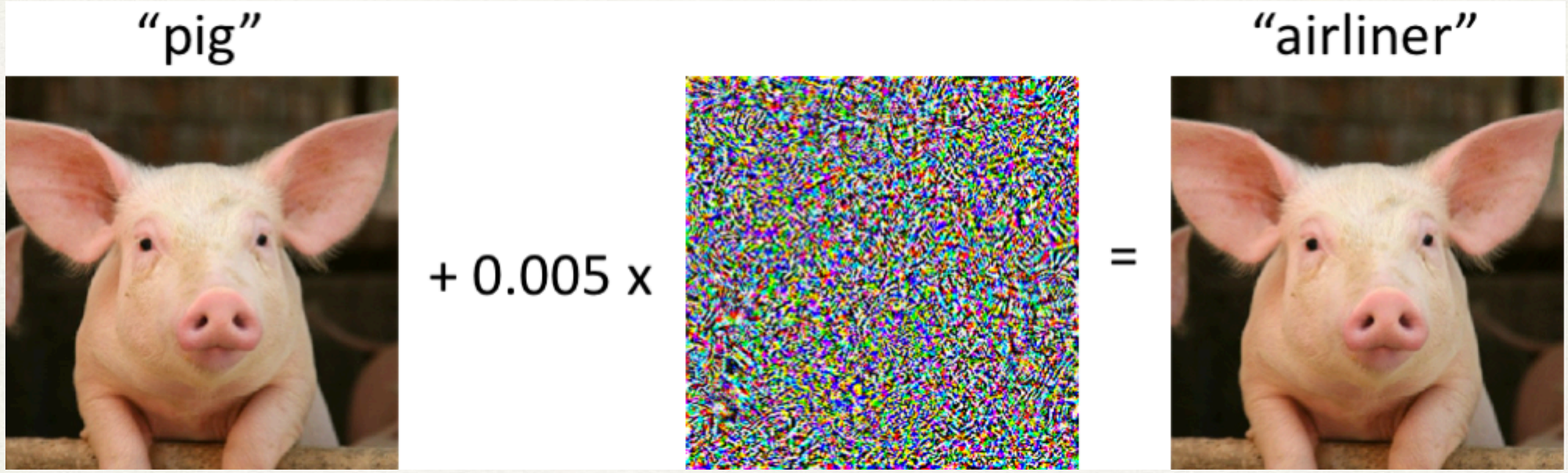


Image: Mądry, Schmidt

More info:
http://gradientscience.org/intro_adversarial/

Adversarial Attacks

- Standard **training**

$$\min_{\mathbf{w}} \quad f_{\mathbf{w}}(\mathbf{x}_i)$$

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

change **model**

- **Attacking**

$$\max_{\mathbf{x} \in R_\infty(\mathbf{x}_i, \varepsilon)} \quad f_{\mathbf{w}}(\mathbf{x}_i)$$

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

change **data**

- by **Projected Gradient Descent!**

4

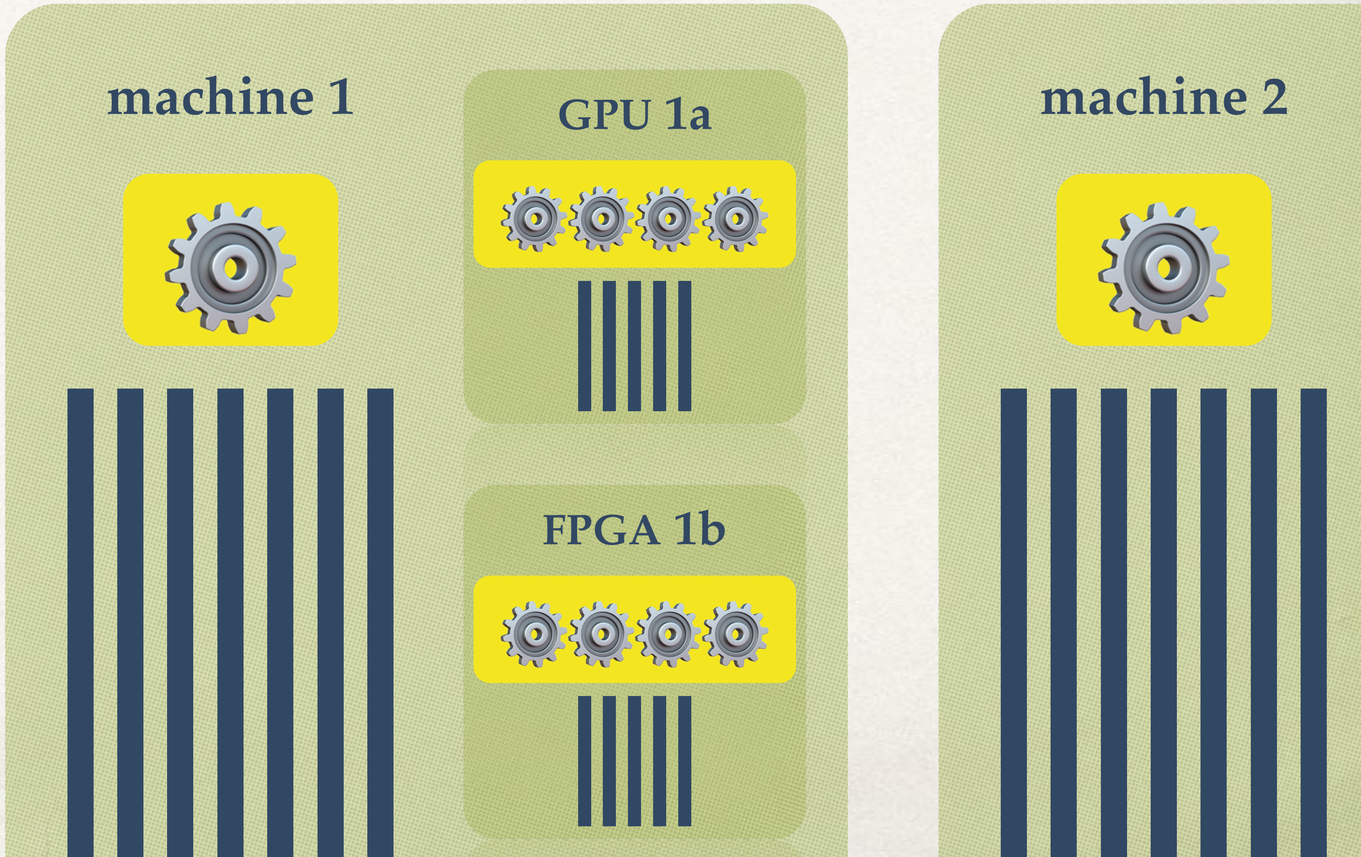
Privacy

- Secure Multiparty Computation
- Differential Privacy
- Privacy / inference Attacks

5

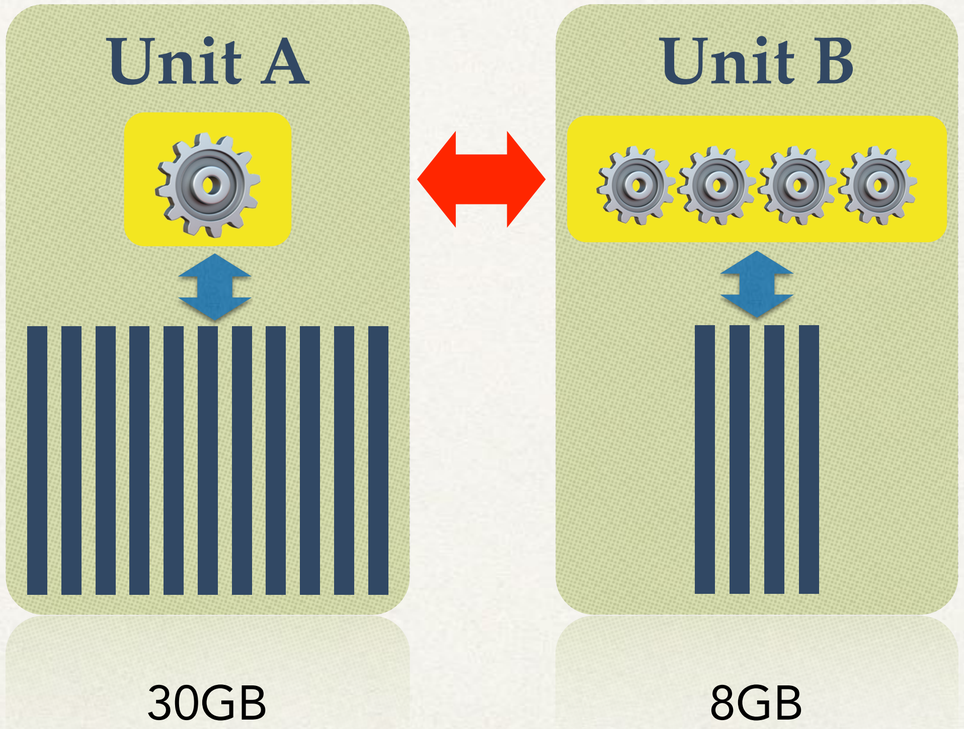
Leveraging Heterogenous Systems

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



Leveraging Heterogenous Systems

duality gap as selection criterion

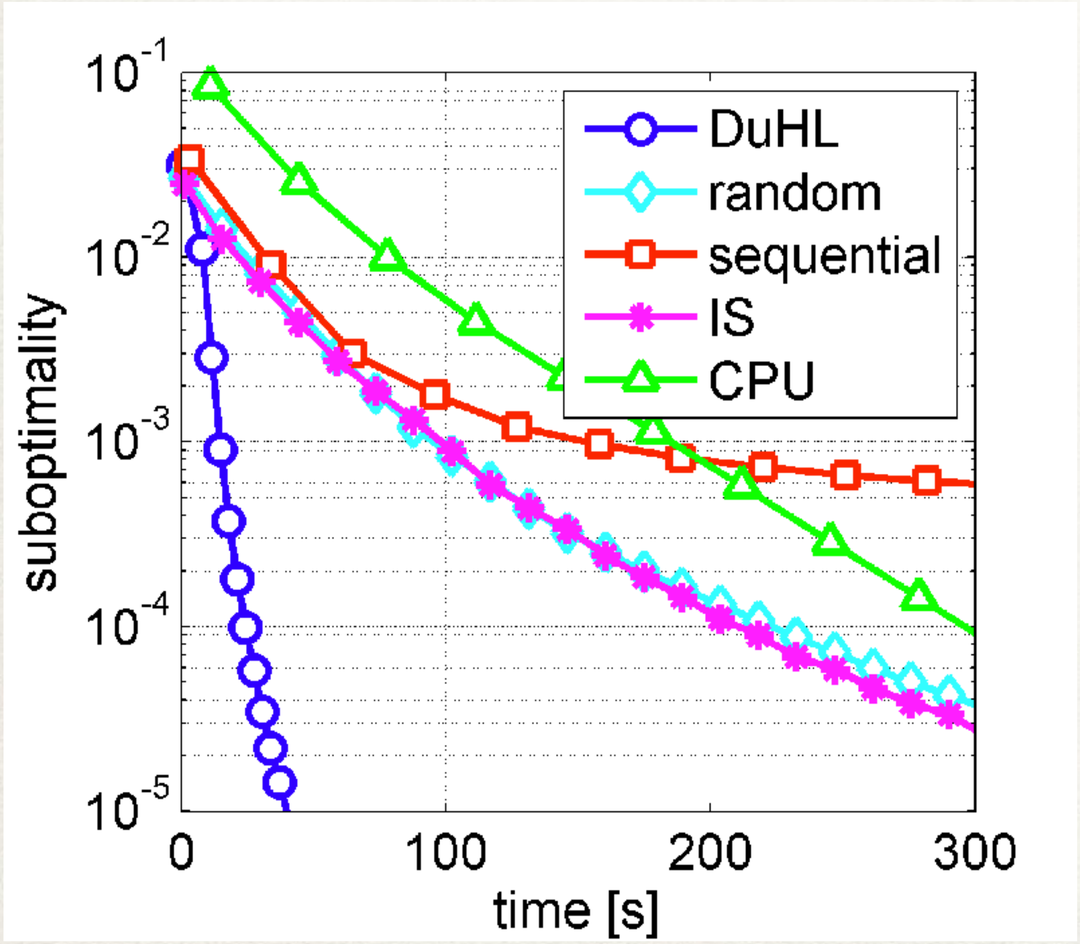


adaptive importance sampling

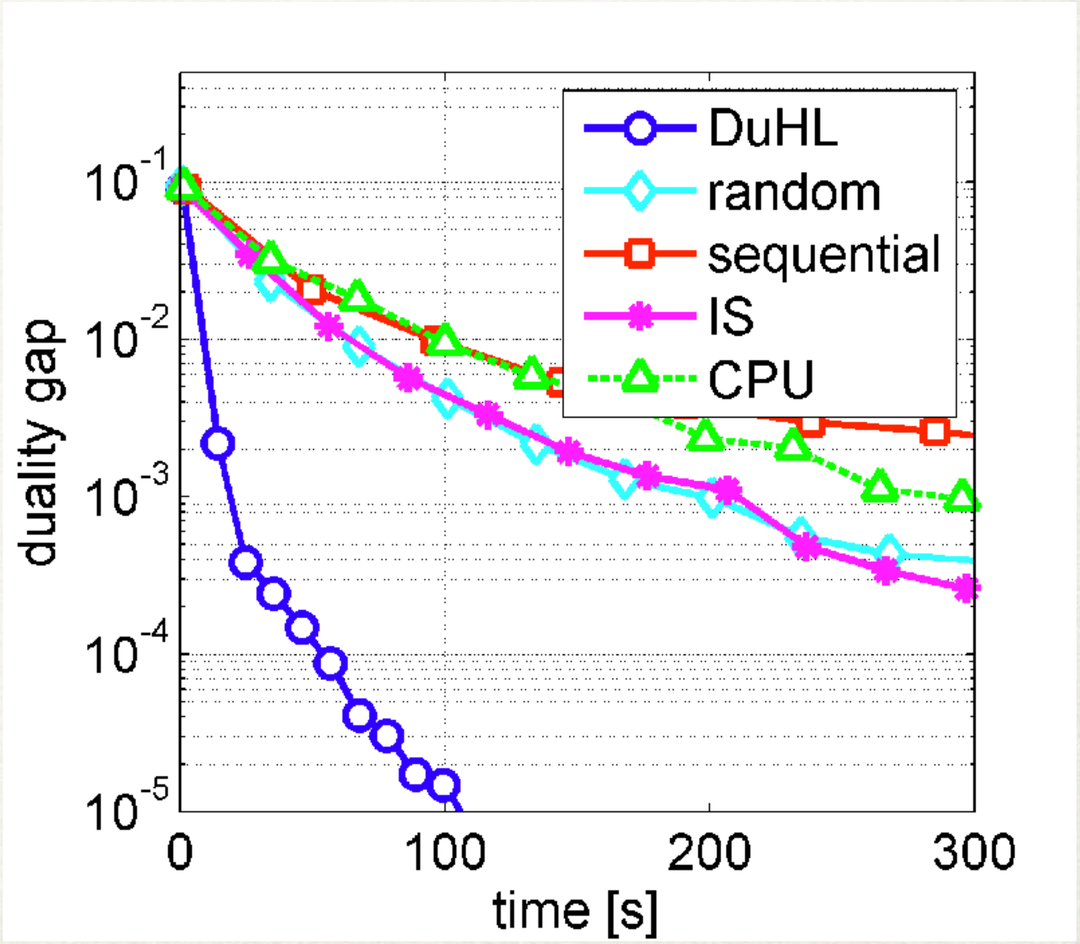
AISTATS 2017, 2018
NIPS 2017a,b

Experiments

RAM ⟷ GPU, 30GB dataset

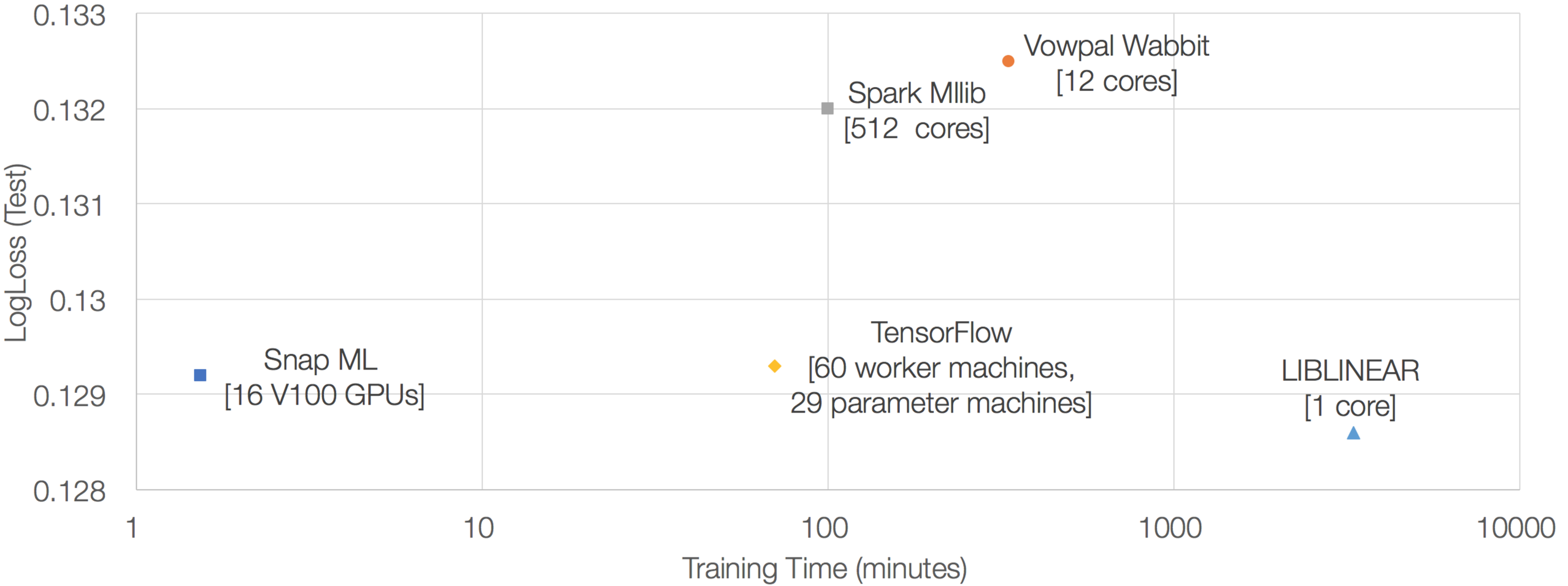


Lasso



SVM

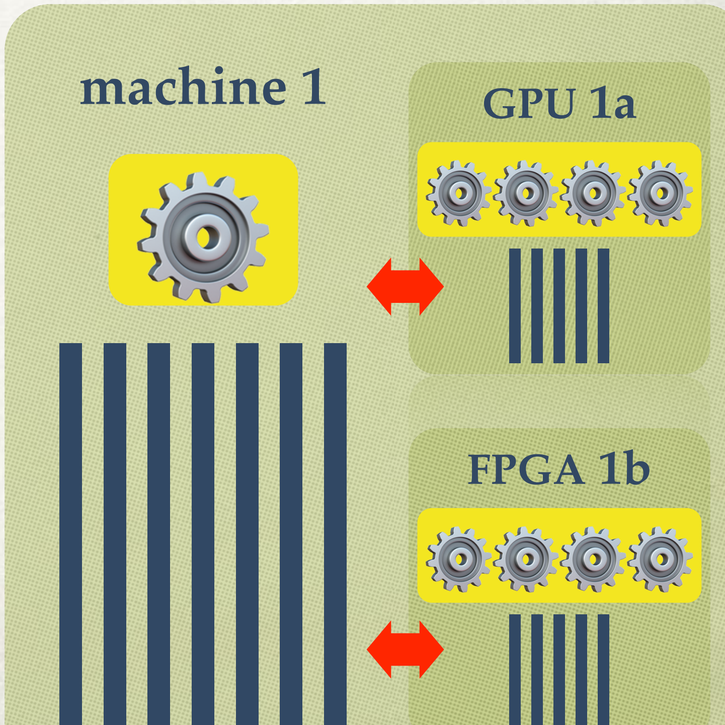
Experiments



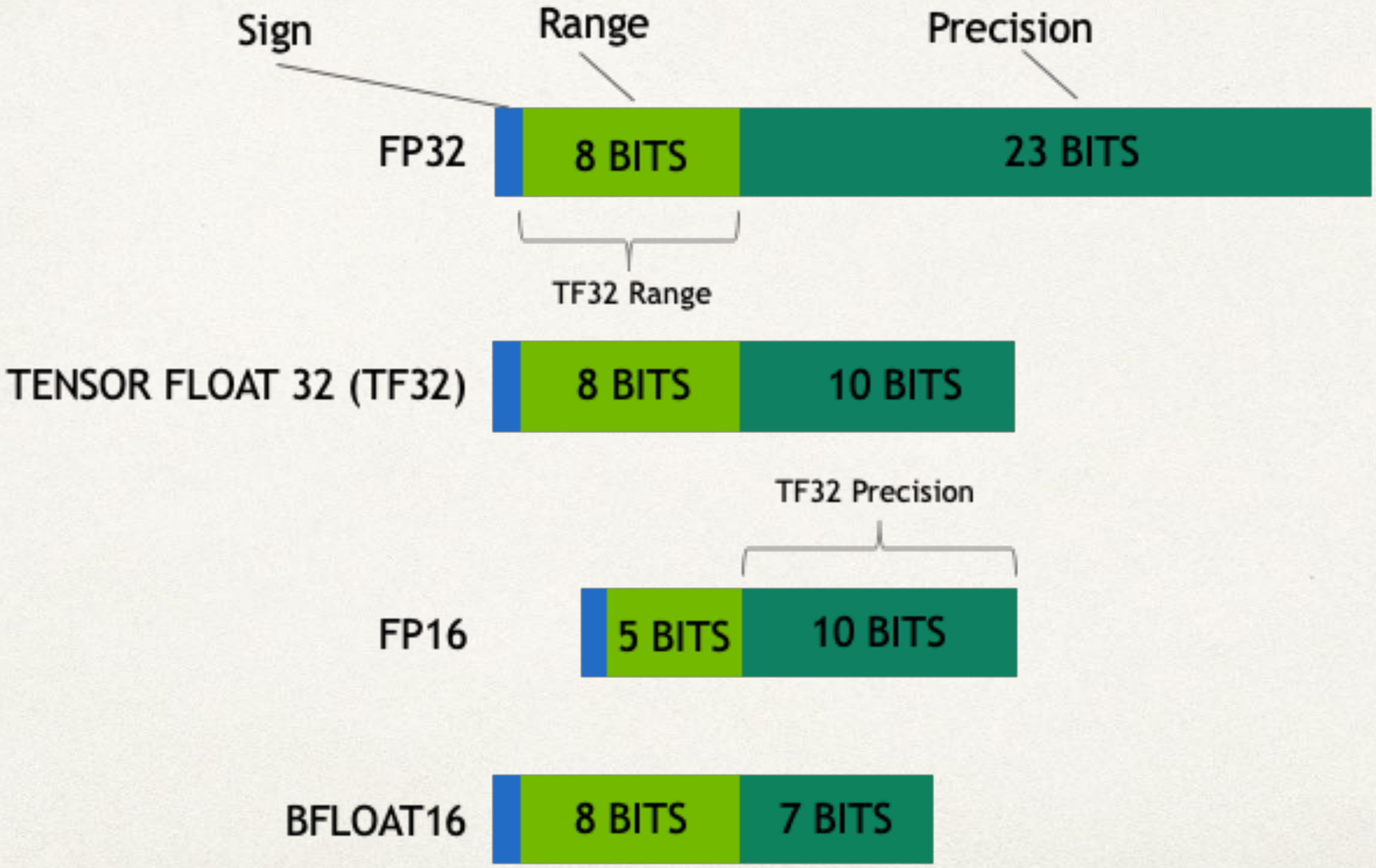
terabyte click log dataset, IBM cloud implementation *[arXiv]*

Trends - Systems

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



Number formats for DL



Open Source Project:

MLbench - Distributed Machine Learning Benchmark

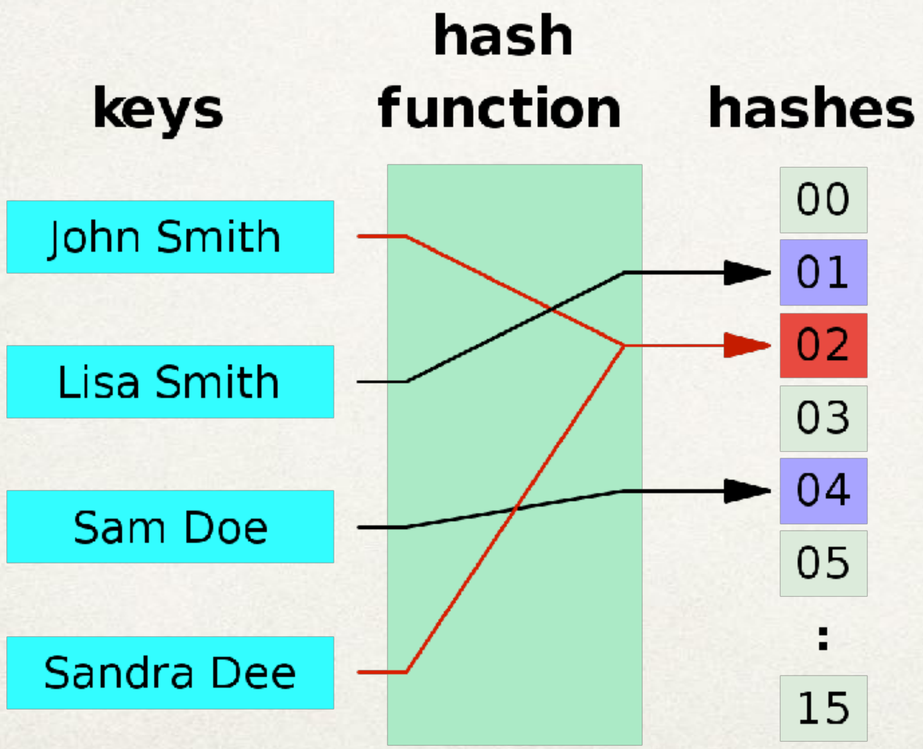
Public and reproducible reference implementations and benchmarks for distributed machine learning **algorithms**, **frameworks** and **systems**.

mlbench.github.io


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
Cloud ML
elasticity
colab
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
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...)

Practical tricks

- **feature hashing**



- **limited precision operations**

Auto ML

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

Thanks!

mlo.epfl.ch
tml.epfl.ch