Mobile Edge Artificial Intelligence: Opportunities and Challenges

Motivations

Yuanming Shi

ShanghaiTech University

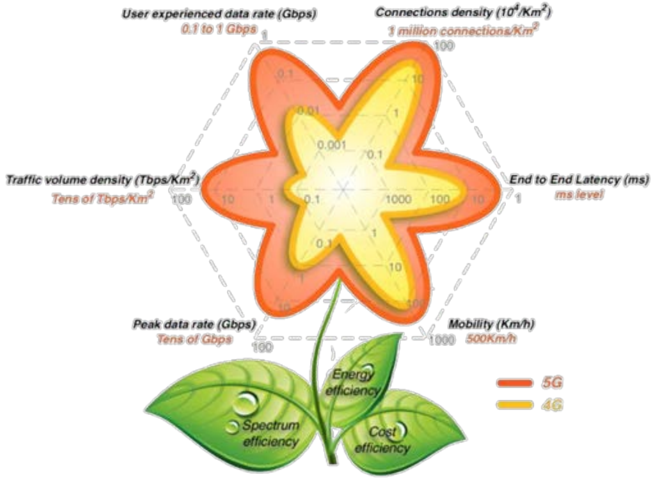
上海科技大学
ShanghaiTech University

Why 6G?

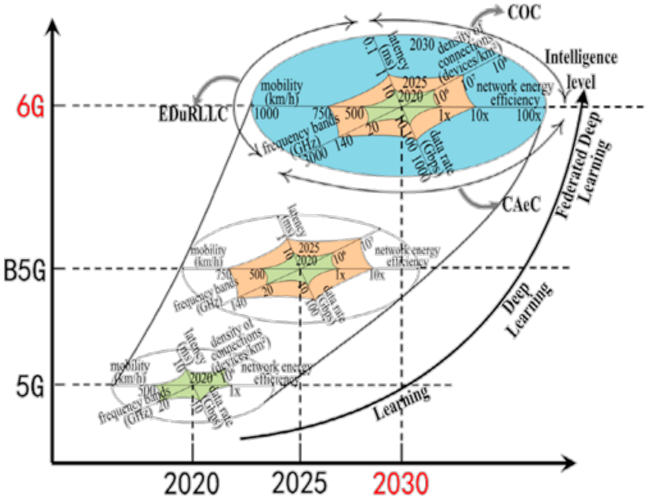
Fig. credit: Walid

What will 6G be?

- **6G networks:** from “connected things” to “connected intelligence”



5G: connected things

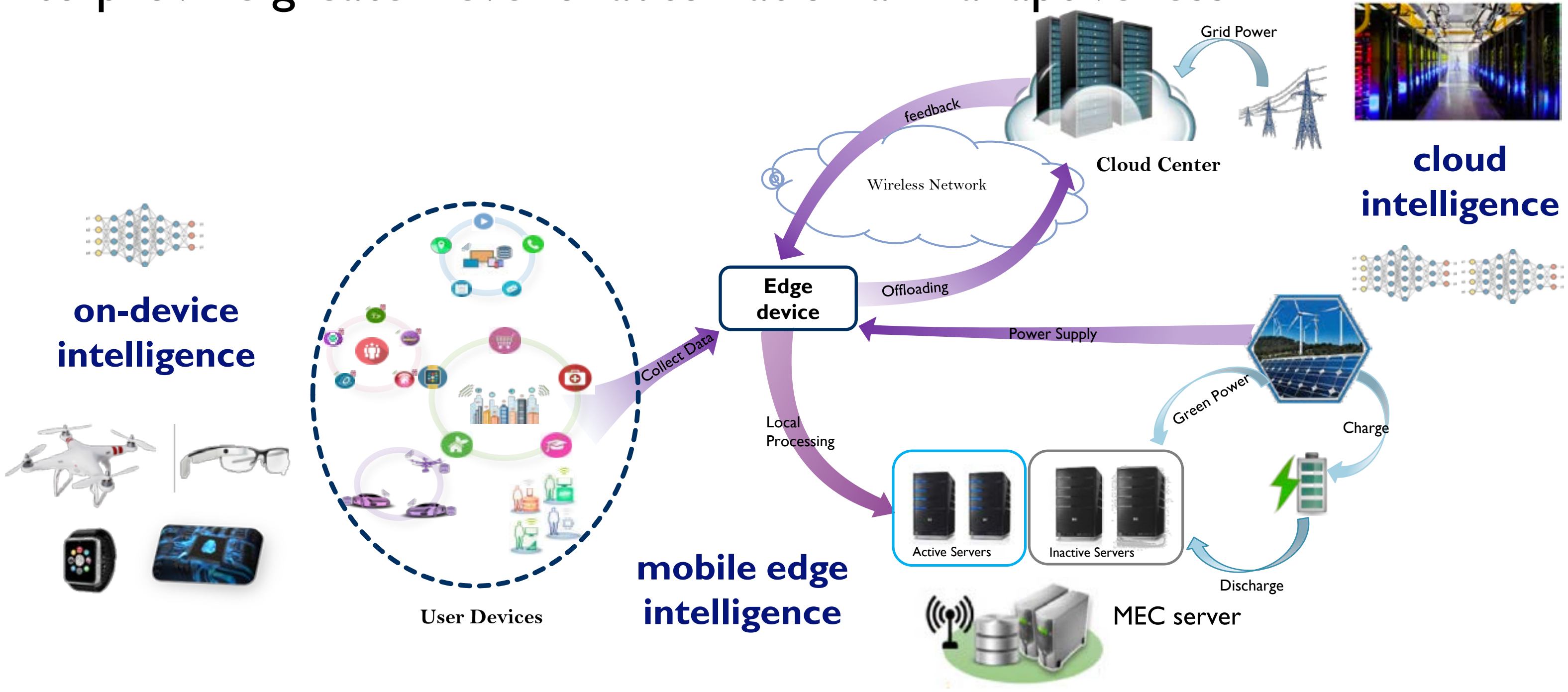


6G: connected intelligence

[Ref] K. B. Letaief, W. Chen, **Y. Shi**, J. Zhang, and Y. Zhang, “The roadmap to 6G - AI empowered wireless networks,” *IEEE Commun. Mag.*, vol. 57, no. 8, pp. 84-90, Aug. 2019.

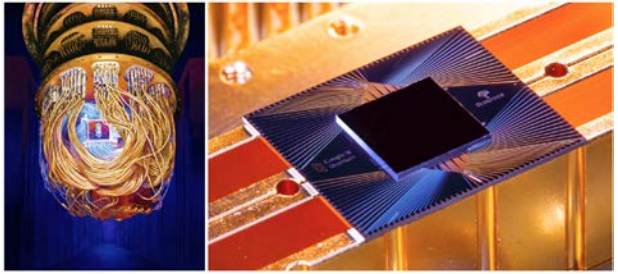
Connected intelligence via AI

- **Make networks full of AI:** embed intelligence across whole network to provide greater level of automation and adaptiveness

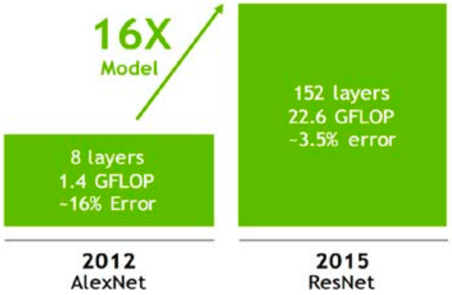


Success of modern AI

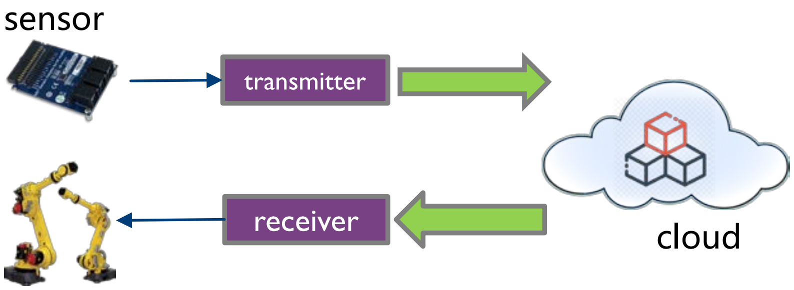
- Two secrets of AI's success: computing power and big data
  - ➢ Computing power: Intel i386, Intel i486, Intel Pentium Intel Core, Nvidia GPU, Google TPU, *Google quantum supremacy*,…
  - ➢ Big data: the world's most valuable resource is no longer oil, but data



Challenges of modern AI



model size



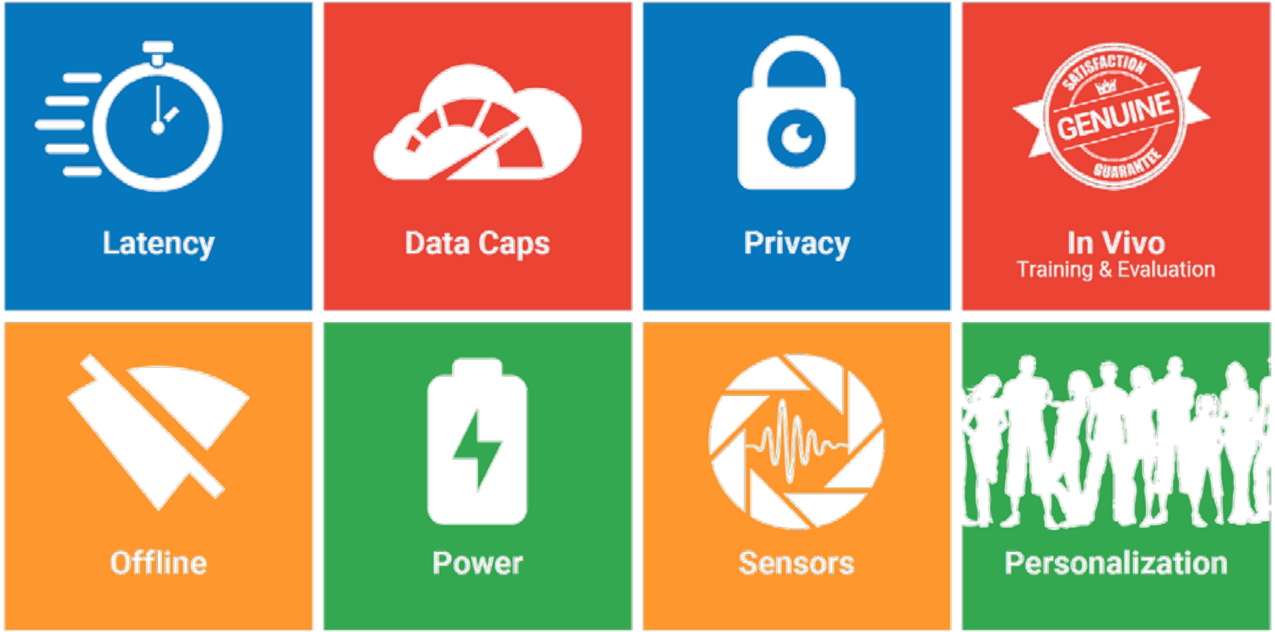
speed

energy

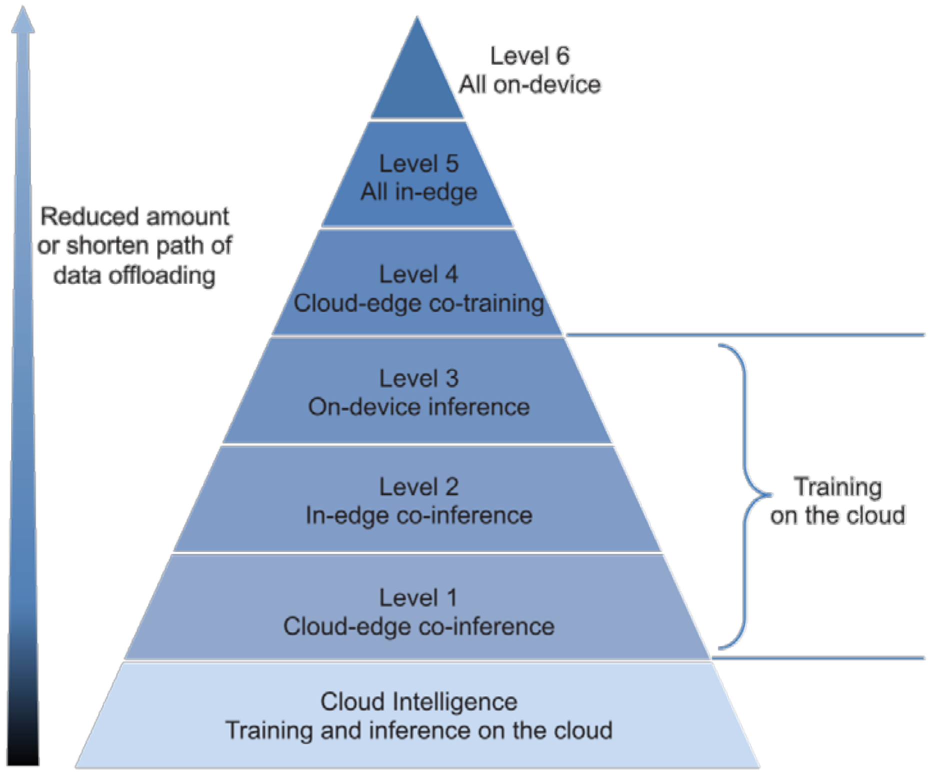
privacy

Solution: mobile edge AI

- Processing at "edge" instead of "cloud"



Levels of edge AI



Six levels of edge AI based on the path of data offloading: cloud-edge-device coordination via data offloading

Fig. credit: Zhou

This talk

- **Part I: mathematics in edge AI**
  - Provable guarantees for nonconvex machine learning
  - Communication-efficient distributed machine learning
- **Part II: edge inference process**
  - Communication-efficient on-device distributed inference
  - Energy-efficient edge cooperative inference
- **Part III: edge training process**
  - Over-the-air computation for federated learning
  - Intelligent reflecting surface empowered federated learning

Mobile Edge Artificial Intelligence: Opportunities and Challenges

Part I: Theory

Yuanming Shi

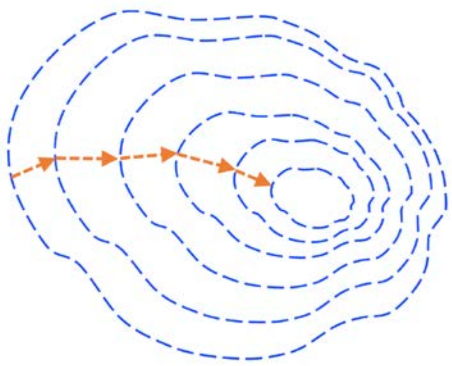
ShanghaiTech University

上海科技大学
ShanghaiTech University

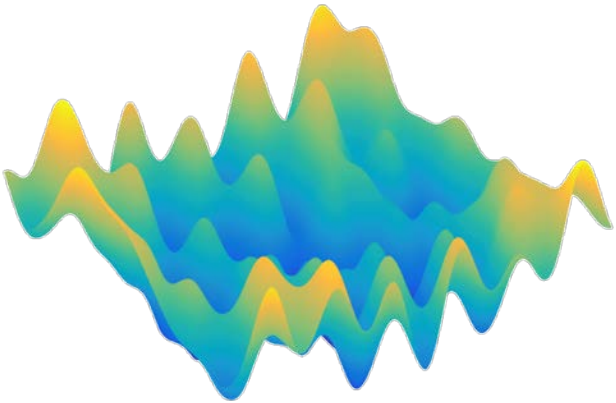
Outline

- **Motivations**
  - Taming nonconvexity in statistical machine learning
  - Communication challenges in distributed machine learning
- **Two Vignettes:**
  - **Provable guarantees for nonconvex machine learning**
    - Why nonconvex optimization?
    - Blind demixing via implicitly regularized Wirtinger flow
  - **Communication-efficient distributed machine learning**
    - Why gradient quantization?
    - Learning polynomial neural networks via quantized SGD

Vignettes A: Provable guarantees for nonconvex machine learning



Why nonconvex optimization?

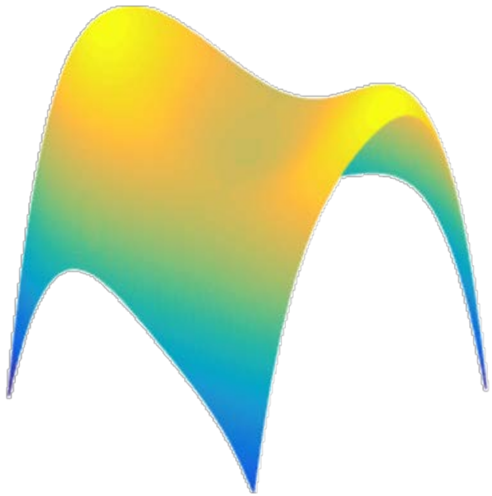


Nonconvex problems are everywhere

- Empirical risk minimization is usually nonconvex

$$\underset{\boldsymbol{x}}{\text{minimize}} \quad f(\boldsymbol{x};\boldsymbol{\theta})$$

  - low-rank matrix completion
  - blind deconvolution/demixing
  - dictionary learning
  - phase retrieval
  - mixture models
  - deep learning
  - …



Nonconvex optimization may be super scary

- **Challenges:** saddle points, local optima, bumps,…

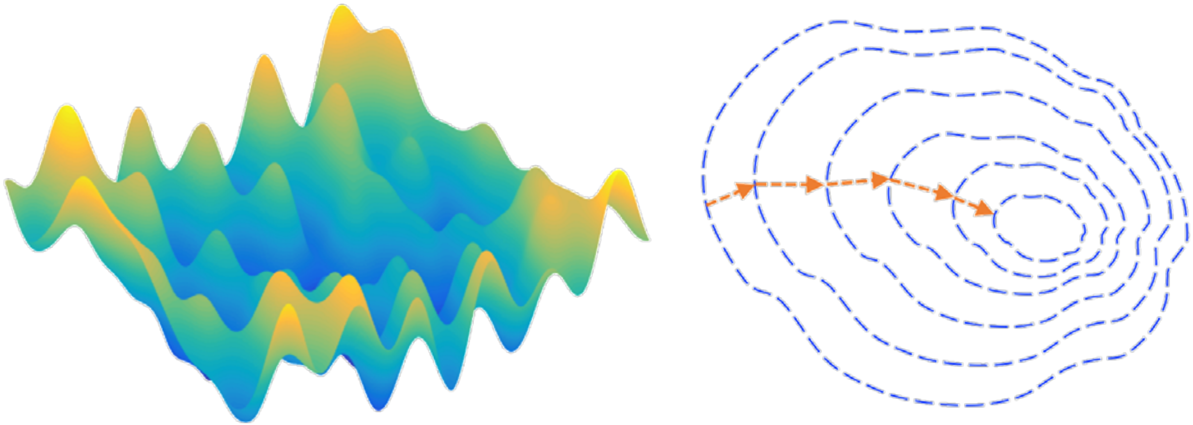
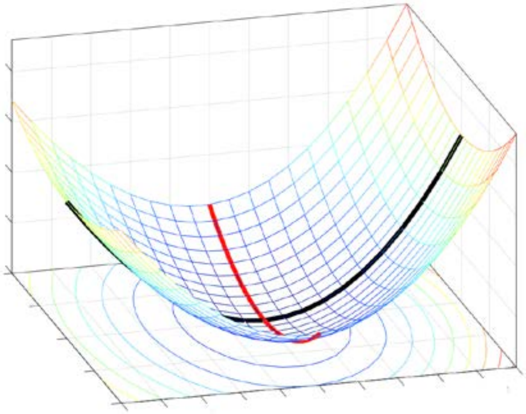


Fig. credit: Chen

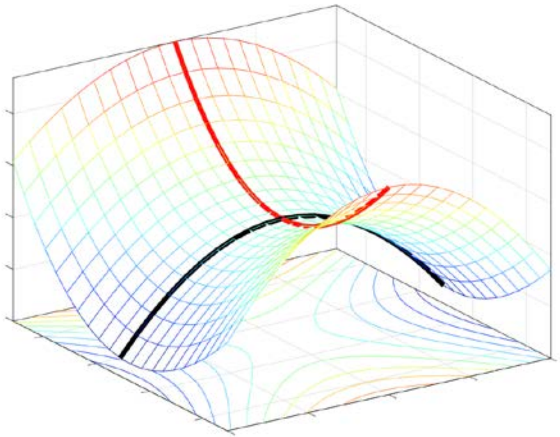
- **Fact:** they are usually solved on a daily basis via simple algorithms like (stochastic) gradient descent

Sometimes they are much nicer than we think

- Under certain statistical models, we see benign global geometry: no spurious local optima



global minimum



saddle point

Statistical models come to rescue

- **Blessings:** when data are generated by certain statistical models, problems are often much nicer than worst-case instances

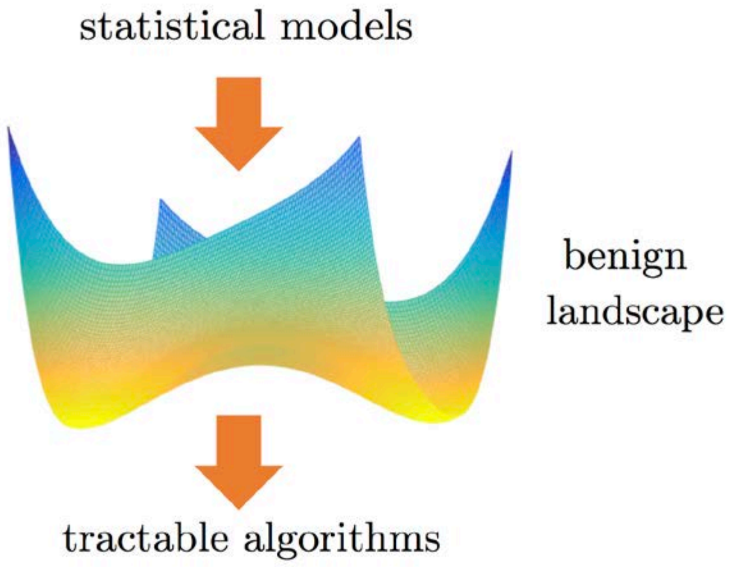
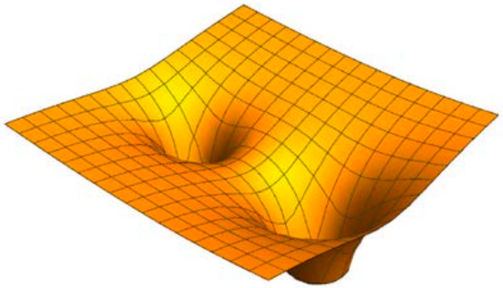


Fig. credit: Chen

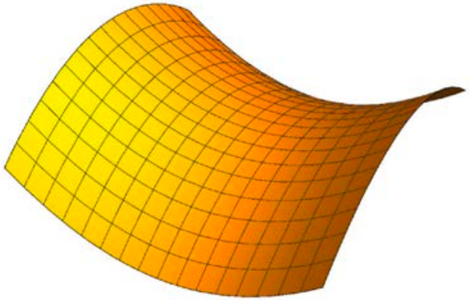
First-order stationary points

- Saddle points and local minima:

$$\lambda_{\min}(\nabla^2 f(\boldsymbol{z})) \begin{cases} > 0 & \text{local minimum} \\ = 0 & \text{local minimum or saddle point} \\ < 0 & \text{strict saddle point} \end{cases}$$



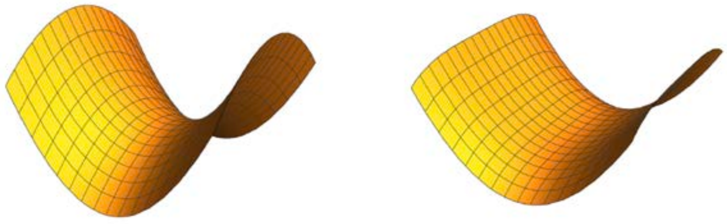
Local minima



Saddle points/local maxima

First-order stationary points

- **Applications:** PCA, matrix completion, dictionary learning etc.
  - **Local minima:** either all local minima **are** global minima or all local minima **as good as** global minima
  - **Saddle points: very poor** compared to global minima; **several** such points



Strict saddle point Non-strict saddle point

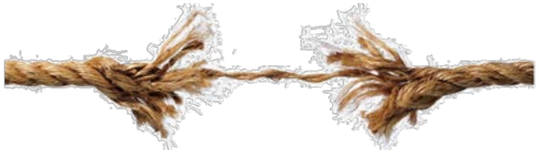
- **Bottomline:** local minima much more desirable than saddle points

How to escape saddle points efficiently?

Statistics meets optimization

- **Proposal:** separation of landscape analysis and generic algorithm design

landscape analysis (statistics)



generic algorithms (optimization)

all local minima are global minima

- dictionary learning (Sun et al. '15)
- phase retrieval (Sun et al. '16)
- matrix completion (Ge et al. '16)
- synchronization (Bandeira et al. '16)
- inverting deep neural nets (Hand et al. '17)
- ...

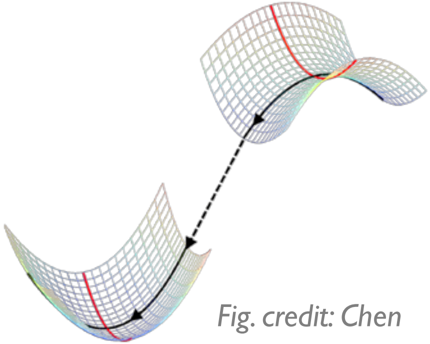


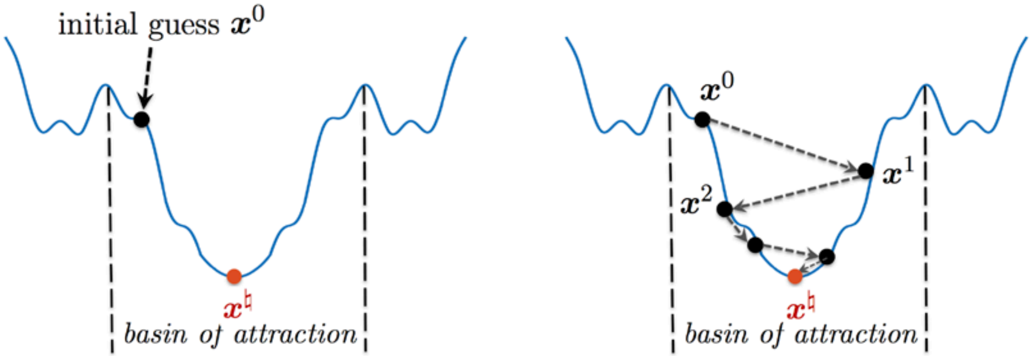
Fig. credit: Chen

all the saddle points can be escaped

- gradient descent (Lee et al. '16)
- trust region method (Sun et al. '16)
- perturbed GD (Jin et al. '17)
- cubic regularization (Agarwal et al. '17)
- Natasha (Allen-Zhu '17)
- ...

Issue: conservative computational guarantees for specific problems (e.g., phase retrieval, blind deconvolution, matrix completion)

Blind demixing via implicitly regularized Wirtinger flow



Solution: blending landscape and convergence analysis

Case study: blind deconvolution

- In many science and engineering problems, the observed signal can be modeled as:

$$z(t) = f(t) * g(t)$$

where $*$ is the convolution operator

  - $f(t)$ is a physical signal of interest
  - $g(t)$ is the impulse response of the sensory system

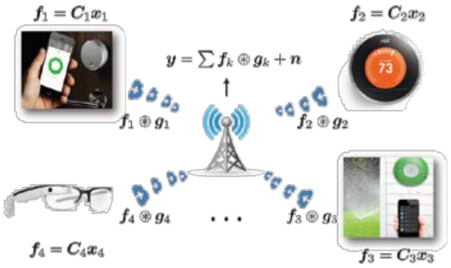
- **Applications:** astronomy, neuroscience, image processing, computer vision, wireless communications, microscopy data processing,…

- **Blind deconvolution:** estimate $f(t)$ and $g(t)$ given $z(t)$

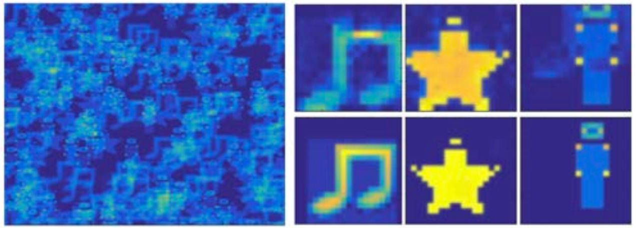
Case study: blind demixing

- The received measurement consists of the sum of all convolved signals

$$z(t) = \sum_{i=1}^{s} f_i(t) * g_i(t)$$



low-latency communication for IoT



convolutional dictionary learning (multi kernel)

- **Applications:** IoT, dictionary learning, neural spike sorting,…
- **Blind demixing:** estimate $\{f_i(t)\}$ and $\{g_i(t)\}$ given $z(t)$

Bilinear model

- Translate into the frequency domain…

$$z = \sum_{i=1}^{s} f_i \odot g_i \in \mathbb{C}^m$$

- **Subspace assumptions:** f_i and g_i lie in some known low-dimensional subspaces

$$f_i = A_i x_i^\natural \in \mathbb{C}^m \qquad g_i = B h_i^\natural \in \mathbb{C}^m$$

where $A_i = [a_{i1}, \cdots, a_{im}]^* \in \mathbb{C}^{m \times L}$, $B = [b_1, \cdots, b_m]^* \in \mathbb{C}^{m \times K}$ and $L, K \ll m$

$$a_{ij} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(0, I) \qquad \{b_j\} : \text{partial Fourier basis}$$

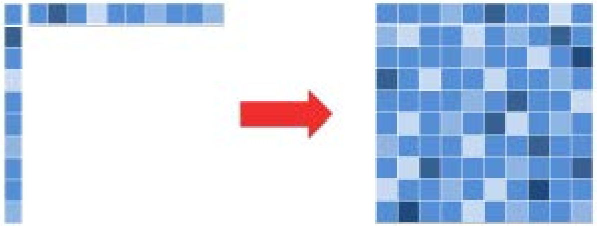
- **Demixing from bilinear measurements:**

$$\text{find} \quad \{x_i\}, \{h_i\} \quad \text{subject to} \quad z_j = \sum_{i=1}^{s} b_j^* h_i x_i^* a_{ij}, \quad 1 \le j \le m$$

An equivalent view: low-rank factorization

- **Lifting:** introduce $\boldsymbol{M}_k^\natural = \boldsymbol{h}_k^\natural \boldsymbol{x}_k^{\natural *}$ to linearize constraints

$$z_j = \sum_{i=1}^{s} \boldsymbol{b}_i^* \boldsymbol{h}_i^\natural \boldsymbol{x}_i^{\natural *} \boldsymbol{a}_{ij} = \sum_{i=1}^{s} \boldsymbol{b}_i^* \underbrace{(\boldsymbol{h}_i^\natural \boldsymbol{x}_i^{\natural *})}_{\boldsymbol{M}_i^\natural \in \mathbb{C}^{K \times L}} \boldsymbol{a}_{ij}$$



- **Low-rank matrix optimization problem**

$$\begin{aligned} \text{find} \quad & \{\boldsymbol{M}_i\} \\ \text{subject to} \quad & z_j = \sum_{i=1}^{s} \boldsymbol{b}_i^* \boldsymbol{M}_i \boldsymbol{a}_{ij}, \quad j = 1, \cdots, m \\ & \operatorname{rank}(\boldsymbol{M}_i) = 1, \; i = 1, \cdots, s, \end{aligned}$$

Convex relaxation

- Ling and Strohmer (TIT'2017) proposed to solve the **nuclear norm minimization problem:**

$$\begin{aligned} &\text{minimize} && \sum_{k=1}^{s} \|\boldsymbol{M}_k\|_* \\ &\text{subject to} && z_j = \sum_{k=1}^{s} \boldsymbol{b}_k^* \boldsymbol{M}_k \boldsymbol{a}_{kj}, \quad j = 1, \cdots, m \end{aligned}$$

$\boldsymbol{a}_{kj} \overset{\text{i.i.d.}}{\sim} \mathcal{CN}(\boldsymbol{0}, \boldsymbol{I})$

$\{\boldsymbol{b}_j\}$: partial Fourier basis

  - **Sample-efficient:** $m \gtrsim s^2 \max\{K, L\} \log^2 m$ samples for exact recovery if $\{\boldsymbol{b}_j\}$ is incoherent w.r.t. $\{\boldsymbol{h}_k^\natural\}$
  - **Computational-expensive:** SDP in the lifting space

Can we solve the nonconvex matrix optimization problem directly?

A natural least-squares formulation

- **Goal:** demixing from bilinear measurements

Given: $$y_j = \sum_{i=1}^{s} \boldsymbol{b}_j^* \boldsymbol{h}_i^\natural \boldsymbol{x}_i^{\natural *} \boldsymbol{a}_{ij}, \quad 1 \le j \le m$$

$$\underset{\{\boldsymbol{h}_k\},\{\boldsymbol{x}_k\}}{\text{minimize}} \; f(\boldsymbol{h}, \boldsymbol{x}) := \sum_{j=1}^{m} \sum_{k=1}^{s} \left(\boldsymbol{b}_j^* \boldsymbol{h}_k \boldsymbol{x}_k^* \boldsymbol{a}_{kj} - y_j \right)^2$$

  - **Pros:** computational-efficient in the natural parameter space
  - **Cons:** $f(\cdot)$ is nonconvex: bilinear constraint, scaling ambiguity

Wirtinger flow

- Least-square minimization via Wirtinger flow (Candes, Li, Soltanolkotabi '14)

$$\underset{\{\boldsymbol{h}_k\},\{\boldsymbol{x}_k\}}{\text{minimize}} \; f(\boldsymbol{h},\boldsymbol{x}) := \sum_{j=1}^{m}\sum_{k=1}^{s}\left(\boldsymbol{b}_j^*\boldsymbol{h}_k\boldsymbol{x}_k^*\boldsymbol{a}_{kj} - y_j\right)^2$$

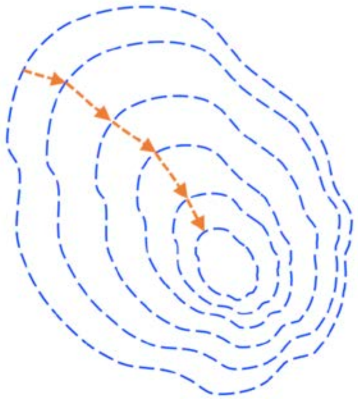
  - **Spectral initialization by top eigenvector of**

$$\boldsymbol{M}_k := \textstyle\sum_{j=1}^{m} \boldsymbol{y}_j\boldsymbol{b}_j\boldsymbol{a}_{kj}^*, \quad k = 1,\cdots,s$$

  - **Gradient iterations**

$$\boldsymbol{h}_k^{t+1} = \boldsymbol{h}_k^t - \eta\frac{1}{\|\boldsymbol{x}_k^t\|_2^2}\nabla_{\boldsymbol{h}_k} f(\boldsymbol{h}^t,\boldsymbol{x}^t)$$

$$\boldsymbol{x}_k^{t+1} = \boldsymbol{x}_k^t - \eta\frac{1}{\|\boldsymbol{h}_k^t\|_2^2}\nabla_{\boldsymbol{x}_k} f(\boldsymbol{h}^t,\boldsymbol{x}^t)$$



Two-stage approach

- **Initialize** within local basin sufficiently close to ground-truth (i.e., strongly convex, no saddle points/ local minima)
- **Iterative refinement** via some iterative optimization algorithms

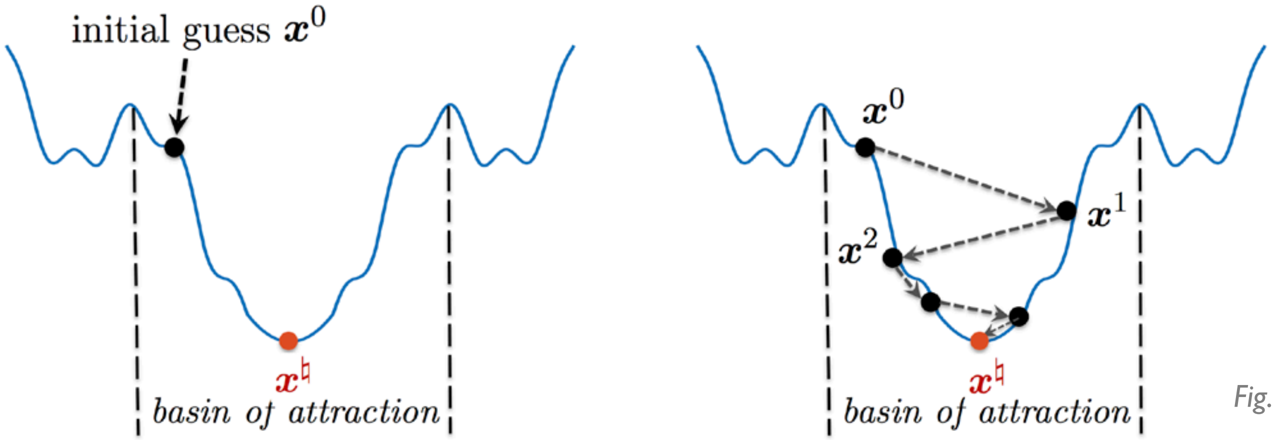
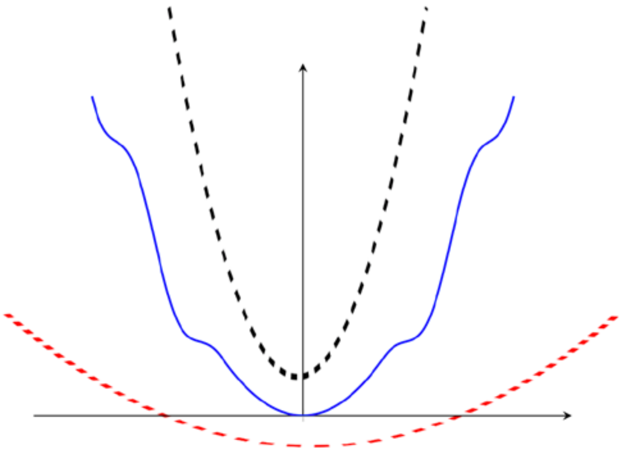


Fig. credit: Chen

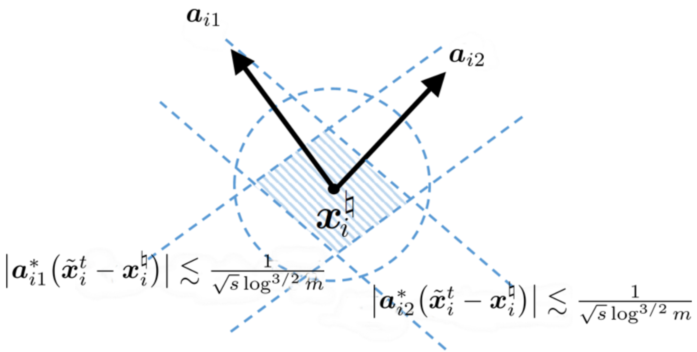
Gradient descent theory

- Two standard conditions that enable **geometric convergence** of GD
  - (local) restricted strong convexity
  - (local) smoothness



Gradient descent theory

- **Question:** which region enjoys both strong convexity and smoothness?



  - x is not far away from $x^\natural$ (convexity)
  - x is incoherent w.r.t. sampling vectors (incoherence region for smoothness)

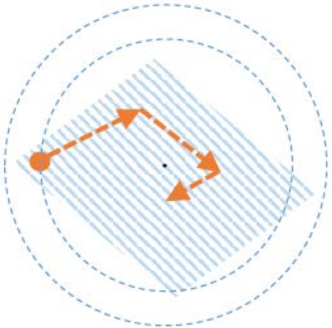
Prior works suggest enforcing *regularization* (e.g., regularized loss [Ling & Strohmer'17]) to promote incoherence

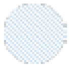
Our finding: WF is implicitly regularized

- WF (GD) implicitly forces iterates to remain **incoherent** with $\{\boldsymbol{a}_{ij}\}$

$$\max_{1\le i\le s, 1\le j\le m} \left| \boldsymbol{a}_{ij}^{*} \left(\alpha_i^t \boldsymbol{x}_i^t - \boldsymbol{x}_i^\natural \right) \right| \lesssim \frac{1}{\sqrt{s}\log^{3/2} m} \|\boldsymbol{x}_i^\natural\|_2$$

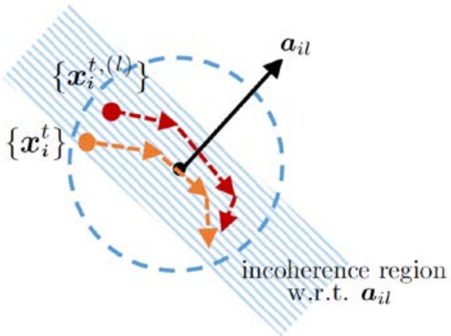
  - ➢ cannot be derived from generic optimization theory
  - ➢ relies on finer *statistical analysis* for entire trajectory of GD





region of local strong convexity and smoothness

Key proof idea: leave-one-out analysis



- introduce leave-one-out iterates $\boldsymbol{x}_i^{t,(l)}$ by running WF without *l*-th sample
- leave-one-out iterate $\boldsymbol{x}_i^{t,(l)}$ is independent of $\boldsymbol{a}_{il}$
- leave-one-out iterate $\boldsymbol{x}_i^{t,(l)} \approx$ true iterate $\boldsymbol{x}_i^t$
- $\boldsymbol{x}_i^t$ is nearly independent of (i.e., nearly orthogonal to) $\boldsymbol{a}_{il}$

Theoretical guarantees

- With i.i.d. Gaussian design, WF **(regularization-free)** achieves
  - **Incoherence**

$$\max_{1\le i\le s, 1\le j\le m} \left| \boldsymbol{a}_{ij}^{*} \left(\alpha_i^t \boldsymbol{x}_i^t - \boldsymbol{x}_i^{\natural} \right) \right| \lesssim \frac{1}{\sqrt{s}\log^{3/2} m} \|\boldsymbol{x}_i^{\natural}\|_2$$

  - **Near-linear convergence rate**

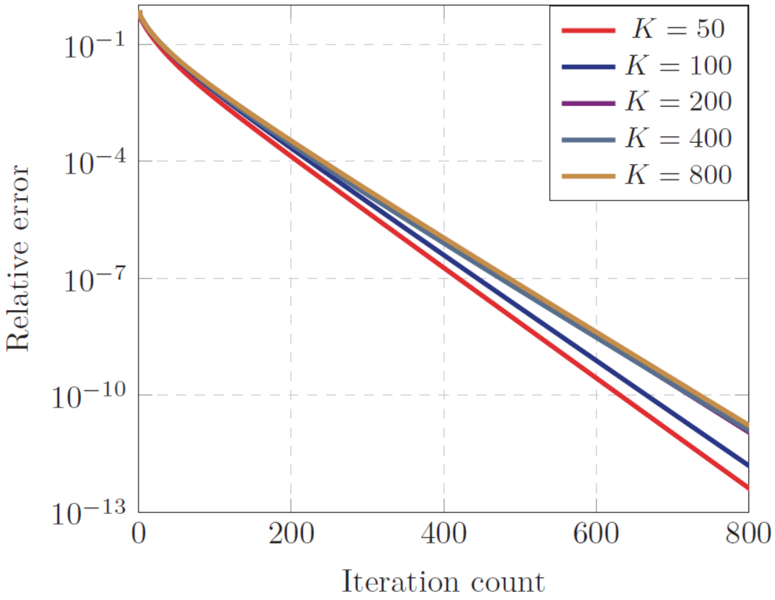
$$\mathrm{dist}(\boldsymbol{z}^t, \boldsymbol{z}^{\natural}) \lesssim \left(1 - \frac{\eta}{16\kappa}\right)^t \frac{1}{\log^2 m}$$

- **Summary:**
  - **Sample size:** $m \gtrsim s^2 \max\{K, L\} \mathrm{poly} \log m$
  - **Stepsize:** $\eta \asymp s^{-1}$ vs. $\eta \lesssim (sm)^{-1}$ [Ling & Strohmer'17]
  - **Computational complexity:** $\mathcal{O}(s \log \frac{1}{\varepsilon})$ vs. $\mathcal{O}(sm \log \frac{1}{\varepsilon})$ [Ling & Strohmer'17]

[Ref] J. Dong and Y. Shi, "Nonconvex demixing from bilinear measurements," *IEEE Trans. Signal Process.*, vol. 66, no. 19, pp. 5152-5166, Oct., 2018.

Numerical results

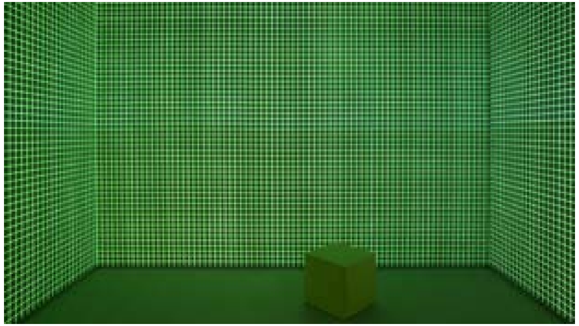
- stepsize: $\eta = 0.1$
- number of users: $s = 10$
- sample size: $m = 50K$



linear convergence:
WF attains ε- accuracy within $\mathcal{O}(s\log\frac{1}{\varepsilon})$ iterations

Vignettes B: Communication-efficient distributed machine learning

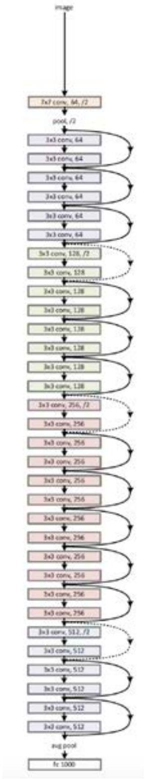
Why gradient quantization?



The practical problem

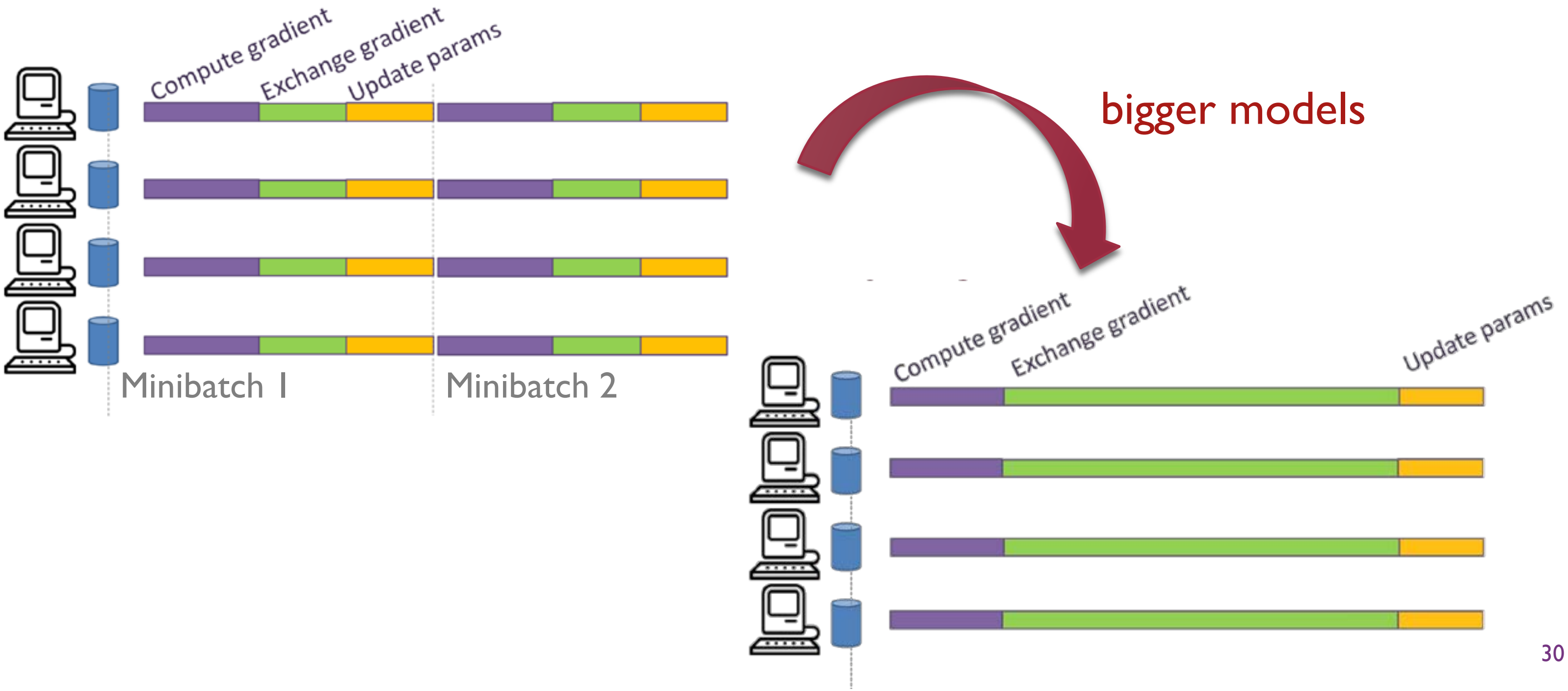
- **Goal:** training large-scale machine learning models efficiently

- **Large datasets:**
  - ImageNet: 1.6 million images (~300GB)
  - NIST2000 Switchboard dataset: 2000 hours
- **Large models:**
  - ResNet-152 [He et al. 2015]: 152 layers, 60 million parameters
  - LACEA [Yu et al. 2016]: 22 layers, 65 million parameters



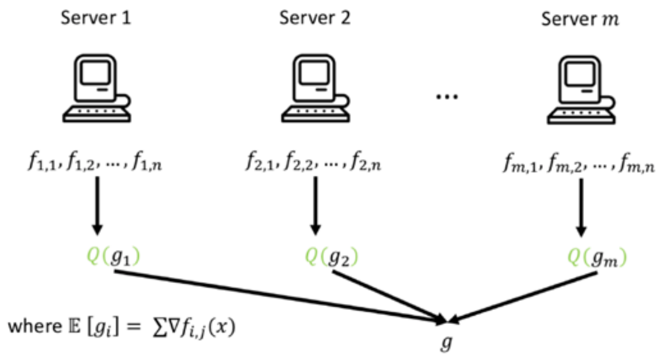
Data parallel stochastic gradient descent

- **Challenge:** communication is a bottleneck to scalability for large model



Quantized SGD

- **Idea:** stochastically quantize each coordinate



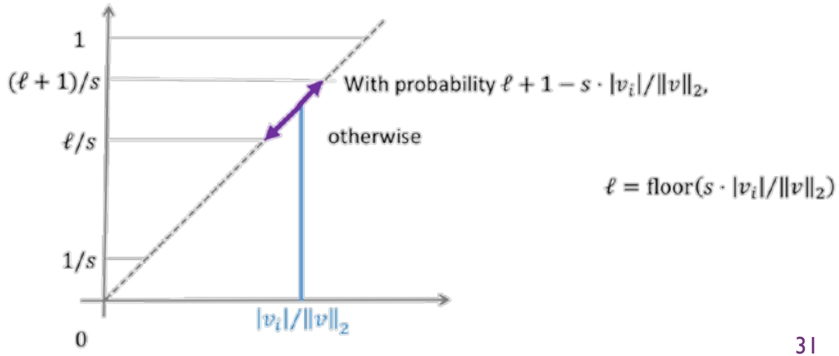
Update: $x_{t+1} \leftarrow x_t - \eta \cdot g$

Question: how to provide optimality guarantees of quantized SGD for nonconvex machine learning?

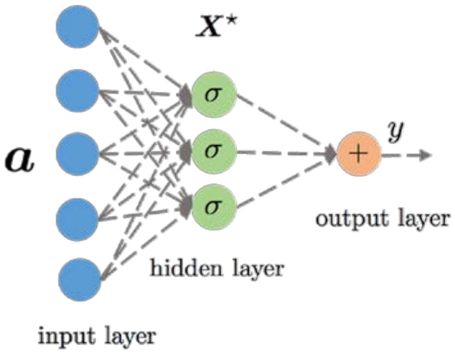
Q is a quantization function which can be communicated with fewer bits

$$Q[v; s] = \|v\|_2 \cdot \mathrm{sgn}(v_i) \cdot \xi_i(v, s)$$

ξ_i is defined by

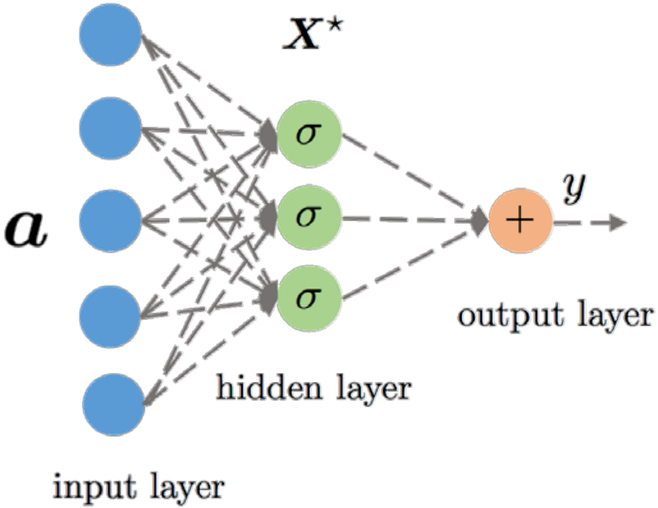


Learning polynomial neural networks via quantized SGD



Polynomial neural networks

- Learning neural networks with quadratic activation



input features: $\boldsymbol{a}$

weights: $\boldsymbol{X}^\star = [\boldsymbol{x}_1^\star, \cdots, \boldsymbol{x}_r^\star]$

output:

$$y = \sum_{i=1}^r \sigma(\boldsymbol{a}^T \boldsymbol{x}^\star) \overset{\sigma(z)=z^2}{:=} \sum_{i=1}^r (\boldsymbol{a}^T \boldsymbol{x}_i^\star)^2$$

Quantized stochastic gradient descent

- Mini-batch SGD

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \mu \frac{1}{m} \sum_{j=1}^{m} \nabla \mathcal{L}_{i_t^{(j)}} \left(\boldsymbol{W}_t \right)$$

  - sample indices $i_t^{(j)}$ uniformly with replacement from $\{1, 2, 3, \ldots, n\}$
  - the generalized gradient of the loss function

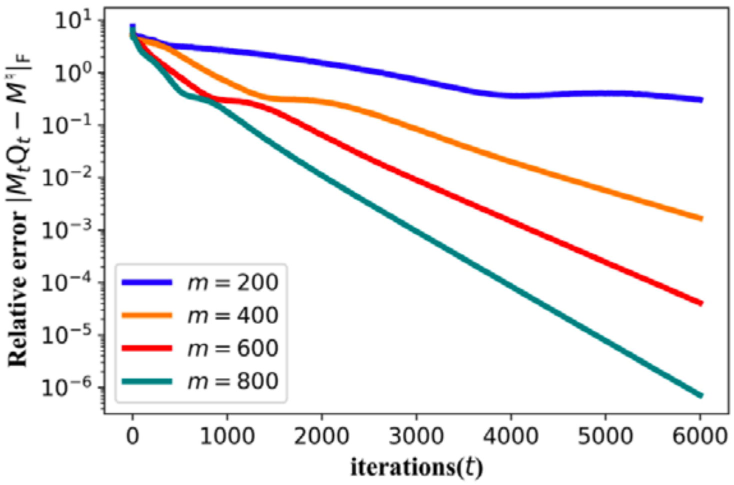
$$\nabla \mathcal{L}_i \left(\boldsymbol{W} \right) = (\left\| \boldsymbol{x}_i^T \boldsymbol{W}_t \right\|_2^2 - y_i) \boldsymbol{x}_i \boldsymbol{x}_i^T \boldsymbol{W}$$

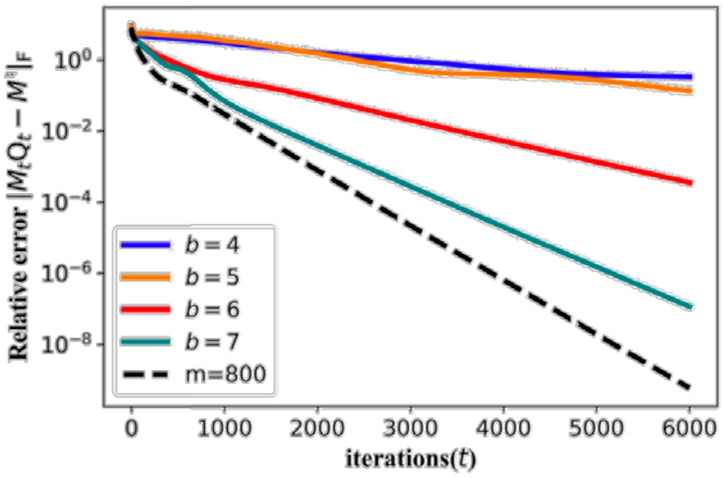
- **Quantized SGD**

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \mu \cdot \frac{1}{K} \sum_{k=1}^{K} Q_s \left(\nabla \left\{ \frac{1}{m_k} \sum_{j=1}^{m_k} \mathcal{L}_{i_t^{(j)}} \left(\boldsymbol{W}_t \right) \right\} \right)$$

Provable guarantees for QSGD

- *Theorem 1:* SGD converges at linear rate to the globally optimal solution
- *Theorem 2:* QSGD provably maintains similar convergence rate of SGD





Concluding remarks

- **Implicitly regularized Wirtinger flow**
  - **Implicit regularization:** vanilla gradient descent automatically forces iterates to stay *incoherent*
  - Even **simplest** nonconvex methods are remarkably efficient under suitable **statistical** models

- **Communication-efficient quantized SGD**
  - QSGD provably maintains the similar convergence rate of SGD to a globally optimal solution
  - Significantly reduce the communication cost: tradeoffs between computation and communication

Future directions

- **Deep and machine learning with provable guarantees**
  - information theory, random matrix theory, interpretability,…

- **Communication-efficient learning algorithms**
  - vector quantization schemes, decentralized algorithms, zero-order algorithms, second-order algorithms, federated optimization, ADMM, …

Mobile Edge Artificial Intelligence: Opportunities and Challenges

Part II: Inference

Yuanming Shi

ShanghaiTech University

上海科技大学
ShanghaiTech University

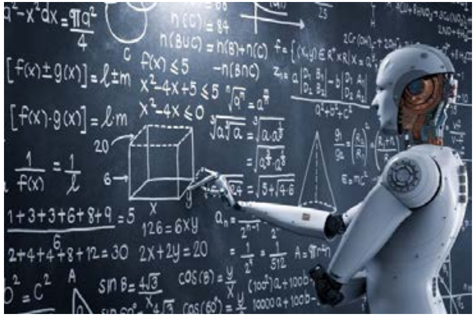
Outline

- **Motivations**
  - Latency, power, storage
- **Two vignettes:**
  - **Communication-efficient on-device distributed inference**
    - Why on-device inference?
    - Data shuffling via generalized interference alignment
  - **Energy-efficient edge cooperative inference**
    - Why inference at network edge?
    - Edge inference via wireless cooperative transmission

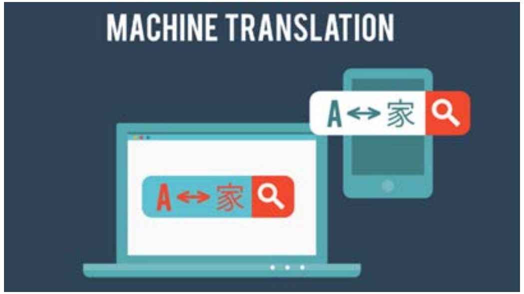
Why edge inference?

AI is changing our lives

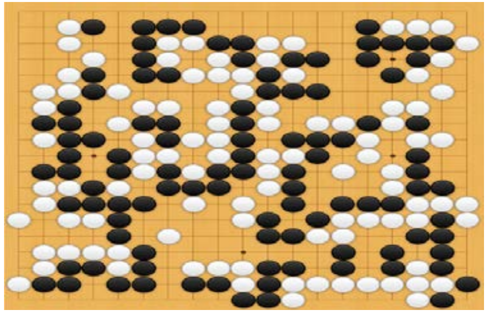
self-driving car



smart robots



machine translation



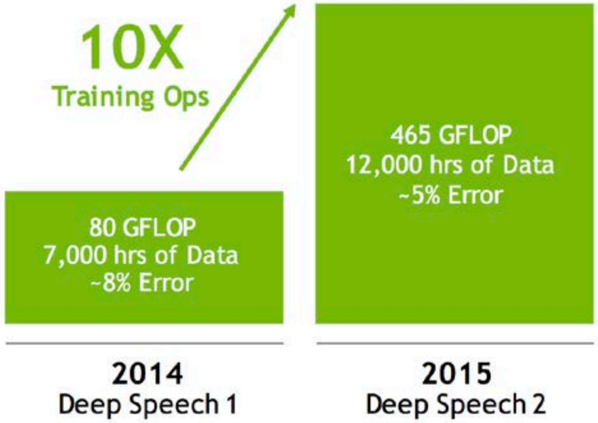
AlphaGo

Models are getting larger

image recognition

Microsoft

speech recognition



Baidu百度

Fig. credit: Dally

The first challenge: model size

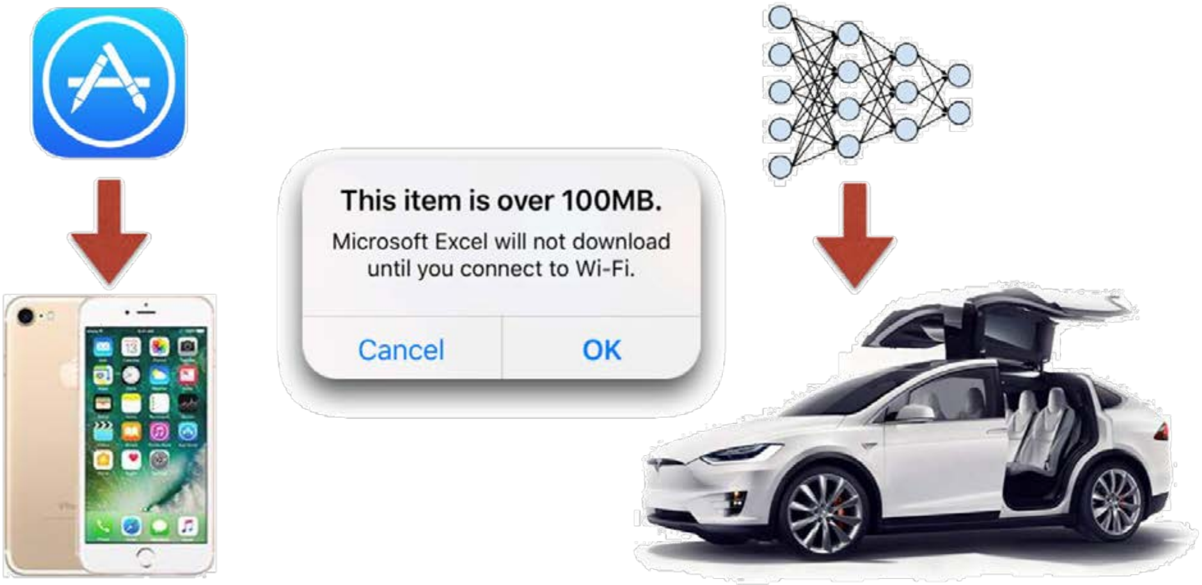


Fig. credit: Han

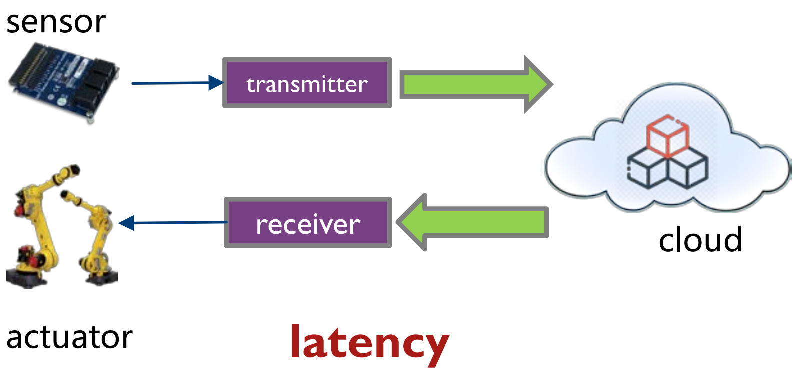
difficult to distribute large models through over-the-air update

The second challenging: speed

	Error rate	Training time
ResNet18:	10.76%	2.5 days
ResNet50:	7.02%	5 days
ResNet101:	6.21%	1 week
ResNet152:	6.16%	1.5 weeks

long training time limits ML researcher's productivity

communication



latency

processing at "Edge" instead of the "Cloud"

The third challenge: energy

AlphaGo: 1920 CPUs and 280 GPUs,
$3000 electric bill per game

on mobile: drains battery
on data-center: increases TCO

larger model-more memory reference-more energy

<u>How to make deep learning more efficient?</u>

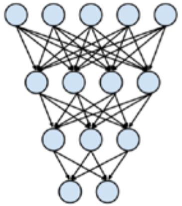
low latency, low power

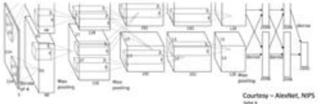
Vignettes A: On-device distributed inference

low latency

On-device inference: the setup

weights/parameters





model

training hardware

inference hardware

MapReduce: a general computing framework

- Active research area: how to fit different jobs into this framework

N subfiles, *K* servers, *Q* keys

input File

N subfiles

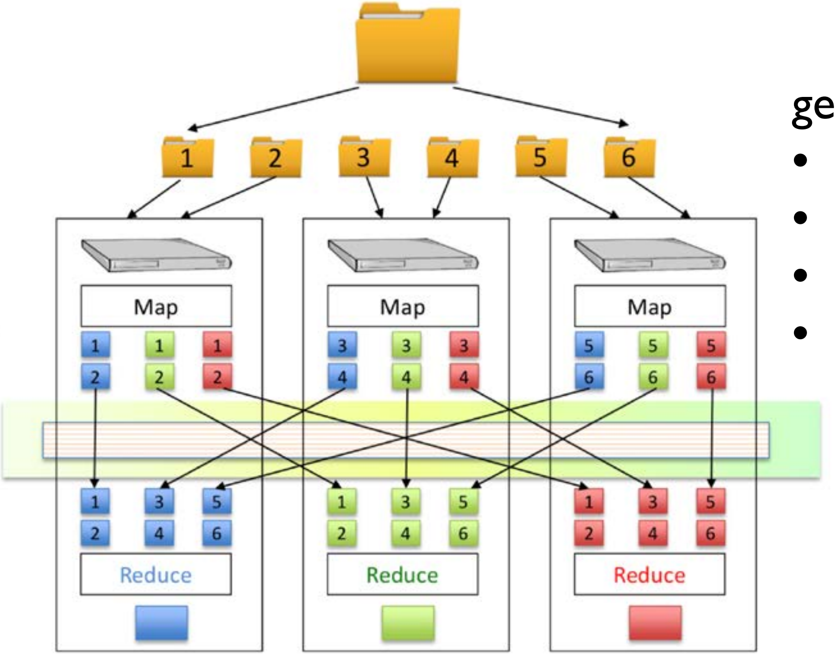
K servers

intermediate (key, value)

(blue, 1 , 1)

shuffling phase

Q keys



general framework

- Matrix
- Distributed ML
- Page rank
- …

Fig. credit: Avestimehr

Wireless MapReduce: computation model

- **Goal:** low-latency (communication-efficient) on-device inference
- **Challenges:** the dataset is too large to be stored in a single mobile device (e.g., a feature library of objects)
- **Solution:** stored N files $\{f_1, \cdots, f_N\}$ across devices, each can only store up to μ files, supported by distributed computing framework MapReduce

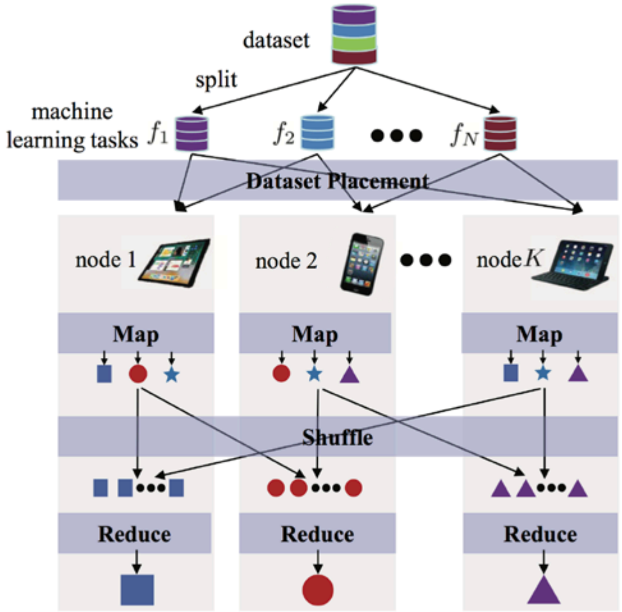
$$\phi_k(d_k; f_1, \cdots, f_N) = h_k(g_{k,1}(d_k; f_1), \cdots, g_{k,N}(d_k; f_N))$$

  - **Map** function: $w_{k,t} = g_{k,t}(d_k; f_t)$ (d_k input data)
  - **Reduce** function: $h_k(w_{k,1}, \cdots, w_{k,N})$ ($w_{k,t}$ intermediate values)

Wireless MapReduce: computation model

- Dataset placement phase: determine the index set of files stored at each node $\mathcal{F}_k \subseteq [N]$
- Map phase: compute intermediate values locally $\{w_{s,t} : s \in [K], t \in \mathcal{F}_k\}$
- Shuffle phase: exchange intermediate values wirelessly among nodes
- Reduce phase: construct the output value using the reduce function

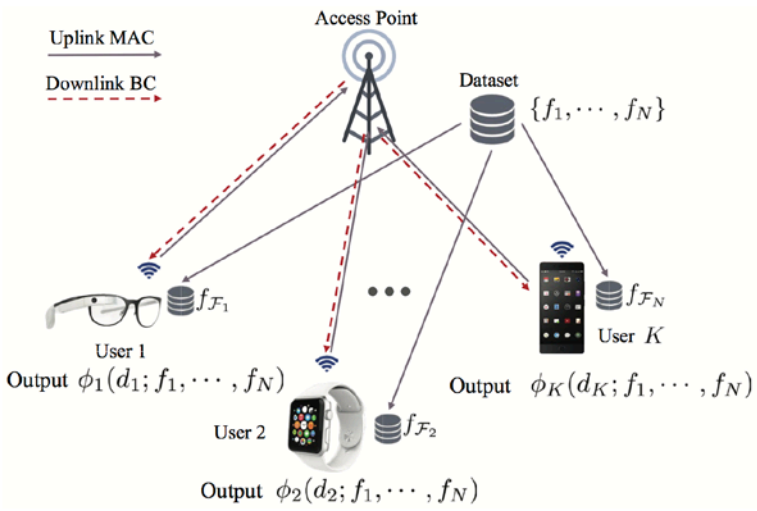
$$h_k(w_{k,1}, \cdots, w_{k,N})$$



on-device distributed inference via wireless MapReduce

Wireless MapReduce: communication model

- Goal: K users (each with L antennas) exchange intermediate values via a wireless access point (M antennas)
  - entire set of messages (intermediate values) $\{W_1, \cdots, W_T\}, T = KN$
  - index set of messages (computed locally) available at user k : $\mathcal{T}_k \subseteq [T]$
  - index set of messages required by user k: $\mathcal{R}_k \subseteq [T]$



wireless distributed computing system

message delivery problem with side information

Wireless MapReduce: communication model

- Uplink multiple access stage:

$$\boldsymbol{y} = \sum_{k=1}^{K} (\boldsymbol{H}_k^u \otimes \boldsymbol{I}_r)\boldsymbol{x}_k + \boldsymbol{n}^u$$

  - $\boldsymbol{y} \in \mathbb{C}^{Mr}$: received at the AP; $\boldsymbol{x}_k \in \mathbb{C}^{Lr}$: transmitted by user k; r: channel uses

- Downlink broadcasting stage:

$$\boldsymbol{z}_k = (\boldsymbol{H}_k^d \otimes \boldsymbol{I}_r)\boldsymbol{y} + \boldsymbol{n}_k^d$$

  - $\boldsymbol{z}_k \in \mathbb{C}^{Lr}$: received by mobile user k

- Overall input-output relationship from mobile user to mobile user

$$\boldsymbol{z}_k = \sum_{i=1}^{K} (\boldsymbol{H}_{ki} \otimes \boldsymbol{I}_r)\boldsymbol{x}_i + \boldsymbol{n}_k$$

$$\boldsymbol{H}_{ki} = \boldsymbol{H}_k^d \boldsymbol{H}_i^u \in \mathbb{C}^{L \times L}$$

$$\boldsymbol{n}_k = (\boldsymbol{H}_k^d \otimes \boldsymbol{I}_r)\boldsymbol{n}^u + \boldsymbol{n}_k^d$$

Interference alignment conditions

- Precoding matrix: $\boldsymbol{V}_{kj} \in \mathbb{C}^{Lr \times d}$, $\boldsymbol{x}_k = \sum_{j \in \mathcal{T}_k} \boldsymbol{V}_{kj} \boldsymbol{s}_j$
- Decoding matrix: $\boldsymbol{U}_{kl} \in \mathbb{C}^{d \times Lr}$

$$\begin{aligned}\tilde{\boldsymbol{z}}_{kl} &= \boldsymbol{U}_{kl}\boldsymbol{z}_k = \boldsymbol{U}_{kl}\sum_{i=1}^{K}(\boldsymbol{H}_{ki}\otimes\boldsymbol{I}_r)\sum_{j\in\mathcal{T}_i}\boldsymbol{V}_{ij}\boldsymbol{s}_j + \boldsymbol{n}_{kl}\\ &= \mathcal{I}_1(\underbrace{\boldsymbol{s}_l}_{\text{desired message}}) + \mathcal{I}_2(\underbrace{\{\boldsymbol{s}_j : j\in\mathcal{T}_k\}}_{\text{locally available messages}}) + \mathcal{I}_3(\underbrace{\{\boldsymbol{s}_j : j\notin\mathcal{T}_k\cup\{l\}\}}_{\text{interferences}}) + \tilde{\boldsymbol{n}}_{kl}\end{aligned}$$

- **Interference alignment conditions**

$$\det\left(\sum_{i:l\in\mathcal{T}_i}\boldsymbol{U}_{kl}(\boldsymbol{H}_{ki}\otimes\boldsymbol{I}_r)\boldsymbol{V}_{il}\right)\neq 0,$$

$$\sum_{i:j\in\mathcal{T}_i}\boldsymbol{U}_{kl}(\boldsymbol{H}_{ki}\otimes\boldsymbol{I}_r)\boldsymbol{V}_{ij} = \boldsymbol{0},\ j\notin\mathcal{T}_k\cup\{l\}$$

w.l.o.g. $\sum_{i:l\in\mathcal{T}_i}\boldsymbol{U}_{kl}(\boldsymbol{H}_{ki}\otimes\boldsymbol{I}_r)\boldsymbol{V}_{il} = \boldsymbol{I}$

symmetric DoF: $\mathrm{DoF}_{\mathrm{sym}} = d/r$

Generalized low-rank optimization

- Low-rank optimization for interference alignment

$$\mathscr{P}: \underset{\boldsymbol{X}\in\mathbb{C}^{D\times D}}{\text{minimize}} \quad \operatorname{rank}(\boldsymbol{X})$$
$$\text{subject to} \quad \mathcal{A}(\boldsymbol{X}) = \boldsymbol{b}$$

  - the affine constraint encodes the interference alignment conditions

$$\sum_{i:l\in\mathcal{T}_i}\sum_{m=1}^{L}\sum_{n=1}^{L} H_{ki}[m,n]\boldsymbol{X}_{k,l,i,l}[m,n]=\boldsymbol{I},$$

$$\sum_{i:j\in\mathcal{T}_i}\sum_{m=1}^{L}\sum_{n=1}^{L} H_{ki}[m,n]\boldsymbol{X}_{k,l,i,j}[m,n]=\boldsymbol{0},\ \ j\notin\mathcal{T}_k\cup\{l\}$$

  - where $\operatorname{rank}(\boldsymbol{X})=r$, $\boldsymbol{X}\in\mathbb{C}^{LdKT\times LdKT}$, $D=LdKT$

$$\boldsymbol{X}_{k,l,i,j}=[\boldsymbol{X}_{k,l,i,j}[m,n]]=[\boldsymbol{U}_{kl}[m]\boldsymbol{V}_{ij}[n]]$$

Nuclear norm fails

- **Convex relaxation fails:** yields poor performance due to the poor structure of $\mathcal{A}$
  - example: $K = N = 2, \mu = d = L = M = 1$

$$\begin{aligned} \underset{\boldsymbol{X}}{\text{minimize}} \quad & \|\boldsymbol{X}\|_* \\ \text{subject to} \quad & \boldsymbol{X} = \begin{bmatrix} \star & \star & 1/H_{12} & 0 \\ 0 & 1/H_{21} & \star & \star \end{bmatrix} \end{aligned}$$

  - the nuclear norm approach always returns full rank solution while the optimal rank is one

Difference-of-convex programming approach

- Ky Fan k-norm [Watson, 1993]: the sum of largest- k singular values

$$\|\boldsymbol{X}\|_k = \sum_{i=1}^{k} \sigma_i(\boldsymbol{X})$$

  - **The DC representation for rank function**

$$\mathrm{rank}(\boldsymbol{X}) = \min\{k : \|\boldsymbol{X}\|_* - \|\boldsymbol{X}\|_k = 0, k \leq \min\{m, n\}\}$$

- Low-rank optimization via DC programming
  - Find the minimum k such that the optimal objective value is zero

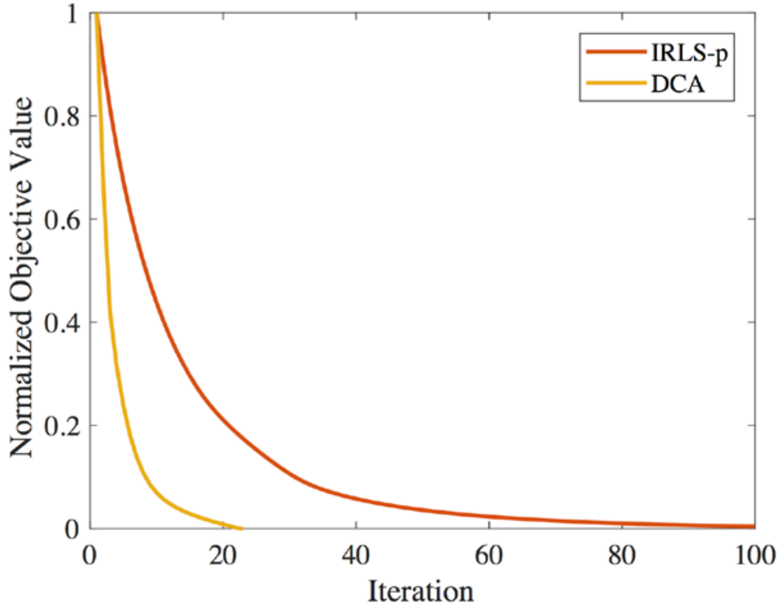
$$\underset{\boldsymbol{X} \in \mathbb{C}^{D \times D}}{\mathrm{minimize}} \|\boldsymbol{X}\|_* - \|\boldsymbol{X}\|_k, \quad \text{subject to} \quad \mathcal{A}(\boldsymbol{X}) = \boldsymbol{b}$$

  - Apply the majorization-minimization (MM) algorithm to iteratively solve a convex approximation subproblem

$$\underset{\boldsymbol{X} \in \mathbb{C}^{D \times D}}{\mathrm{minimize}} \|\boldsymbol{X}\|_* - \mathrm{Tr}(\partial \|\boldsymbol{X}_t\|_k^{\mathsf{H}} \boldsymbol{X}), \quad \text{subject to} \quad \mathcal{A}(\boldsymbol{X}) = \boldsymbol{b}$$

Numerical results

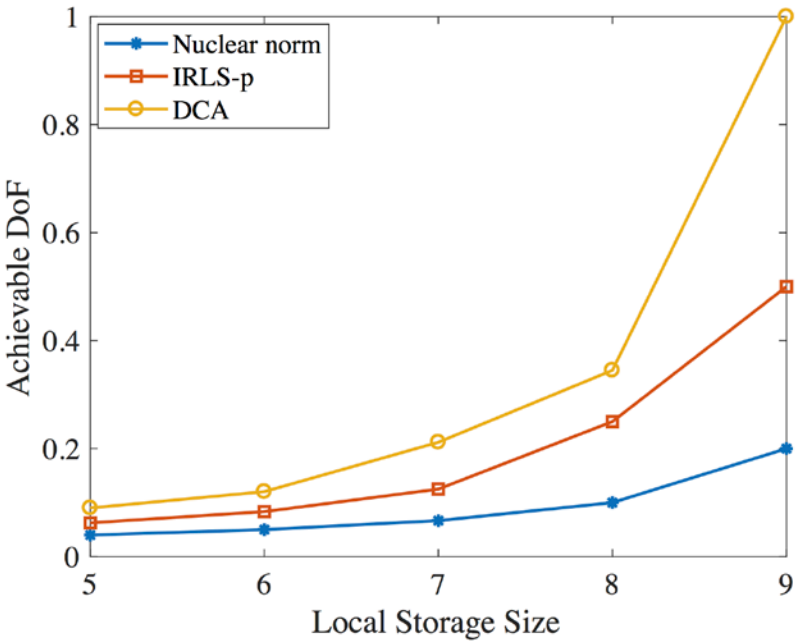
- Convergence results



IRLS-p: iterative reweighted least square algorithm

Numerical results

- Maximum achievable symmetric DoF over local storage size of each user

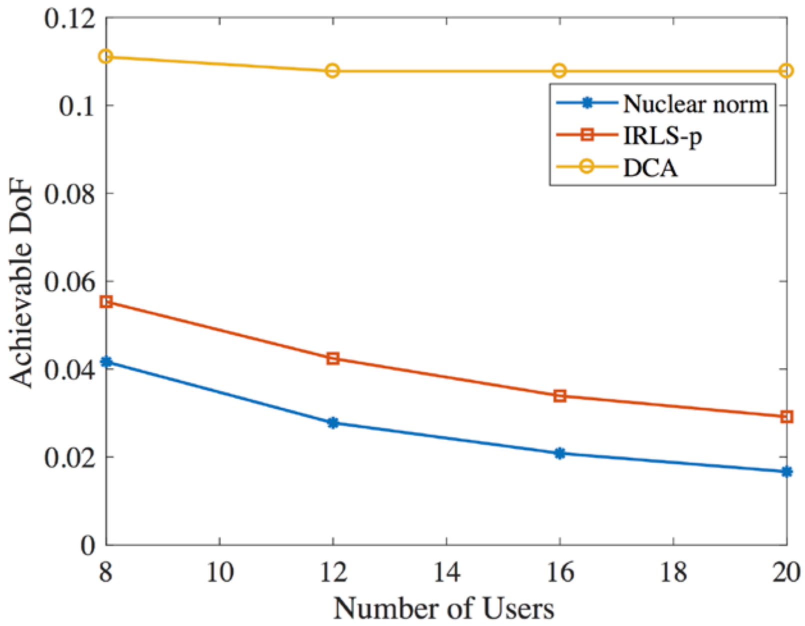


Insights on DC framework:

1. DC function provides a tight approximation for rank function
2. DC algorithm finds better solution for rank minimization problem

Numerical results

- A scalable framework for on-device distributed inference



Insights on more devices:

1. More messages are requested
2. Each file is stored at more devices
3. Opportunities of collaboration for mobile users increase

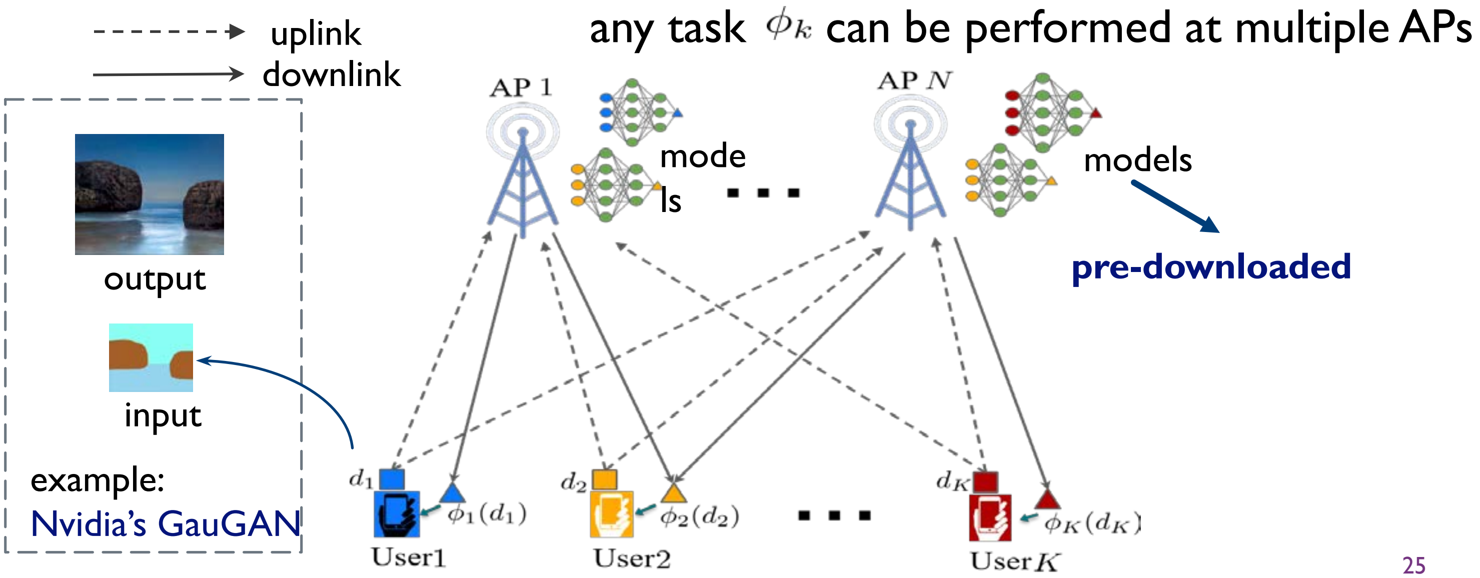
Vignettes B: Edge cooperative inference

low power

Edge inference for deep neural networks

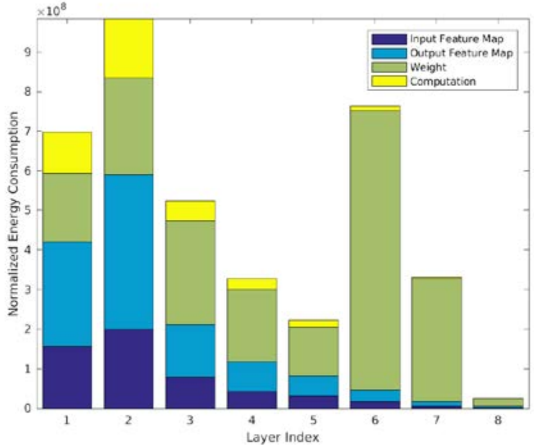
- **Goal:** energy-efficient edge processing framework to execute deep learning inference tasks at the edge computing nodes

which APs shall compute for me?



Computation power consumption

- **Goal:** estimate the power consumption for deep model inference
- Example: power consumption estimation for AlexNet [Sze' CVPR 17]



- Cooperative inference tasks at multiple APs:
  - *Computation replication:* high compute power
  - *Cooperative transmission:* low transmit power
- **Solution:**
  - minimize the sum of computation and transmission power consumption

Signal model

- **Proposal:** group sparse beamforming for total power minimization
  - received signal at l-th mobile user: $y_l = \sum_{n=1}^{N}\sum_{k=1}^{K} \boldsymbol{h}_{nl}^{\mathsf{H}} \boldsymbol{v}_{nk} s_k + z_l = \sum_{k=1}^{K} \boldsymbol{h}_l^{\mathsf{H}} \boldsymbol{v}_k s_k + z_l,$
  - beamforming vector for $\phi_k(d_k)$ at the n-th AP: $\boldsymbol{v}_{nk} \in \mathbb{C}^L$
  - group sparse aggregative beamforming vector

$$\boldsymbol{v} = [\underbrace{\boldsymbol{v}_{11}^{\mathsf{H}}, \cdots, \boldsymbol{v}_{N1}^{\mathsf{H}}}_{\boldsymbol{v}_1}, \cdots, \underbrace{\boldsymbol{v}_{1k}^{\mathsf{H}}, \cdots, \boldsymbol{v}_{Nk}^{\mathsf{H}}}_{\boldsymbol{v}_k}, \cdots, \boldsymbol{v}_{NK}^{\mathsf{H}}]^{\mathsf{H}}, \quad \mathcal{T}(\boldsymbol{v}) = \{(n,k) | \boldsymbol{v}_{nk} \neq \boldsymbol{0}\}$$

  - if $\boldsymbol{v}_{nk}$ is set as zero, task $\phi_k(d_k)$ will not be performed at the n-th AP
  - the signal-to-interference-plus-noise-ratio (SINR) for users

$$\mathrm{SINR}_k(\boldsymbol{v}; \boldsymbol{h}_k) = \frac{|\boldsymbol{h}_k^{\mathsf{H}} \boldsymbol{v}_k|^2}{\sum_{l \neq k} |\boldsymbol{h}_k^{\mathsf{H}} \boldsymbol{v}_l|^2 + \sigma_k^2}$$

Probabilistic group sparse beamforming

- **Goal:** total power consumption under probabilistic QoS constraints

$$\begin{aligned} \mathscr{P}: \min_{\boldsymbol{v}\in\mathbb{C}^{NKL}} \quad & \sum_{n,k}\frac{1}{\eta_n}\|\boldsymbol{v}_{nk}\|_2^2+\sum_{n,k}P_{nk}^{\mathrm{c}}I_{(n,k)\in\mathcal{T}(\boldsymbol{v})} \\ \text{s.t.} \quad & \Pr\left(\mathrm{SINR}_k(\boldsymbol{v};\boldsymbol{h}_k)\geq\gamma_k\right)\geq 1-\zeta, k\in[K] \\ & \sum_{k=1}^{K}\|\boldsymbol{v}_{nk}\|_2^2\leq P_n^{\mathrm{Tx}}, n\in[N]. \end{aligned}$$

transmission and computation power consumption

(maximum transmit power)

- Channel state information (CSI) uncertainty
  - Additive error: $\boldsymbol{h}_k=\hat{\boldsymbol{h}}_k+\boldsymbol{e}_k$, $\mathbb{E}[\boldsymbol{e}_k]=\mathbf{0}$
  - Limited precision of feedback, delays in CSI acquisition...
- **Challenges:** 1) group sparse objective function; 2) probabilistic QoS constraints

Probabilistic QoS constraints

- **General idea:** obtaining D independent samples of the random channel coefficient vector $\boldsymbol{h}_k$; find a solution such that the confidence level of

$$\Pr\left(\mathrm{SINR}_k(\boldsymbol{v};\boldsymbol{h}_k) \geq \gamma_k\right) \geq 1-\epsilon$$

  is no less than $1-\delta$.

- **Limitations of existing methods:**
  - **Scenario generation (SG):**
    - too conservative, performance deteriorates when samples size D increases
    - required sample size $\sum_{i=1}^{NKL-1}\binom{D}{i}\epsilon^i(1-\epsilon)^{D-i} \leq \delta$
  - **Stochastic Programming:**
    - High computation cost, increasing linearly with sample size D
    - No available statistical guarantee

Statistical learning for robust optimization

- **Proposal:** statistical learning based robust optimization approximation
  - constructing a **high probability region** $\mathcal{U}_k$ such that

    $\Pr(\boldsymbol{h}_k \in \mathcal{U}_k) \geq 1-\epsilon$ with confidence at least $1-\delta$
  - imposing target SINR constraints for all elements in high probability region

    $$\mathrm{SINR}_k(\boldsymbol{v};\boldsymbol{h}_k) \geq \gamma_k, \boldsymbol{h}_k \in \mathcal{U}_k$$

- **Statistical learning method for constructing** $\mathcal{U}_k$
  - ellipsoidal uncertainty sets $\mathcal{U}_k = \{\boldsymbol{h}_k : (\boldsymbol{h}_k - \hat{\boldsymbol{h}}_k)^T \boldsymbol{\Sigma}_k^{-1} (\boldsymbol{h}_k - \hat{\boldsymbol{h}}_k) \leq s_k\}$
  - split dataset into two parts $\mathcal{D}^1 = \{\tilde{\boldsymbol{h}}^{(1)}, \cdots, \tilde{\boldsymbol{h}}^{(D_1)}\}$ $\mathcal{D}^2 = \{\tilde{\boldsymbol{h}}^{(D_1+1)}, \cdots, \tilde{\boldsymbol{h}}^{(D)}\}$
  - **Shape learning:** $\hat{\boldsymbol{h}}_k$ sample mean and $\boldsymbol{\Sigma}_k$ sample variance of $\mathcal{D}^1$

    (omitting the correlation between $\boldsymbol{h}_{kn}$, $\boldsymbol{\Sigma}_k$ becomes block diagonal)

Statistical learning for robust optimization

- **Statistical learning method for constructing $\mathcal{U}_k$**
  - size calibration via quantile estimation for s_k
  - compute the function value $\mathcal{G}(\xi)=(\xi-\hat{\boldsymbol{h}}_k)^T\boldsymbol{\Sigma}_k^{-1}(\xi-\hat{\boldsymbol{h}}_k)$ with respect to each sample in $\mathcal{D}^2=\{\tilde{\boldsymbol{h}}^{(D_1+1)},\cdots,\tilde{\boldsymbol{h}}^{(D)}\}$, set s_k as the $j^\star$-th largest value

$$j^\star=\min_{1\le j\le D-D_1}\left\{j:\sum_{k=0}^{j-1}\binom{D-D_1}{k}(1-\epsilon)^k\epsilon^{D-D_1-k}\ge 1-\delta\right\}$$

  - required sample size: $D>\log\delta/\log(1-\epsilon)$

- **Tractable reformulation**

$$\boxed{\Pr\left(\mathrm{SINR}_k(\boldsymbol{v};\boldsymbol{h}_k)\ge\gamma_k\right)\ge 1-\zeta \longrightarrow \boldsymbol{H}_k^{\mathsf{H}}\left(\frac{1}{\gamma_k}\boldsymbol{v}_k\boldsymbol{v}_k^{\mathsf{H}}-\sum_{l\ne k}\boldsymbol{v}_l\boldsymbol{v}_l^{\mathsf{H}}\right)\boldsymbol{H}_k\succeq\boldsymbol{Q}_k,\lambda_k\ge 0}$$

$$\boldsymbol{H}_k=\left[\hat{\boldsymbol{h}}_k\ \sqrt{s_k}\boldsymbol{\Delta}_k\right],\boldsymbol{\Sigma}_k=\boldsymbol{\Delta}_k\boldsymbol{\Delta}_k^{\mathsf{H}}\qquad \boldsymbol{Q}_k=\begin{bmatrix}\lambda_k+\sigma_k^2 & \boldsymbol{0}\\ \boldsymbol{0} & -\lambda_k\boldsymbol{I}_{NL}\end{bmatrix}$$

Robust optimization reformulation

- Tractable reformulation for robust optimization with S-Lemma

$$
\begin{aligned}
\mathscr{P}_{\mathrm{RGS}}: \underset{\boldsymbol{v}\in\mathbb{C}^{NKL},\boldsymbol{\lambda}\in\mathbb{R}^K}{\text{minimize}} \quad & \sum_{n,l}\frac{1}{\eta_n}\|\boldsymbol{v}_{nl}\|_2^2+\sum_{n,l}P_{nl}^{\mathrm{c}}I_{(n,l)\in\mathcal{T}(\boldsymbol{v})} \\
\text{subject to} \quad & \boldsymbol{H}_k^{\mathsf{H}}\left(\frac{1}{\gamma_k}\boldsymbol{v}_k\boldsymbol{v}_k^{\mathsf{H}}-\sum_{l\neq k}\boldsymbol{v}_l\boldsymbol{v}_l^{\mathsf{H}}\right)\boldsymbol{H}_k\succeq\boldsymbol{Q}_k, \lambda_k\geq 0, \forall k\in[K] \\
& \sum_{l=1}^{K}\|\boldsymbol{v}_{nl}\|_2^2\leq P_n^{\mathrm{Tx}}, \forall n\in[N].
\end{aligned}
$$

- **Challenges**
  - ➢ group sparse objective function
  - ➢ nonconvex quadratic constraints

Low-rank matrix optimization

- **Idea:** matrix lifting for nonconvex quadratic constraints

$$\boldsymbol{V}_{ij} = \begin{bmatrix} \boldsymbol{V}_{ij}[1,1] & \cdots & \boldsymbol{V}_{ij}[1,N] \\ \vdots & \ddots & \vdots \\ \boldsymbol{V}_{ij}[N,1] & \cdots & \boldsymbol{V}_{ij}[N,N] \end{bmatrix} = \boldsymbol{v}_i \boldsymbol{v}_j^{\mathsf{H}} \in \mathbb{C}^{NL \times NL}, \quad \boldsymbol{V} = \boldsymbol{v}\boldsymbol{v}^{\mathsf{H}} = \begin{bmatrix} \boldsymbol{V}_{11} & \cdots & \boldsymbol{V}_{1K} \\ \vdots & \ddots & \vdots \\ \boldsymbol{V}_{K1} & \cdots & \boldsymbol{V}_{KK} \end{bmatrix} \in \mathbb{S}_+^{NKL}$$

- Matrix optimization with rank-one constraint

$$\begin{aligned} \underset{\boldsymbol{V},\boldsymbol{\lambda}}{\text{minimize}} \quad & \sum_{n,l} \left(\frac{1}{\eta_n} \mathrm{Tr}(\boldsymbol{V}_{ll}[n,n]) + P_{nl}^{\mathrm{c}} I_{\mathrm{Tr}(\boldsymbol{V}_{ll}[n,n]) \neq 0} \right) \\ \text{subject to} \quad & \boldsymbol{H}_k^{\mathsf{H}} \left(\frac{1}{\gamma_k} \boldsymbol{V}_{kk} - \sum_{l \neq k} \boldsymbol{V}_{ll} \right) \boldsymbol{H}_k \succeq \boldsymbol{Q}_k, \lambda_k \geq 0, \forall k \in [K] \\ & \sum_{l=1}^{K} \mathrm{Tr}(\boldsymbol{V}_{ll}[n,n]) \leq P_n^{\mathrm{Tx}}, \forall n \in [N] \\ & \boldsymbol{V} \succeq \boldsymbol{0}, \mathrm{rank}(\boldsymbol{V}) = 1. \end{aligned}$$

Reweighted power minimization approach

- **Sparsity:** reweighted ℓ_1-minimization for inducing group sparsity
  - Approximation: $I_{\mathrm{Tr}(\boldsymbol{V}_{ll}[n,n])\neq 0} \approx w_{nl}\mathrm{Tr}(\boldsymbol{V}_{ll}[n,n])$, $w_{nl} = \frac{c}{\mathrm{Tr}(\boldsymbol{V}_{ll}[n,n])+\tau}$
  - Alternatively optimizing $\boldsymbol{V}$ and updating weights w_{nl}

- **Low-rankness:** DC representation for rank-one positive semidefinite matrix

$$\mathrm{rank}(\boldsymbol{M}) = 1 \Leftrightarrow \mathrm{Tr}(\boldsymbol{M}) - \|\boldsymbol{M}\|_2 = 0$$

  - where $\mathrm{Tr}(\boldsymbol{M}) = \sum_{i=1}^{N} \sigma_i(\boldsymbol{M})$ and $\|\boldsymbol{M}\|_2 = \sigma_1(\boldsymbol{M})$

Reweighted power minimization approach

- **Updating $\boldsymbol{V}$**

$$\mathscr{P}_{\mathrm{DC}}: \underset{\boldsymbol{V},\boldsymbol{\lambda}}{\text{minimize}} \quad \sum_{n,l}\left(\frac{1}{\eta_n}+w_{nl}^{[j]}P_{nl}^{\mathrm{c}}\right)\mathrm{Tr}(\boldsymbol{V}_{ll}[n,n])+\mu(\mathrm{Tr}(\boldsymbol{V})-\|\boldsymbol{V}\|_2)$$

$$\text{subject to} \quad \boldsymbol{H}_k^{\mathsf{H}}\left(\frac{1}{\gamma_k}\boldsymbol{V}_{kk}-\sum_{l\neq k}\boldsymbol{V}_{ll}\right)\boldsymbol{H}_k \succeq \boldsymbol{Q}_k, \lambda_k \geq 0, \forall k \in [K]$$

$$\sum_{l=1}^{K}\mathrm{Tr}(\boldsymbol{V}_{ll}[n,n]) \leq P_n^{\mathrm{Tx}}, \forall n \in [N]$$

$$\boldsymbol{V} \succeq \boldsymbol{0},$$

updating w_{nl}

$$w_{nl} = \frac{c}{\mathrm{Tr}(\boldsymbol{V}_{ll}[n,n])+\tau}$$

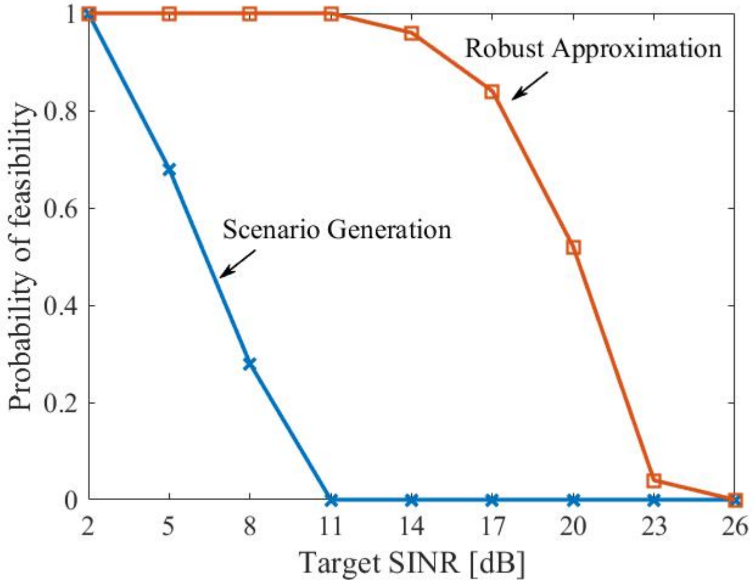
- The DC algorithm via iteratively linearizing the concave part

$$-\|\boldsymbol{V}\|_2 \longrightarrow -\langle \partial\|\boldsymbol{V}\|_2, \boldsymbol{V}\rangle, \partial\|\boldsymbol{V}\|_2 = \boldsymbol{u}_1\boldsymbol{u}_1^{\mathsf{H}}$$

  - $\boldsymbol{u}_1$: the eigenvector corresponding to the largest eigenvalue of $\boldsymbol{V}$

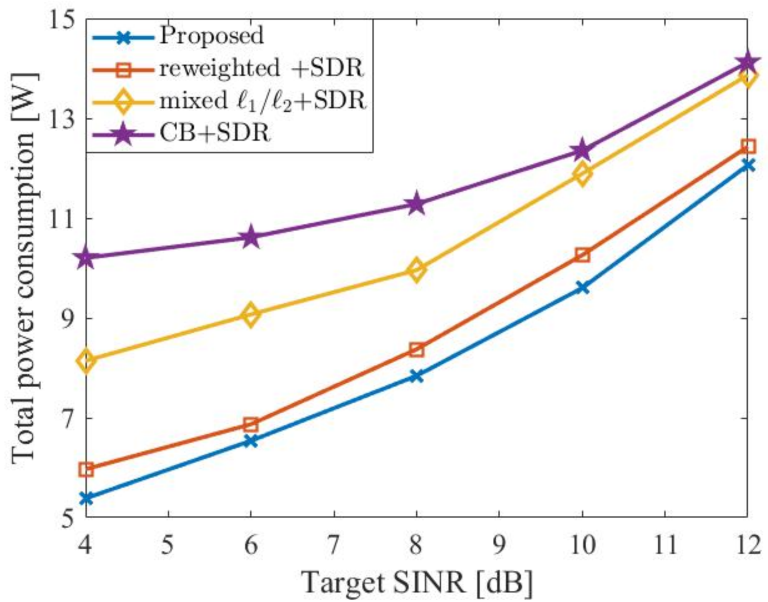
Numerical results

- Performance of our robust optimization approximation approach and scenario generation



Numerical results

- Energy-efficient processing and robust wireless cooperative transmission for executing inference tasks at possibly multiple edge computing nodes



Insights on edge inference:

1. Selecting the optimal set of access points for each inference task via group sparse beamforming
2. A robust optimization approach for joint chance constraints via statistical learning to learn CSI uncertainty set

Concluding remarks

- **Machine learning model inference over wireless networks**
  - On-device inference via wireless distributed computing
  - Edge inference via computation replication and cooperative transmission
- **Sparse and low-rank optimization framework**
  - Inference alignment for data shuffling in wireless MapReduce
  - Joint inference tasking and downlink beamforming for edge inference
- **Nonconvex optimization frameworks**
  - DC algorithm for generalized low-rank matrix optimization
  - Statistical learning for stochastic robust optimization

Future directions

- **On-device distributed inference**
  - model compression, energy efficient inference, full duplex,…
- **Edge cooperative inference**
  - hierarchical inference over cloud-edge-device, low-latency, …
- **Nonconvex optimization via DC and learning approaches**
  - optimality, scalability, applicability, …

Mobile Edge Artificial Intelligence: Opportunities and Challenges

Part III: Training

Yuanming Shi

ShanghaiTech University

上海科技大学
ShanghaiTech University

Outline

- **Motivations**
  - Privacy, federated learning
- **Two vignettes:**
  - **Over-the-air computation for federated learning**
    - Why over-the-air computation?
    - Joint device selection and beamforming design
  - **Intelligent reflecting surface empowered federated learning**
    - Why intelligent reflecting surface?
    - Joint phase shifts and transceiver design

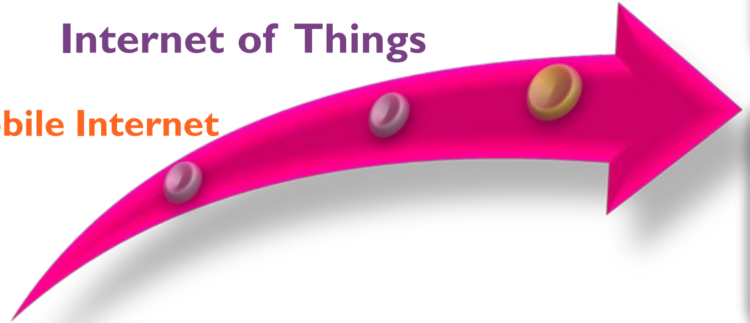
Intelligent IoT ecosystem

(Internet of Skills)

Tactile Internet

Internet of Things

Mobile Internet



Develop computation, communication & AI technologies:
enable smart IoT applications to make ***low-latency decision*** on streaming data

Intelligent IoT applications

Autonomous vehicles

Smart home

Smart city

Smart health

Smart agriculture

Smart drones

Challenges

- Retrieve or infer information from high-dimensional/large-scale data

2.5 exabytes of data
are generated every day (2012)

exabyte → zettabyte → yottabyte...??

We're interested in the ***information*** rather than the data

limited processing ability
(computation, storage, ...)

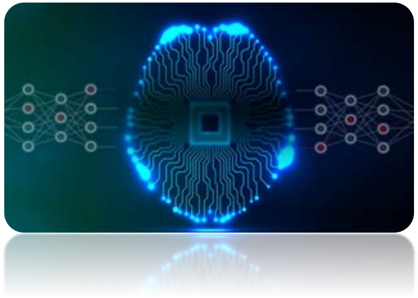
Challenges:

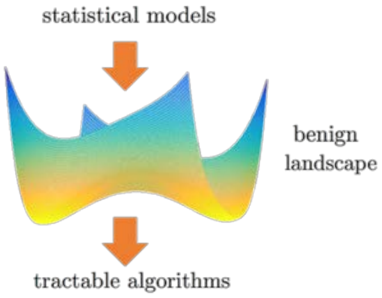
- High computational cost
- Only limited memory is available
- Do NOT want to compromise statistical accuracy

High-dimensional data analysis

(big) data

Models: (deep) machine learning

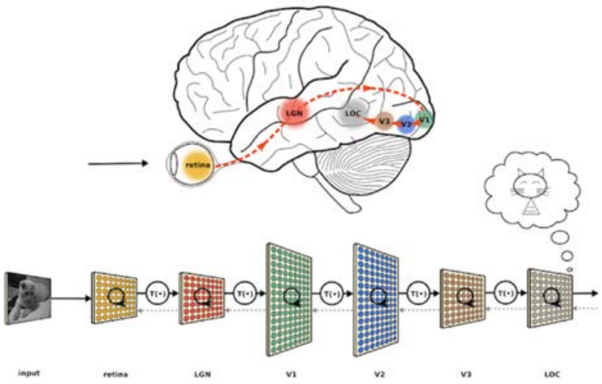




Methods:

1. Large-scale optimization
2. High-dimensional statistics
3. Device-edge-cloud computing

Deep learning: next wave of AI



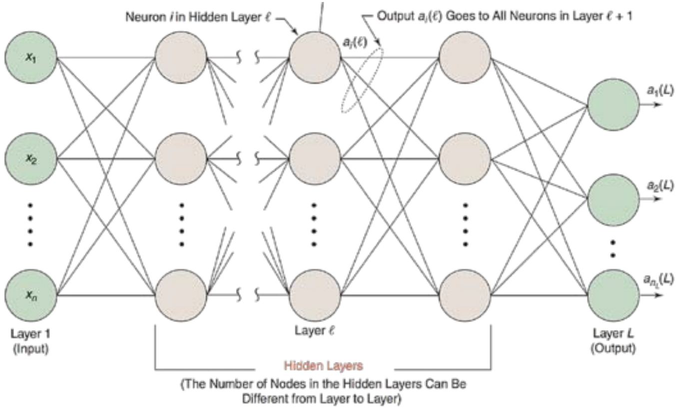


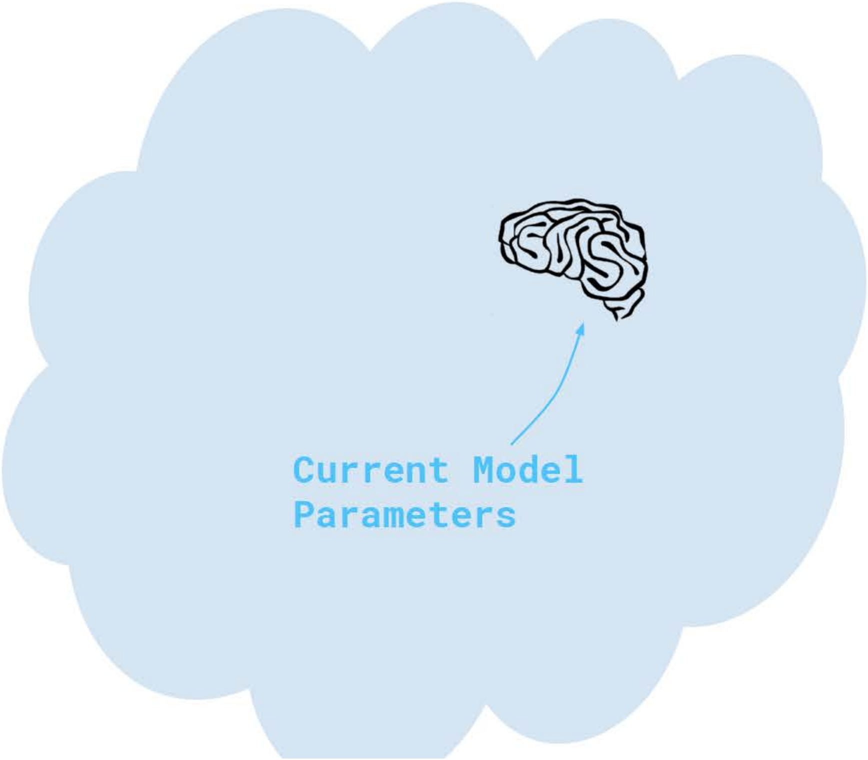
image
recognition

speech
recognition

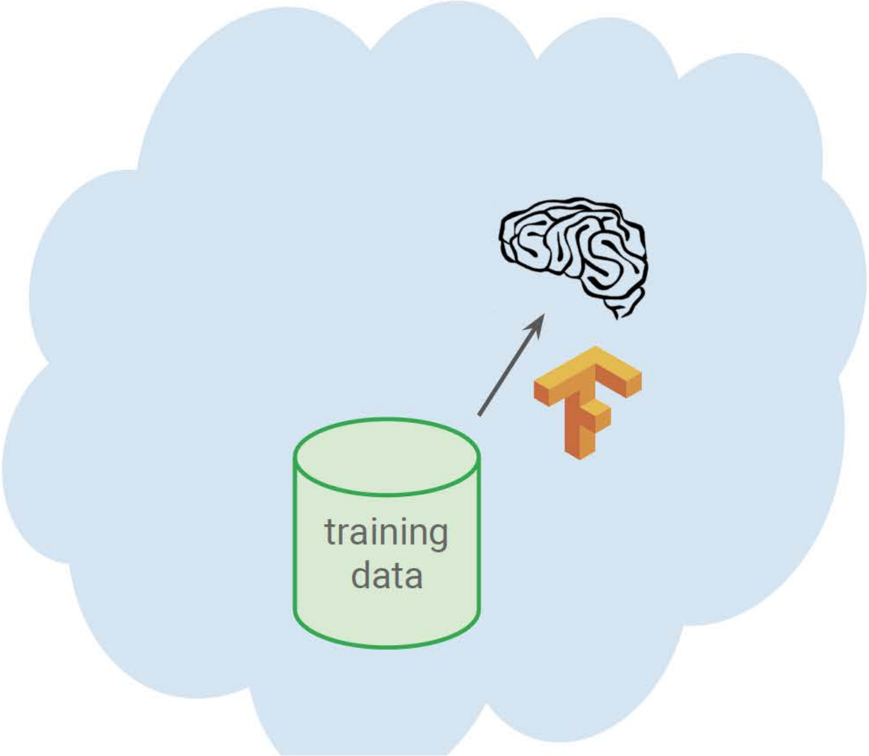
natural language
processing

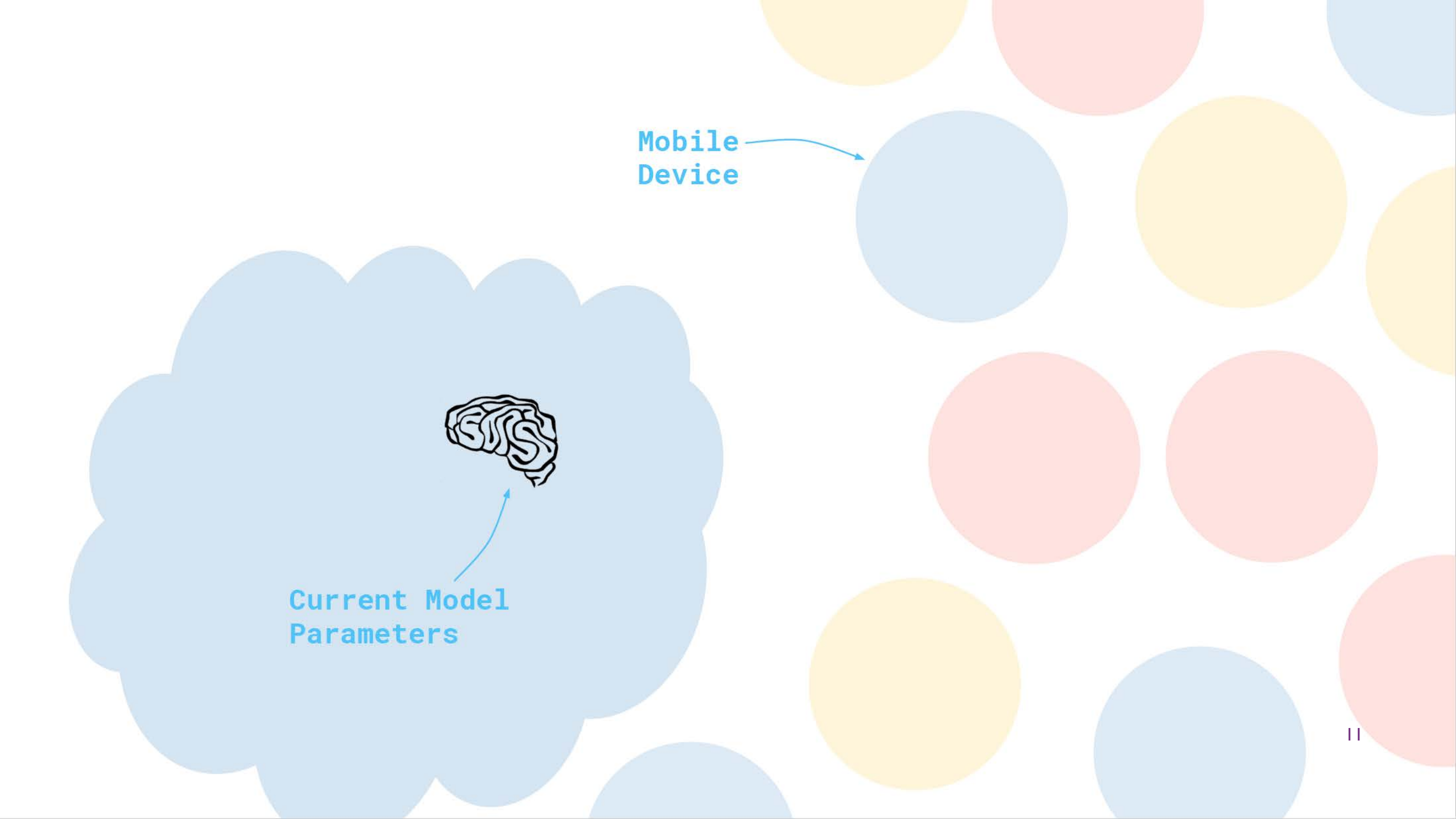
Cloud-centric machine learning

The model lives in the cloud

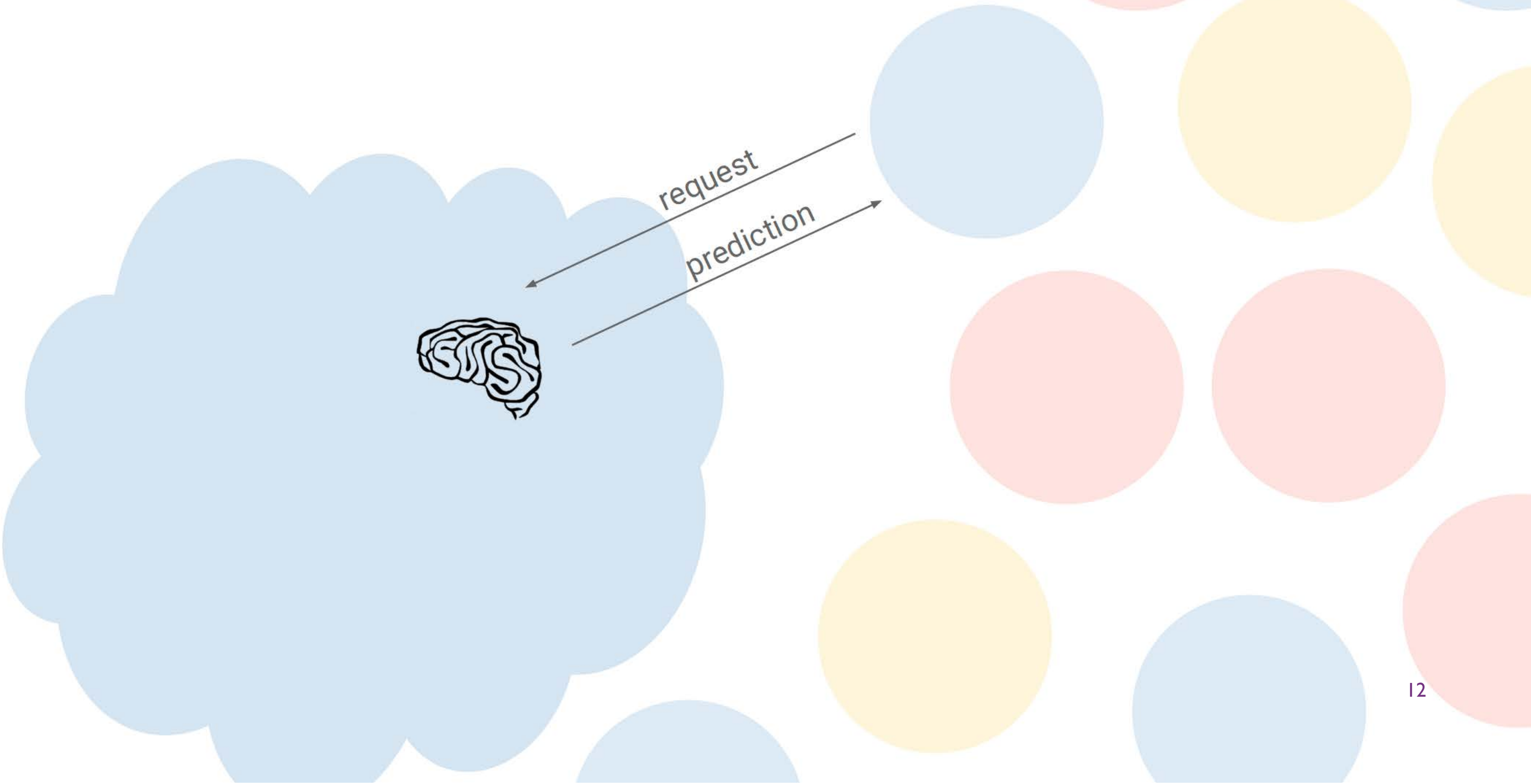


We train models in the cloud

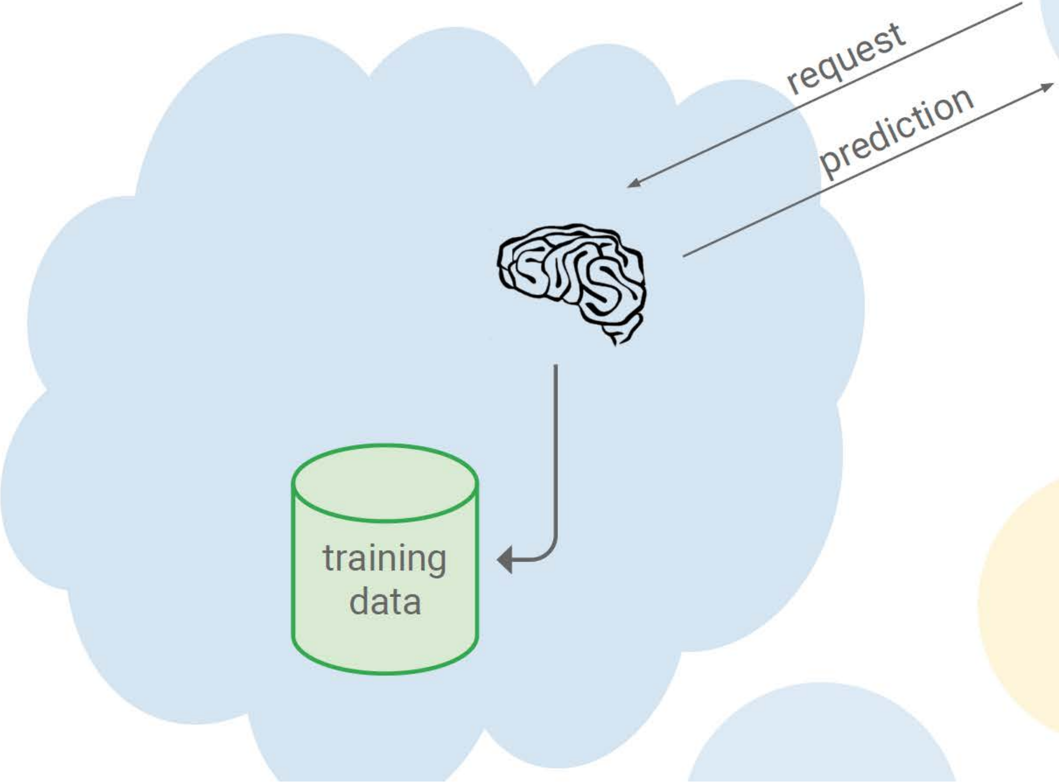



Mobile Device
Current Model Parameters

Make predictions in the cloud



Gather training data in the cloud



And make the models better

Why edge machine learning?

Challenges to modern AI

- **Challenges:** data privacy and confidentiality; small data and fragmented data; data quality and limited labels

Facebook's data privacy scandal

the general data protection regulation (GDPR)

Learning on the edge

- The emerging high-stake AI applications: low-latency, privacy,…

phones

drones

robots

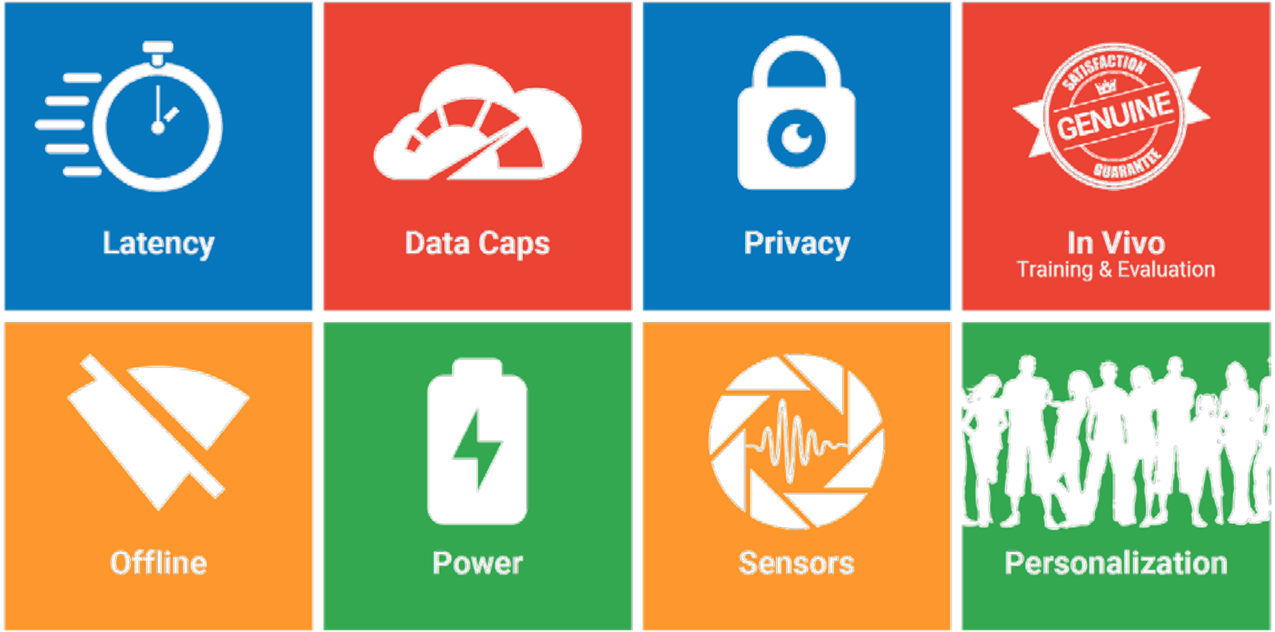
glasses

self driving cars

where to compute?

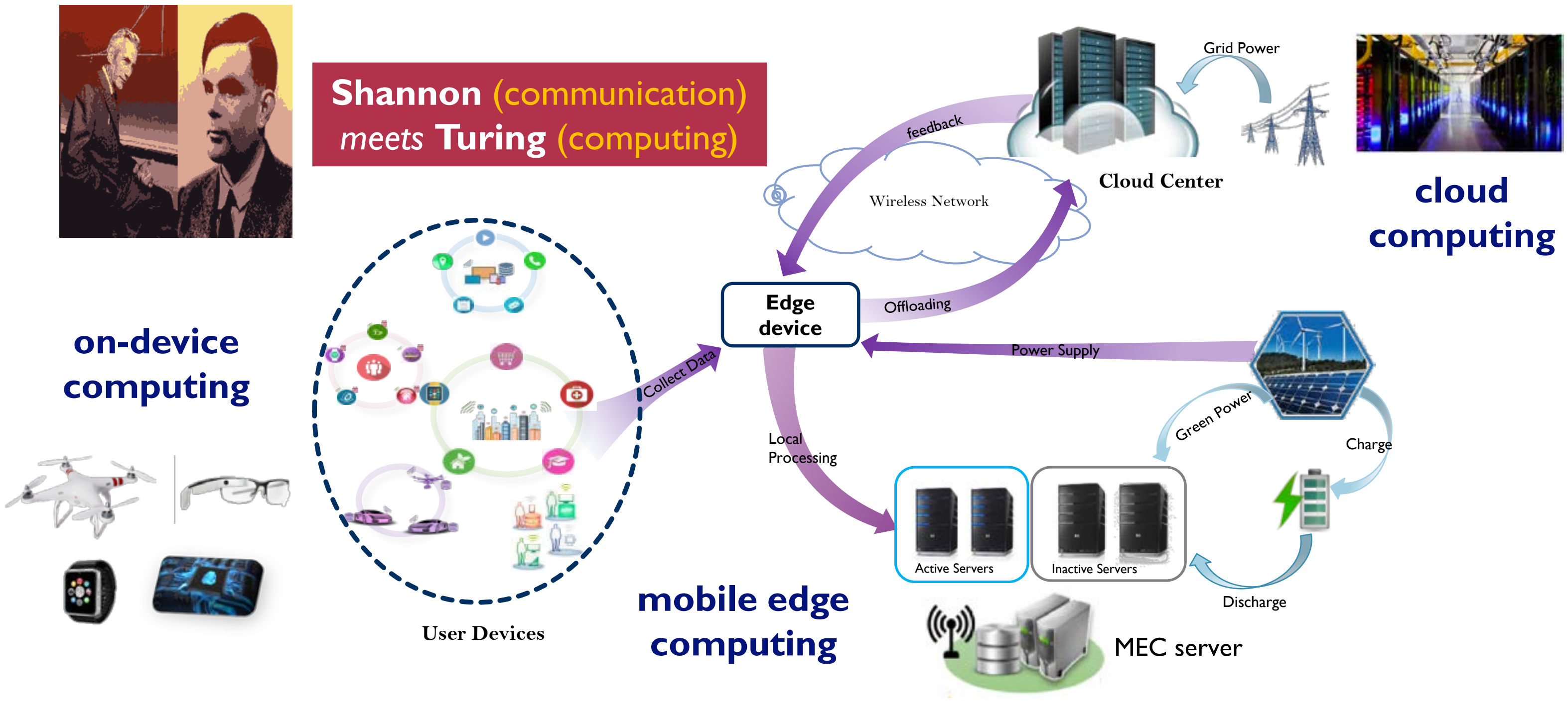
Mobile edge AI

- Processing at "edge" instead of "cloud"



Edge computing ecosystem

- **“Device-edge-cloud”** computing system for mobile AI applications



Edge machine learning

- **Edge ML:** both ML inference and training processes are pushed down into the network edge (bottom)

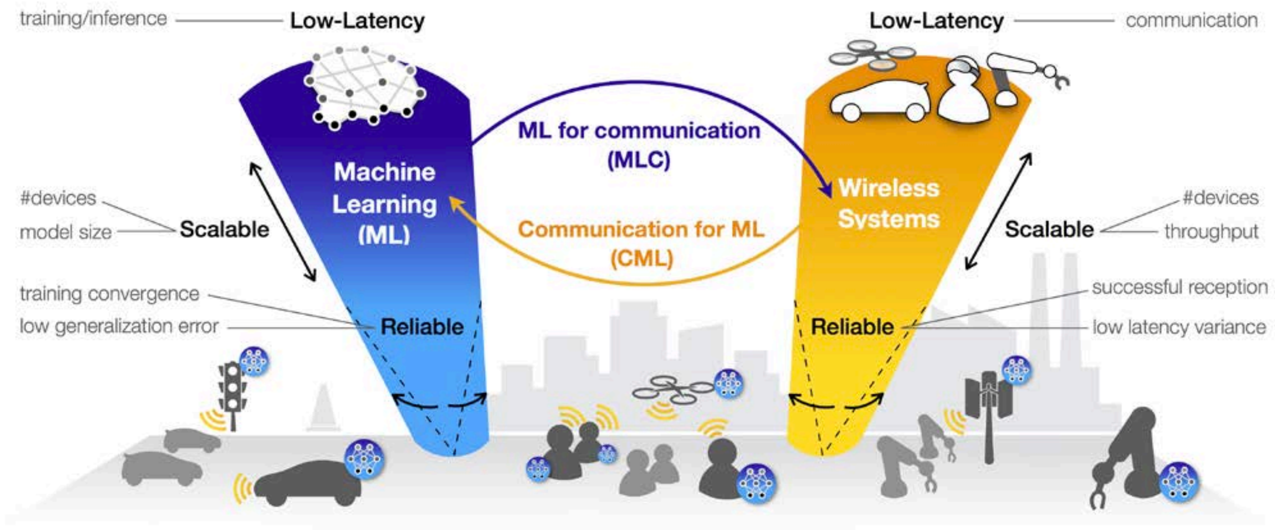
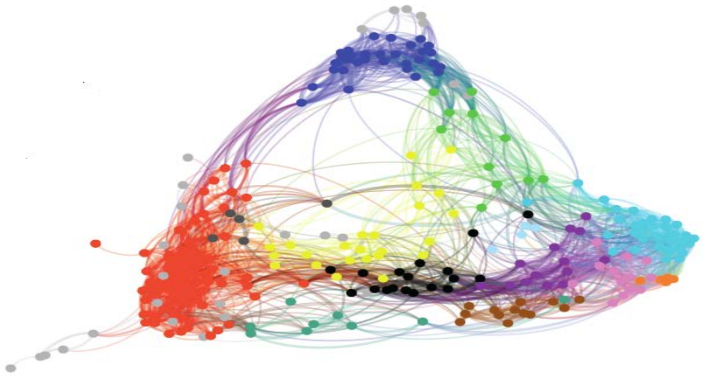


Fig. credit: Park

Vignettes A: Over-the-air computation for federated learning



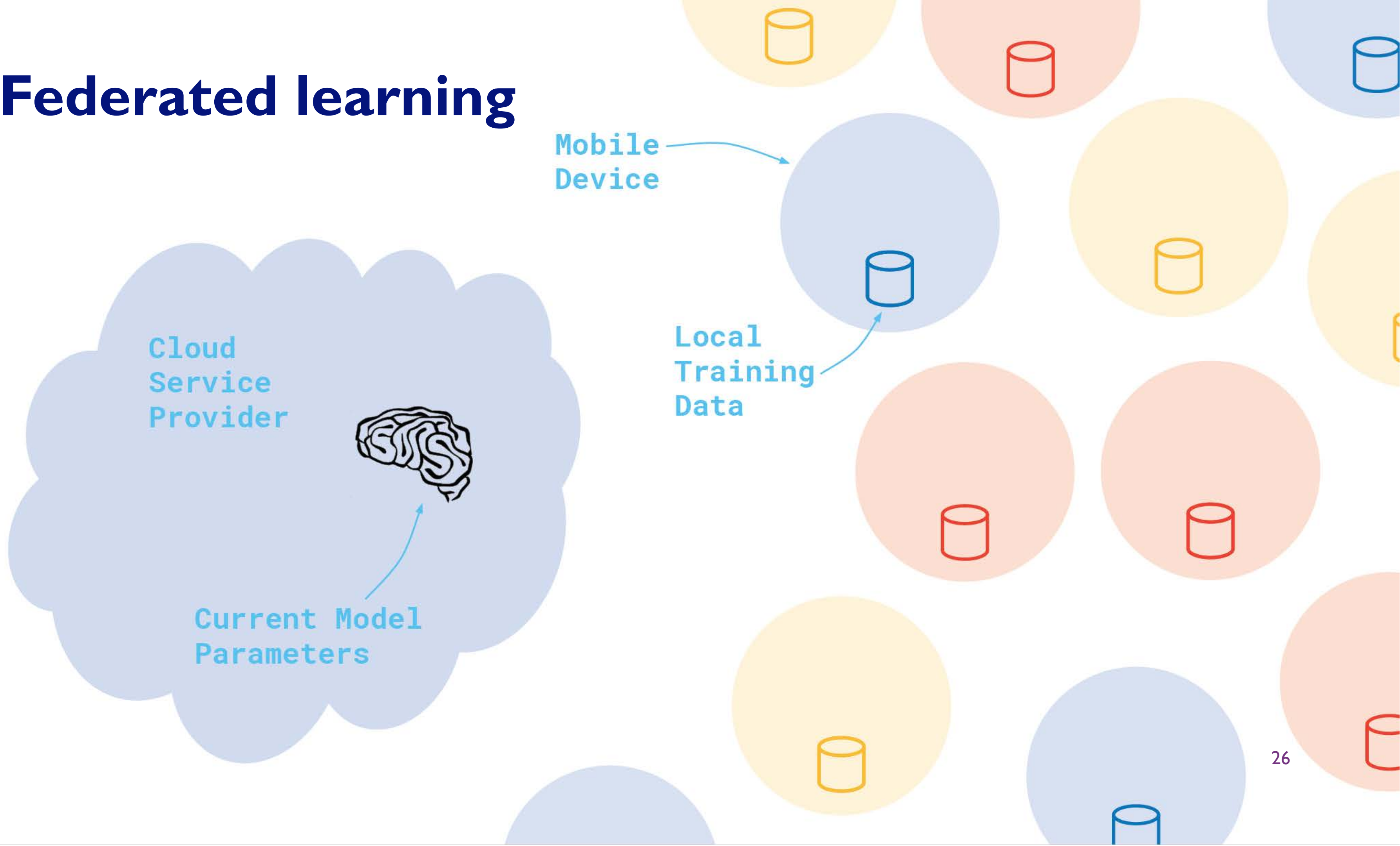
Federated computation and learning

- **Goal:** imbue mobile devices with state of the art machine learning systems ***without centralizing data*** and with ***privacy*** by default

- **Federated computation:** a server coordinates a fleet of participating devices to compute aggregations of devices' private data

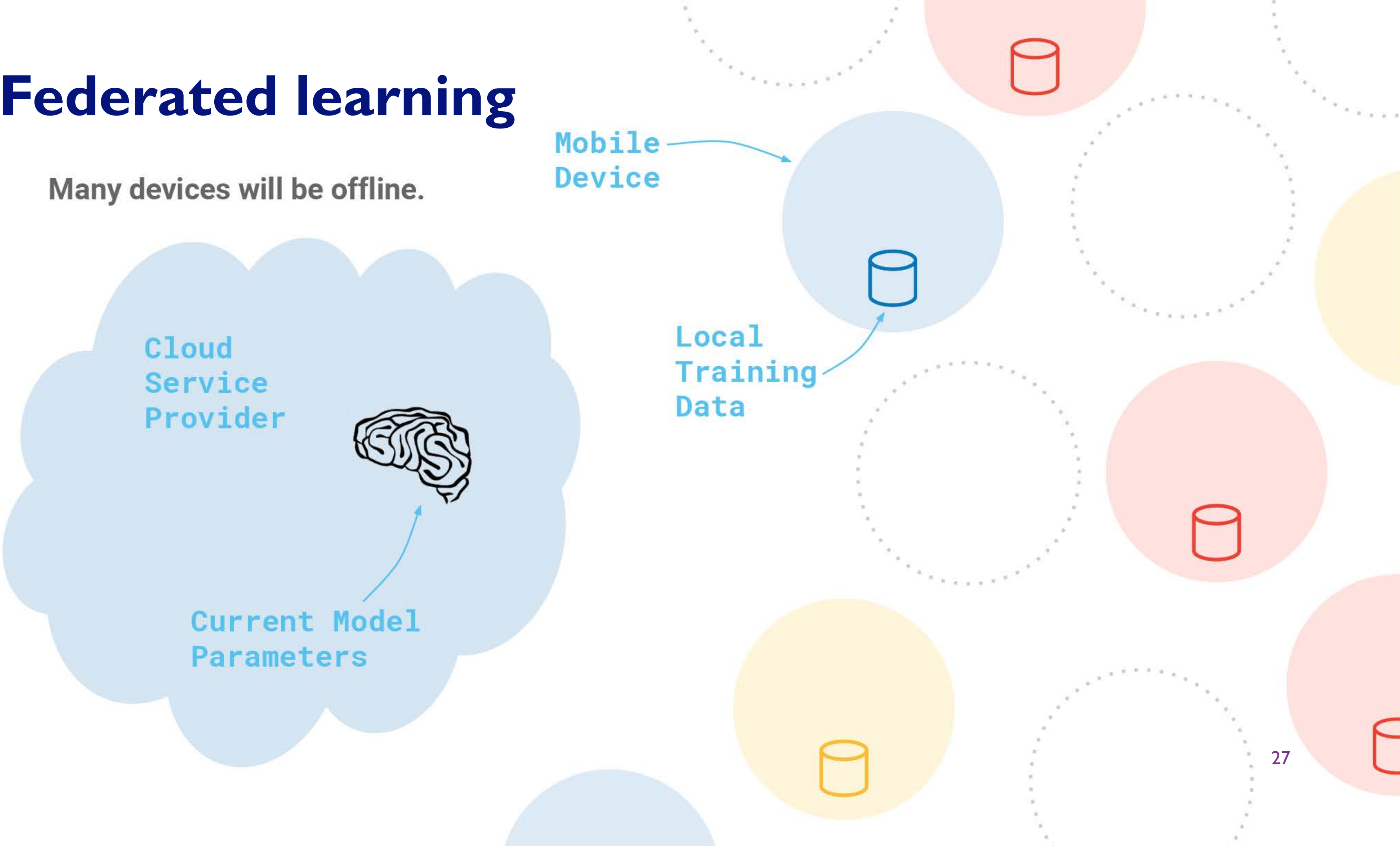
- **Federated learning:** a shared global model is trained via federated computation

Federated learning



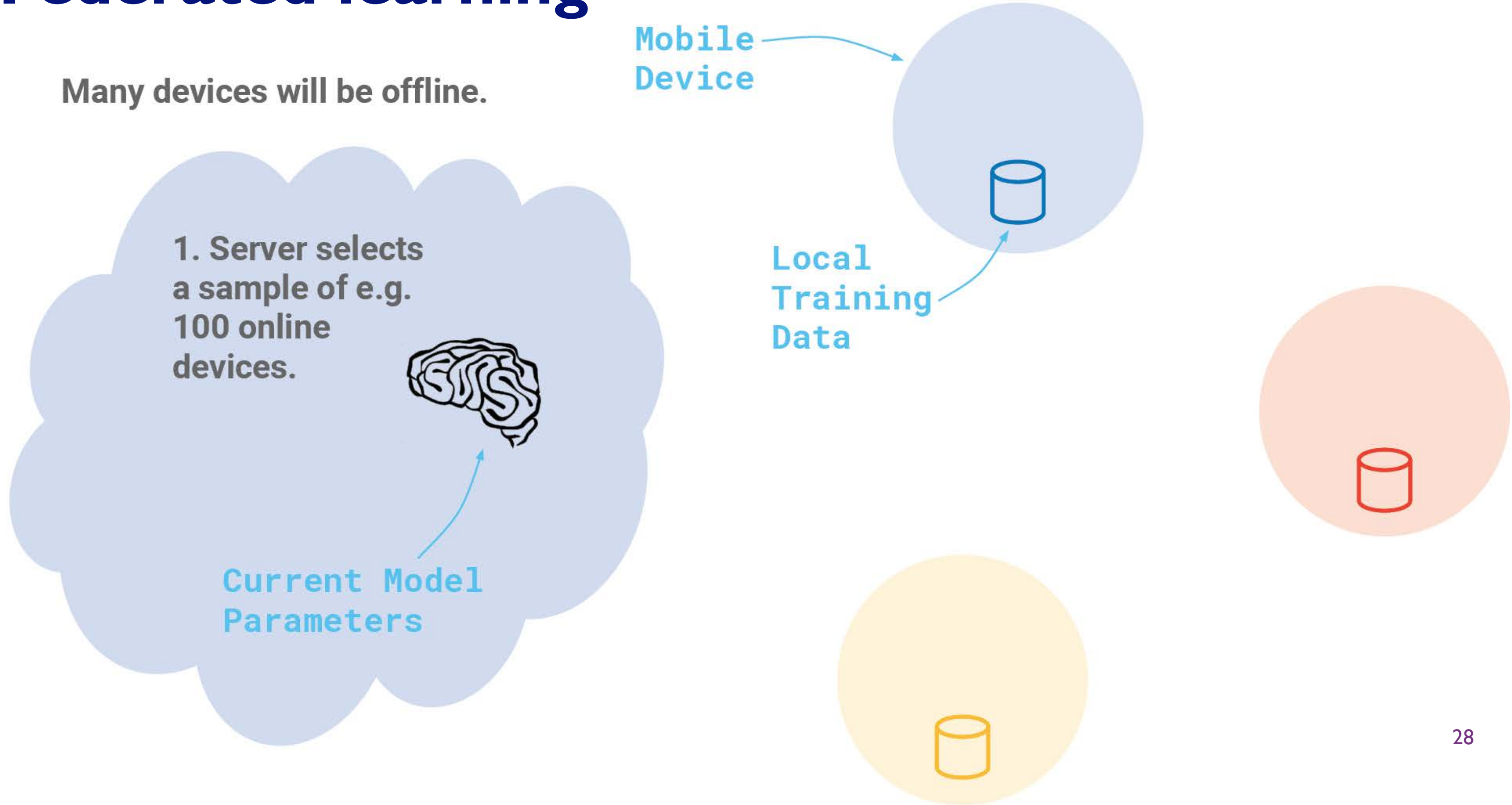
Federated learning

Many devices will be offline.

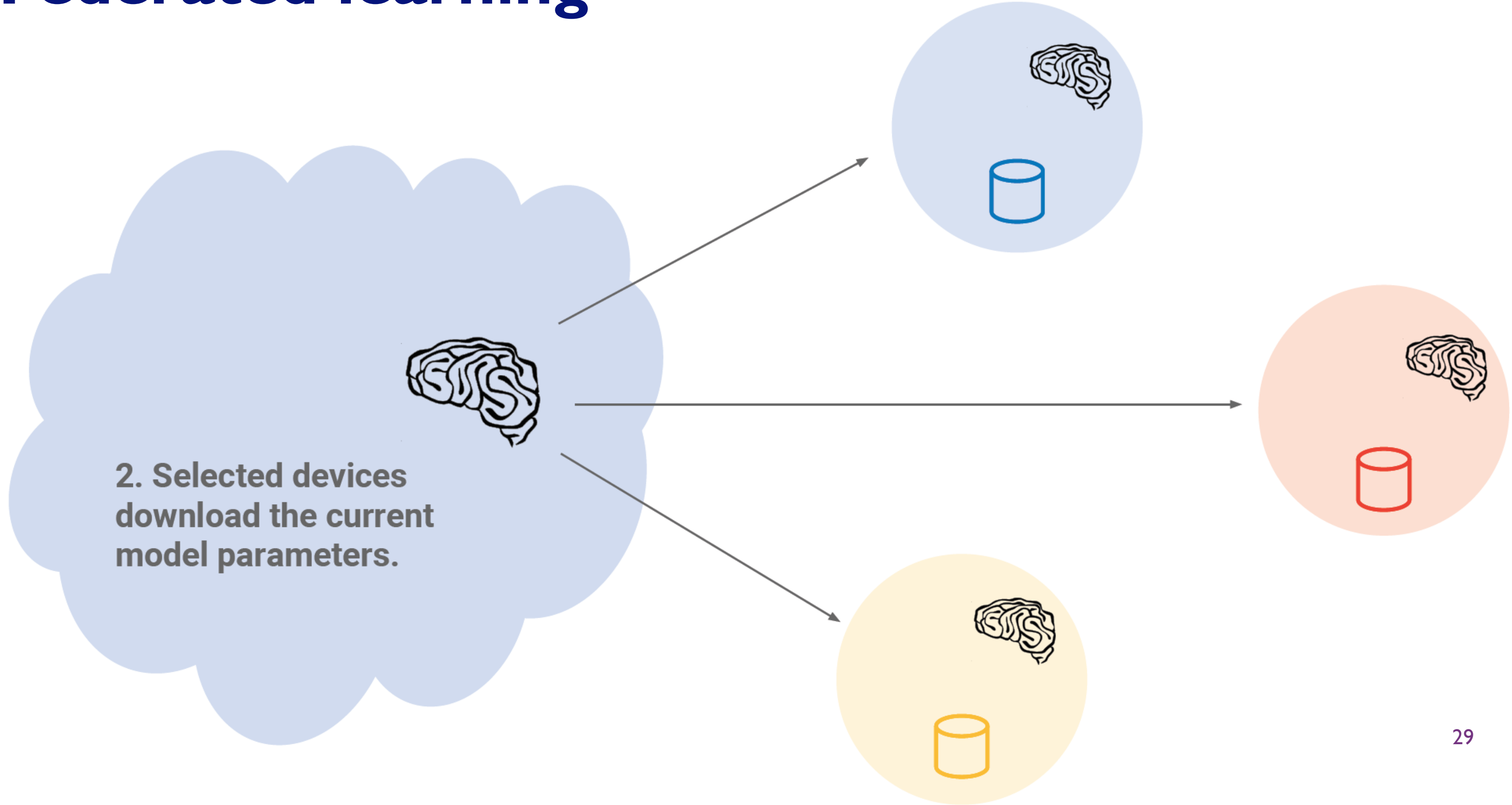


Federated learning

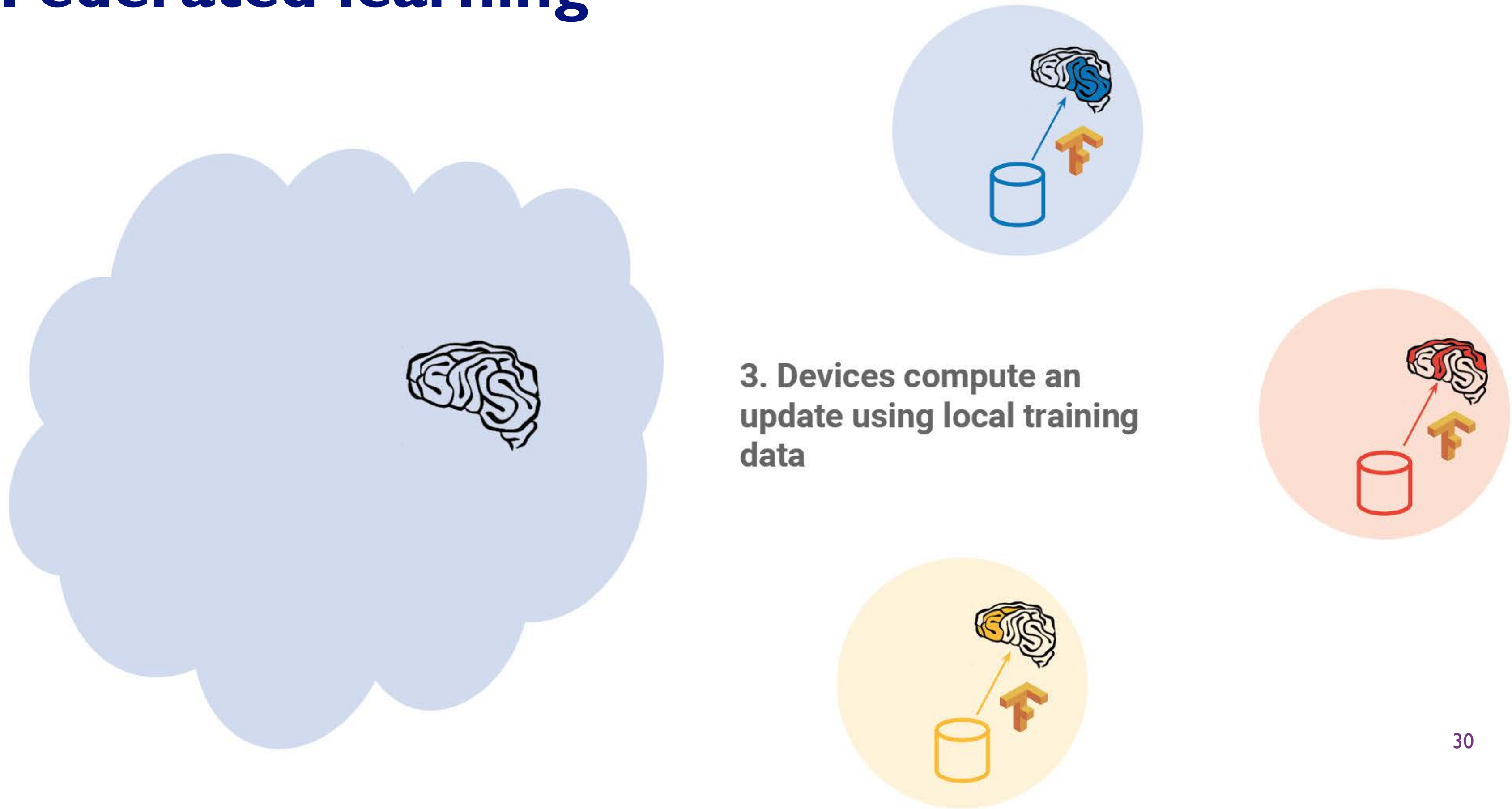
Many devices will be offline.



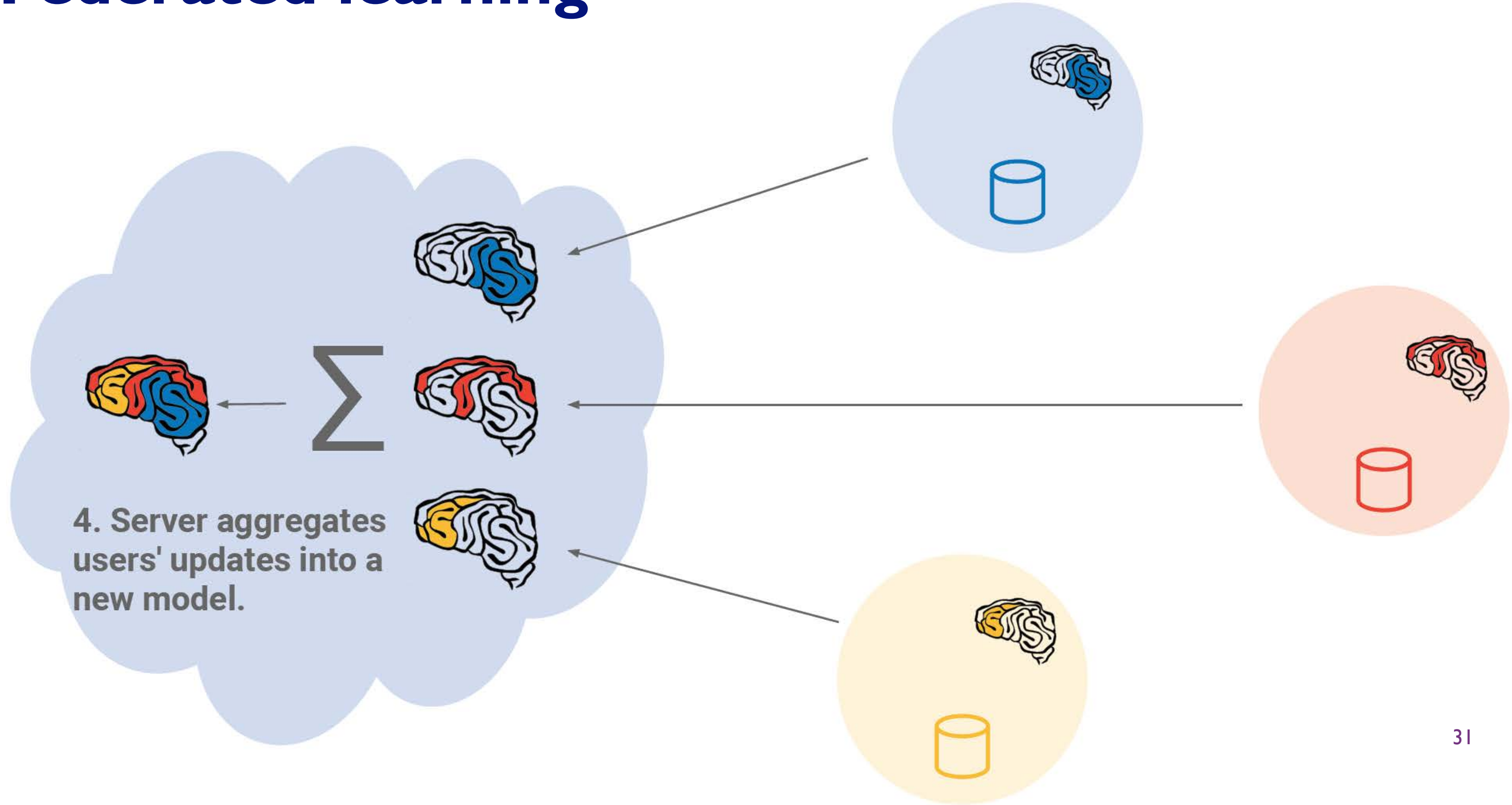
Federated learning



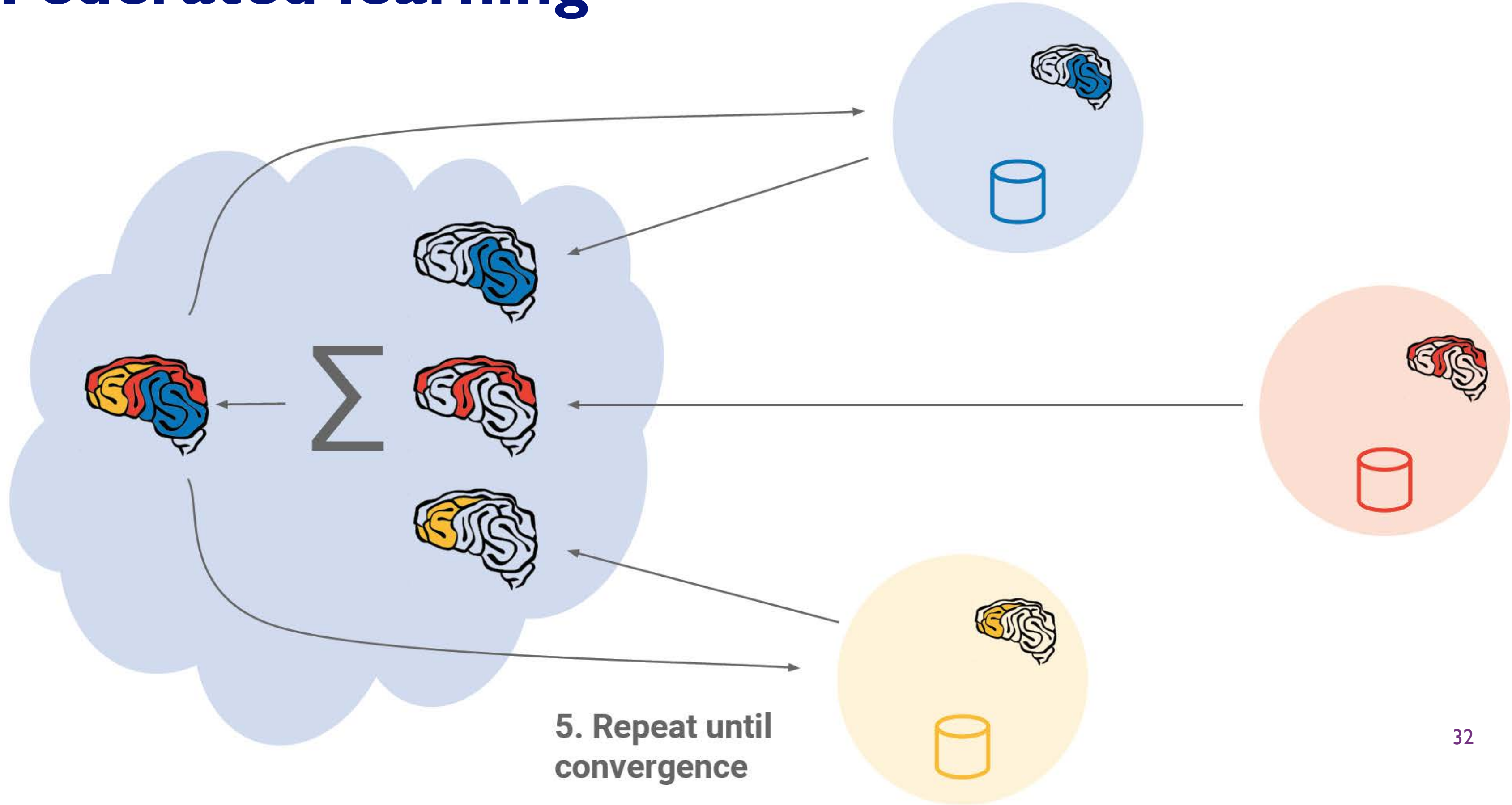
Federated learning



Federated learning



Federated learning



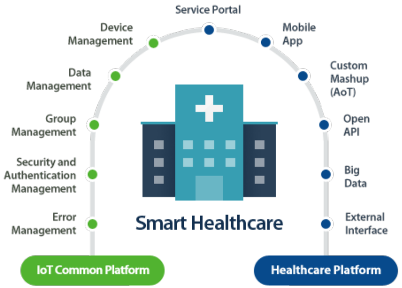
Federated learning: applications

- **Applications:** where the data is generated at the mobile devices and is undesirable/infeasible to be transmitted to centralized servers

financial services

keyboard prediction

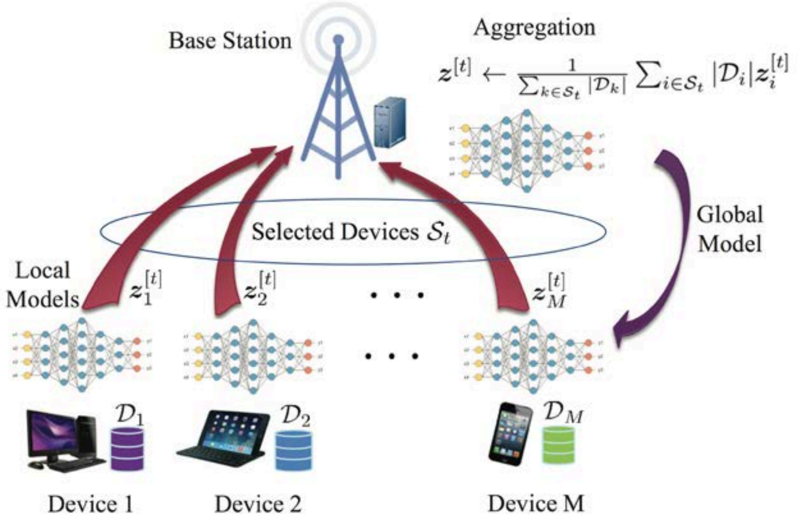
smart retail



smart healthcare

Federated learning over wireless networks

- **Goal:** train a shared global model via ***wireless*** federated computation



on-device distributed federated learning system

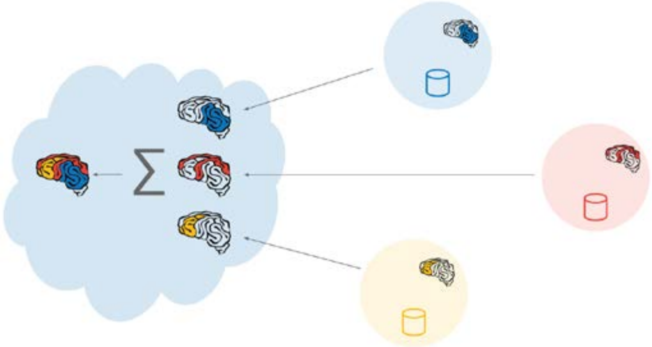
System challenges

- Massively distributed
- Node heterogeneity

Statistical challenges

- Unbalanced
- Non-IID
- Underlying structure

<u>How to efficiently aggregate models over wireless networks?</u>

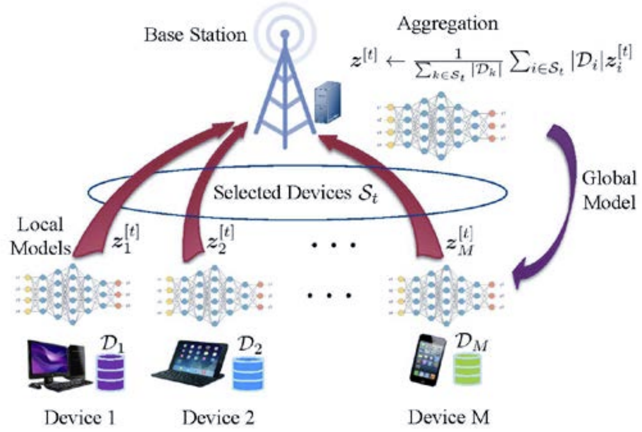


Model aggregation via over-the-air computation

- Aggregating local updates from mobile devices

$$\boxed{z \leftarrow \frac{1}{\sum_{k\in\mathcal{S}}|\mathcal{D}_k|}\sum_{k\in\mathcal{S}}|\mathcal{D}_k|z_k}$$

  - weighted sum of messages
  - M mobile devices and one N-antenna base station
  - $\mathcal{S}\subseteq\{1,\cdots,M\}$ is the set of selected devices
  - $|\mathcal{D}_k|$ is the data size at device k



Over-the-air computation: explore signal superposition of a wireless multiple-access channel for model aggregation

Over-the-air computation

- The estimated value before post-processing at the BS

$$\hat{g} = \frac{1}{\sqrt{\eta}} \boldsymbol{m}^{\mathsf{H}} \boldsymbol{y} = \frac{1}{\sqrt{\eta}} \boldsymbol{m}^{\mathsf{H}} \sum_{i \in \mathcal{S}} \boldsymbol{h}_i b_i z_i + \frac{\boldsymbol{m}^{\mathsf{H}} \boldsymbol{n}}{\sqrt{\eta}}$$

  - b_i is the transmitter scalar, $\boldsymbol{m}$ is the received beamforming vector, η is a normalizing factor
  - target function to be estimated: $g = \sum_{i \in \mathcal{S}} |\mathcal{D}_i| z_i$
  - recovered aggregation vector entry via post-processing: $\hat{z} = \frac{1}{\sum_{i \in \mathcal{S}} |\mathcal{D}_i|} \hat{g}$

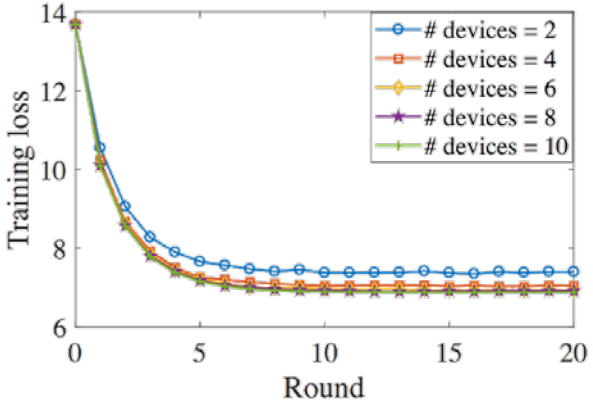
- **Model aggregation error:**

$$\mathsf{MSE}(\hat{g}, g; \mathcal{S}, \boldsymbol{m}) = \frac{\|\boldsymbol{m}\|^2 \sigma^2}{\eta} = \frac{\sigma^2}{P_0} \max_{i \in \mathcal{S}} |\mathcal{D}_i|^2 \frac{\|\boldsymbol{m}\|^2}{\|\boldsymbol{m}^{\mathsf{H}} \boldsymbol{h}_i\|^2}$$

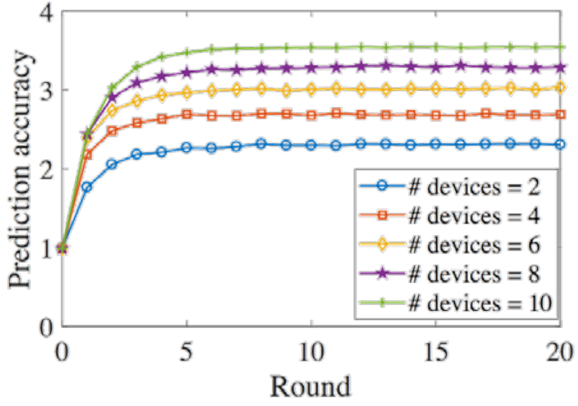
  - Optimal transmitter scalar: $b_i = \sqrt{\eta} |\mathcal{D}_i| \frac{(\boldsymbol{m}^{\mathsf{H}} \boldsymbol{h}_i)^{\mathsf{H}}}{\|\boldsymbol{m}^{\mathsf{H}} \boldsymbol{h}_i\|^2}$

Problem formulation

- **Key observations:**
  - More selected devices yield fast convergence rate of the training process
  - Aggregation error leads to the deterioration of model prediction accuracy



(a) Training loss



(b) Relative prediction accuracy

Problem formulation

- **Goal:** maximize the number of selected devices under target MSE constraint

$$\underset{\mathcal{S}, \boldsymbol{m} \in \mathbb{C}^N}{\text{maximize}} \quad |\mathcal{S}| \qquad \text{subject to} \quad \left(\max_{i \in \mathcal{S}} |\mathcal{D}_i|^2 \frac{\|\boldsymbol{m}\|^2}{\|\boldsymbol{m}^{\mathsf{H}} \boldsymbol{h}_i\|^2} \right) \leq \gamma$$

  - Joint device selection and received beamforming vector design
  - Improve convergence rate in the ***training process***, guarantee prediction accuracy in the ***inference process***
  - **Mixed combinatorial optimization problem**

Sparse and low-rank optimization

- Sparse and low-rank optimization for on-device federated learning

$$\begin{aligned} \underset{\mathcal{S}, \boldsymbol{m} \in \mathbb{C}^N}{\text{maximize}} \quad & |\mathcal{S}| \\ \text{subject to} \quad & \left(\max_{i \in \mathcal{S}} |\mathcal{D}_i|^2 \frac{\|\boldsymbol{m}\|^2}{\|\boldsymbol{m}^{\mathsf{H}} \boldsymbol{h}_i\|^2} \right) \leq \gamma \end{aligned}$$

multicasting duality →

$$\begin{aligned} \underset{\mathcal{S}, \boldsymbol{m} \in \mathbb{C}^N}{\text{maximize}} \quad & |\mathcal{S}| \\ \text{subject to} \quad & \|\boldsymbol{m}\|^2 - \gamma_i \|\boldsymbol{m}^{\mathsf{H}} \boldsymbol{h}_i\|^2 \leq 0, i \in \mathcal{S} \\ & \|\boldsymbol{m}\|^2 \geq 1 \end{aligned}$$

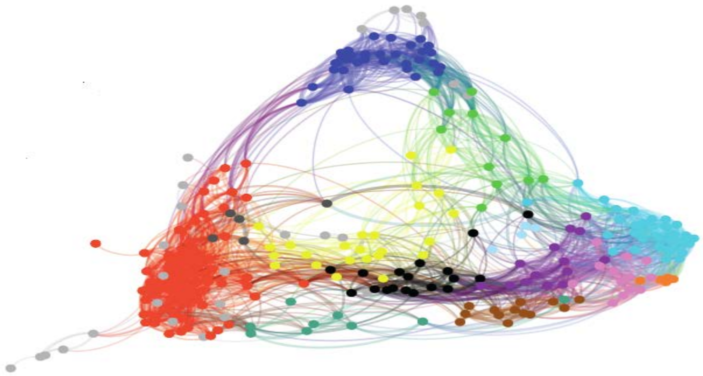
sum of feasibilities ↓

$$\begin{aligned} \underset{\boldsymbol{x} \in \mathbb{R}_+^M, \boldsymbol{m} \in \mathbb{C}^N}{\text{minimize}} \quad & \|\boldsymbol{x}\|_0 \\ \text{subject to} \quad & \|\boldsymbol{m}\|^2 - \gamma_i \|\boldsymbol{m}^{\mathsf{H}} \boldsymbol{h}_i\|^2 \leq x_i, \forall i \\ & \|\boldsymbol{m}\|^2 \geq 1 \end{aligned}$$

matrix lifting ← $\boldsymbol{M} = \boldsymbol{m}\boldsymbol{m}^{\mathsf{H}}$

$$\boxed{\begin{aligned} \mathscr{P}: \underset{\boldsymbol{x} \in \mathbb{R}_+^M, \boldsymbol{M} \in \mathbb{C}^{N \times N}}{\text{minimize}} \quad & \|\boldsymbol{x}\|_0 \\ \text{subject to} \quad & \text{Tr}(\boldsymbol{M}) - \gamma_i \boldsymbol{h}_i^{\mathsf{H}} \boldsymbol{M} \boldsymbol{h}_i \leq x_i, \\ & \boldsymbol{M} \succeq \boldsymbol{0}, \text{Tr}(\boldsymbol{M}) \geq 1 \\ & \text{rank}(\boldsymbol{M}) = 1 \end{aligned}}$$

Sparse and low-rank optimization



Problem analysis

- **Goal:** induce sparsity while satisfying fixed-rank constraint

$$
\begin{aligned}
\mathscr{P}: \underset{\boldsymbol{x}\in\mathbb{R}_{+}^{M}, \boldsymbol{M}\in\mathbb{C}^{N\times N}}{\text{minimize}} \quad & \|\boldsymbol{x}\|_0 \\
\text{subject to} \quad & \mathrm{Tr}(\boldsymbol{M}) - \gamma_i \boldsymbol{h}_i^{\mathsf{H}} \boldsymbol{M} \boldsymbol{h}_i \le x_i, \forall i \\
& \boldsymbol{M} \succeq \mathbf{0}, \mathrm{Tr}(\boldsymbol{M}) \ge 1 \\
& \mathrm{rank}(\boldsymbol{M}) = 1
\end{aligned}
$$

- Limitations of existing methods
  - Sparse optimization: iterative reweighted algorithms are parameters sensitive
  - Low-rank optimization: semidefinite relaxation (SDR) approach (i.e., drop rank-one constraint) has the poor capability of returning rank-one solution

Difference-of-convex functions representation

- Ky Fan k-norm [Fan, PNAS'1951]: the sum of largest-k absolute values

$$\|\boldsymbol{x}\|_k = \sum_{i=1}^{k} |x_{\pi(i)}|$$

  - π is a permutation of $\{1, \cdots, M\}$, where $|x_{\pi(1)}| \geq \cdots \geq |x_{\pi(M)}|$

*MAXIMUM PROPERTIES AND INEQUALITIES FOR THE EIGENVALUES OF COMPLETELY CONTINUOUS OPERATORS**

BY KY FAN

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF NOTRE DAME

Communicated by John von Neumann, September 8, 1951

PNAS'1951

Difference-of-convex functions representation

- DC representation for sparsity function

$$\|\boldsymbol{x}\|_0 = \min\{k : \|\boldsymbol{x}\|_1 - |||\boldsymbol{x}|||_k = 0, 0 \le k \le M\}$$

- DC representation for rank-one positive semidefinite matrix

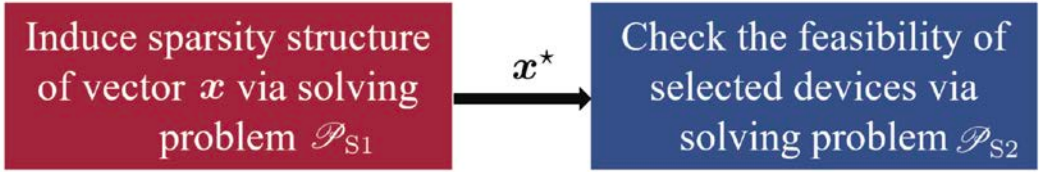
$$\operatorname{rank}(\boldsymbol{M}) = 1 \Leftrightarrow \operatorname{Tr}(\boldsymbol{M}) - \|\boldsymbol{M}\|_2 = 0$$

  - where $\operatorname{Tr}(\boldsymbol{M}) = \sum_{i=1}^{N} \sigma_i(\boldsymbol{M})$ and $\|\boldsymbol{M}\|_2 = \sigma_1(\boldsymbol{M})$

[Ref] J.-y. Gotoh, A. Takeda, and K. Tono, “DC formulations and algorithms for sparse optimization problems,” *Math. Program.*, vol. 169, pp. 141– 176, May 2018.

A DC representation framework

- A two-step framework for device selection



- **Step 1:** obtain the sparse solution such that the objective value achieves zero through increasing k from 0 to M

$$
\begin{aligned}
\mathscr{P}_{\mathrm{S1}} : \underset{\boldsymbol{x},\boldsymbol{M}}{\text{minimize}} \quad & \|\boldsymbol{x}\|_1 - \|\boldsymbol{x}\|_k + \operatorname{Tr}(\boldsymbol{M}) - \|\boldsymbol{M}\|_2 \\
\text{subject to} \quad & \operatorname{Tr}(\boldsymbol{M}) - \gamma_i \boldsymbol{h}_i^{\mathsf{H}} \boldsymbol{M} \boldsymbol{h}_i \leq x_i, \forall i = 1, \cdots, M \\
& \boldsymbol{M} \succeq \mathbf{0}, \quad \operatorname{Tr}(\boldsymbol{M}) \geq 1, \boldsymbol{x} \succeq \mathbf{0}
\end{aligned}
$$

A DC representation framework

- **Step II:** feasibility detection
  - Ordering $\boldsymbol{x}$ in descending order as $x_{\pi(1)} \geq \cdots \geq x_{\pi(M)}$
  - Increasing k from 1 to M, choosing $\mathcal{S}^{[k]}$ as $\{\pi(k), \pi(k+1), \cdots, \pi(M)\}$
- Feasibility detection via DC programming

$$\begin{aligned} \text{find} \quad & \boldsymbol{M} \\ \text{subject to} \quad & \operatorname{Tr}(\boldsymbol{M}) - \gamma_i \boldsymbol{h}_i^{\mathsf{H}} \boldsymbol{M} \boldsymbol{h}_i \leq 0, \forall i \in \mathcal{S}^{[k]} \\ & \boldsymbol{M} \succeq \mathbf{0}, \operatorname{Tr}(\boldsymbol{M}) \geq 1, \operatorname{rank}(\boldsymbol{M}) = 1 \end{aligned}$$

$$\begin{aligned} \mathscr{P}_{\mathrm{S2}} : \underset{\boldsymbol{M}}{\text{minimize}} \quad & \operatorname{Tr}(\boldsymbol{M}) - \|\boldsymbol{M}\|_2 \\ \text{subject to} \quad & \operatorname{Tr}(\boldsymbol{M}) - \gamma_i \boldsymbol{h}_i^{\mathsf{H}} \boldsymbol{M} \boldsymbol{h}_i \leq 0, \forall i \in \mathcal{S}^{[k]} \\ & \boldsymbol{M} \succeq \mathbf{0}, \quad \operatorname{Tr}(\boldsymbol{M}) \geq 1 \end{aligned}$$

DC algorithm with convergence guarantees

- $\mathscr{P}_{\mathrm{S1}}$ and $\mathscr{P}_{\mathrm{S2}}$: minimize the difference of two strongly convex functions

$$\underset{\boldsymbol{X}\in\mathbb{C}^{m\times n}}{\operatorname{minimize}} \quad f(\boldsymbol{X}) = g(\boldsymbol{X}) - h(\boldsymbol{X})$$

  - e.g., $g = \operatorname{Tr}(\boldsymbol{M}) + I_{\mathcal{C}_2}(\boldsymbol{M}) + \frac{\alpha}{2}\|\boldsymbol{M}\|_F^2$ and $h = \|\boldsymbol{M}\|_2 + \frac{\alpha}{2}\|\boldsymbol{M}\|_F^2$

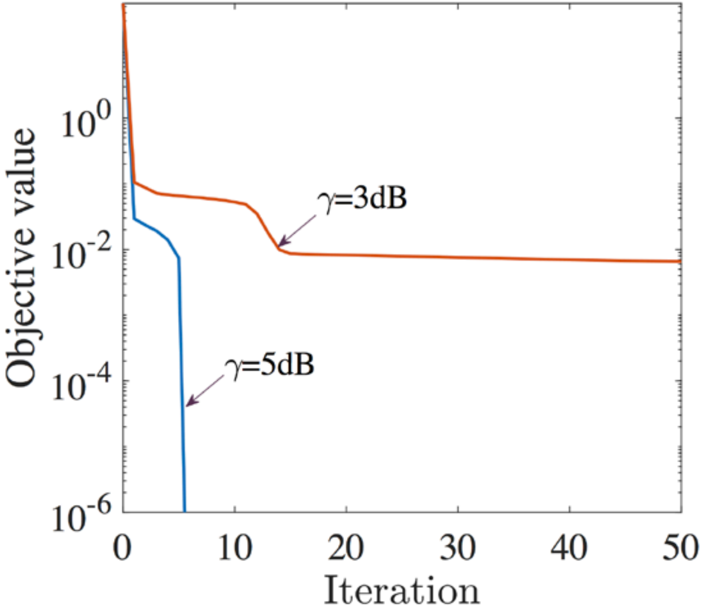
- The DC algorithm via linearizing the concave part

$$\boldsymbol{X}^{[t+1]} = \arg\inf_{\boldsymbol{X}\in\mathcal{X}} \; g(\boldsymbol{X}) - [h(\boldsymbol{X}^{[t]}) + \langle \boldsymbol{X} - \boldsymbol{X}^{[t]}, \partial_{\boldsymbol{X}^{[t]}} h\rangle]$$

  - converge to a critical point with speed $\mathcal{O}(1/t)$

Numerical results

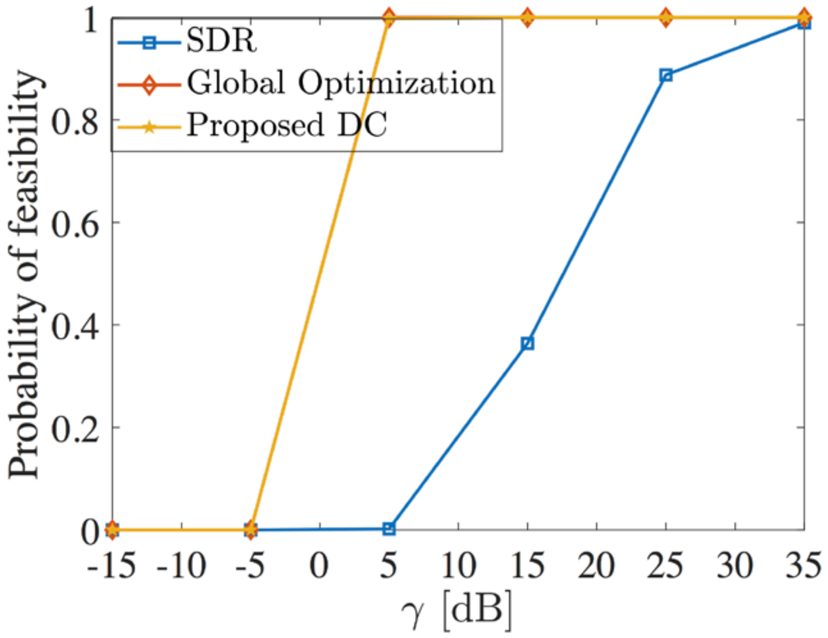
- Convergence of the proposed DC algorithm for problem $\mathscr{P}_{\mathrm{S2}}$



$$
\begin{aligned}
\mathscr{P}_{\mathrm{S2}}: \underset{\boldsymbol{M}}{\text{minimize}} \quad & \operatorname{Tr}(\boldsymbol{M}) - \|\boldsymbol{M}\|_2 \\
\text{subject to} \quad & \operatorname{Tr}(\boldsymbol{M}) - \gamma_i \boldsymbol{h}_i^{\mathsf{H}} \boldsymbol{M} \boldsymbol{h}_i \le 0, \\
& \boldsymbol{M} \succeq \boldsymbol{0}, \quad \operatorname{Tr}(\boldsymbol{M}) \ge 1
\end{aligned}
$$

Numerical results

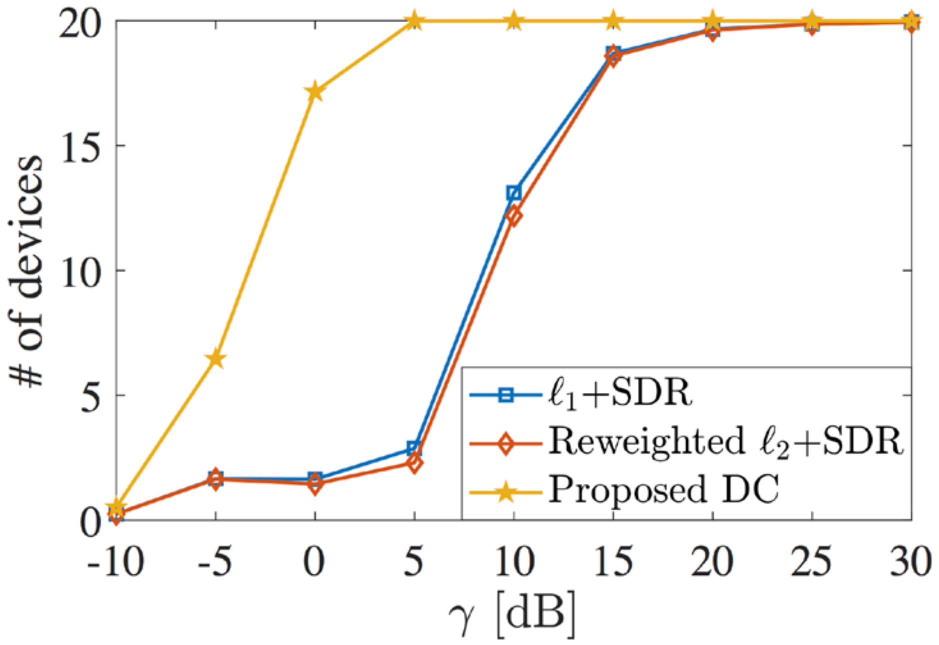
- Probability of feasibility with different algorithms



$$
\begin{aligned}
\text{find} \quad & \boldsymbol{M} \\
\text{subject to} \quad & \mathrm{Tr}(\boldsymbol{M}) - \gamma_i \boldsymbol{h}_i^{\mathsf{H}} \boldsymbol{M} \boldsymbol{h}_i \leq 0, \forall i \in \mathcal{S}^{[k]} \\
& \boldsymbol{M} \succeq \mathbf{0}, \mathrm{Tr}(\boldsymbol{M}) \geq 1, \mathrm{rank}(\boldsymbol{M}) = 1
\end{aligned}
$$

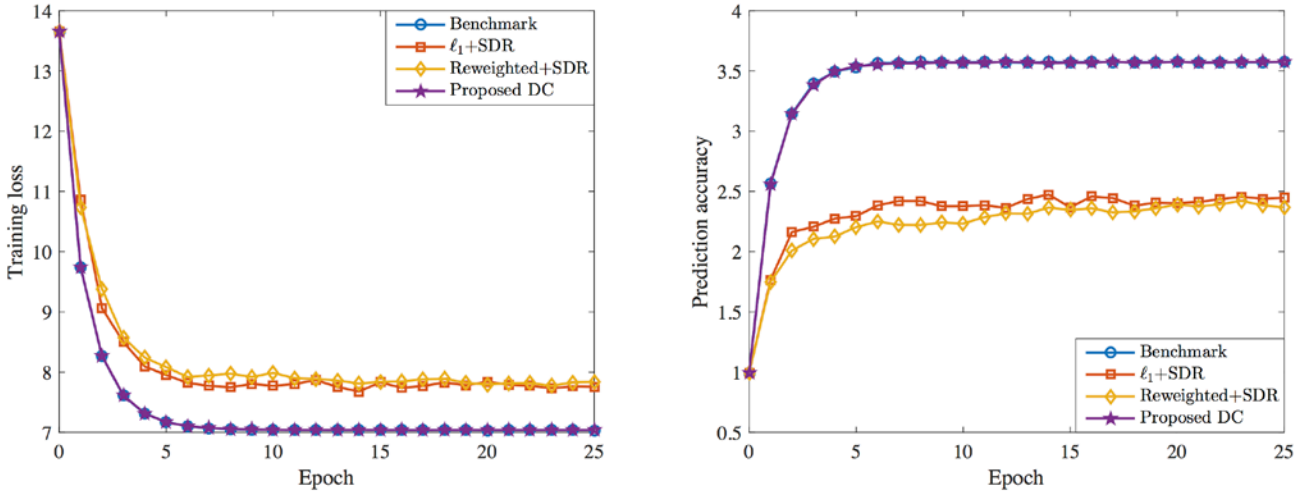
Numerical results

- Average number of selected devices with different algorithms



Numerical results

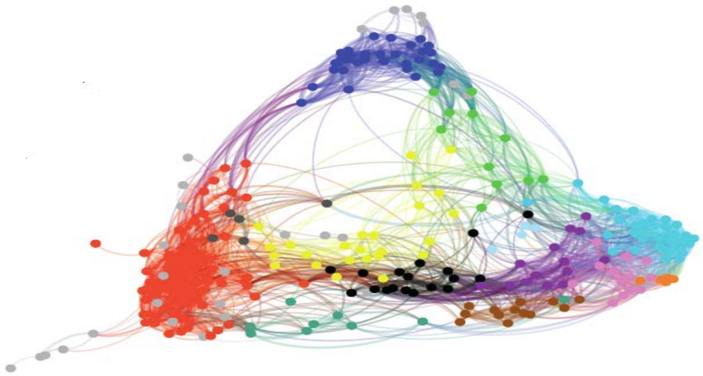
- Performance of proposed fast model aggregation in federated learning
  - Training an SVM classifier on CIFAR-10 dataset



(a) Training loss

(b) Relative prediction accuracy

Vignettes B: Intelligent reflecting surface empowered federated learning



Smart radio environments

- Current wireless networks: no control of radio waves
  - Perceive the environment as an "unintentional adversary" to communication
  - Optimize only the end-points of the communication network
  - No control of the environment, which is viewed as a passive spectator
- Smart radio environments: reconfigure the wireless propagations

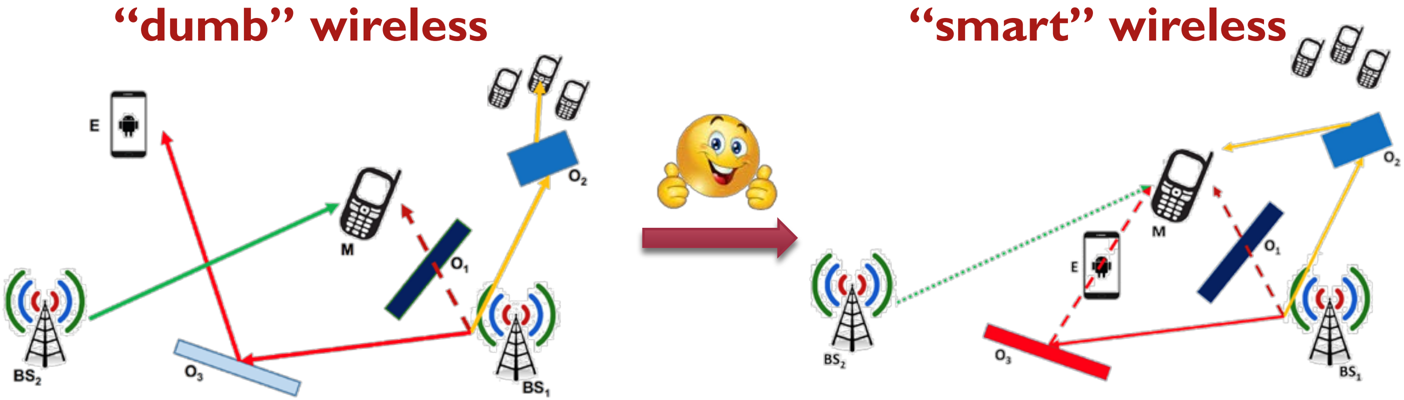


Fig. credit: Renzo

Intelligent reflecting surface

- **Working principle of intelligent reflecting surface (IRS):** different elements of an IRS can reflect the incident signal by controlling its amplitude and/or phase for directional signal enhancement or nulling

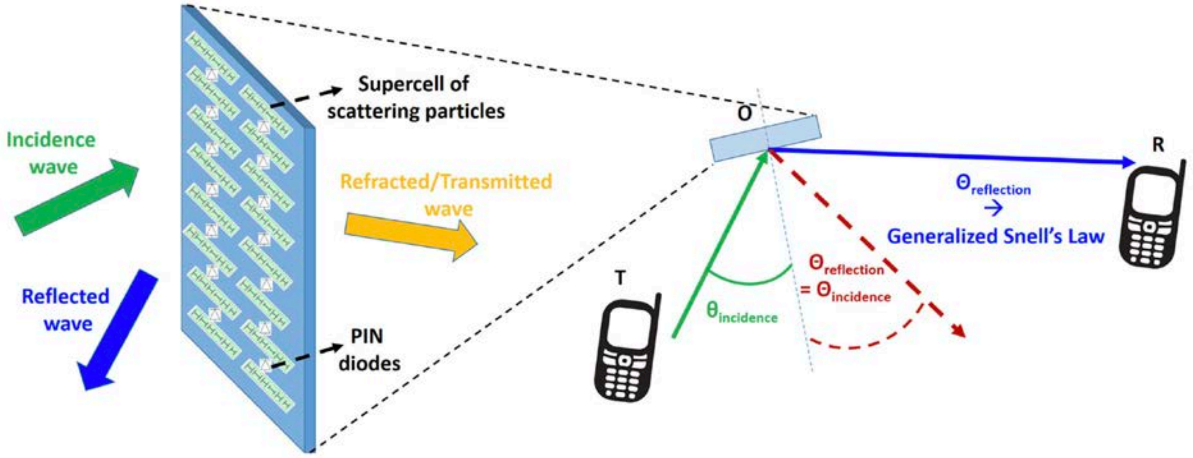


Fig. credit: Renzo

1. no any active transmit module
2. operate in full-duplex mode

improve spectral and energy efficiency

Intelligent reflecting surface

- Architecture of intelligent reflecting surface

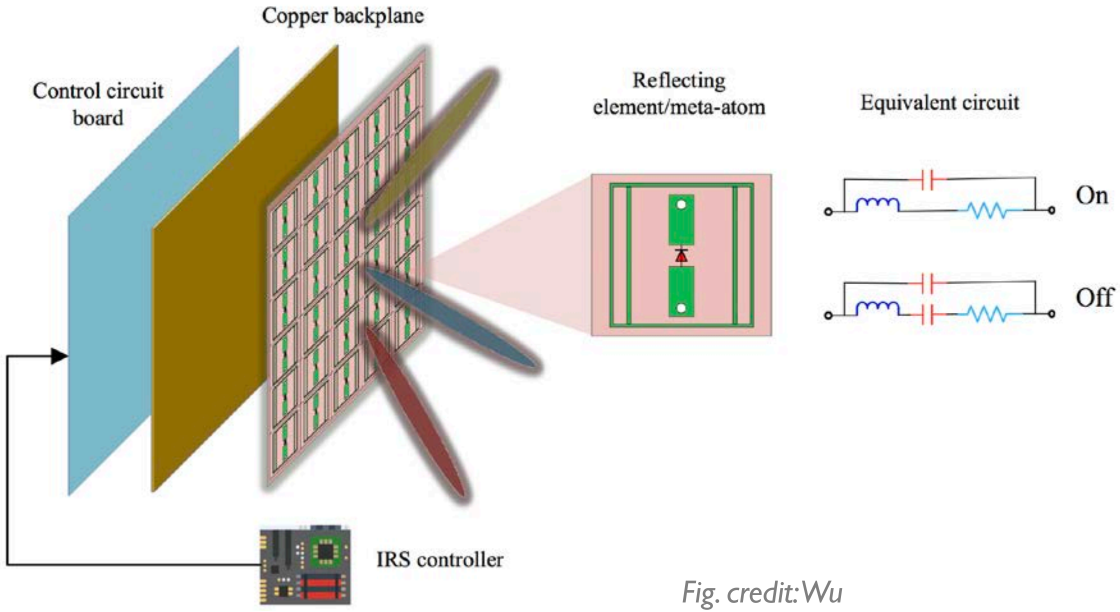


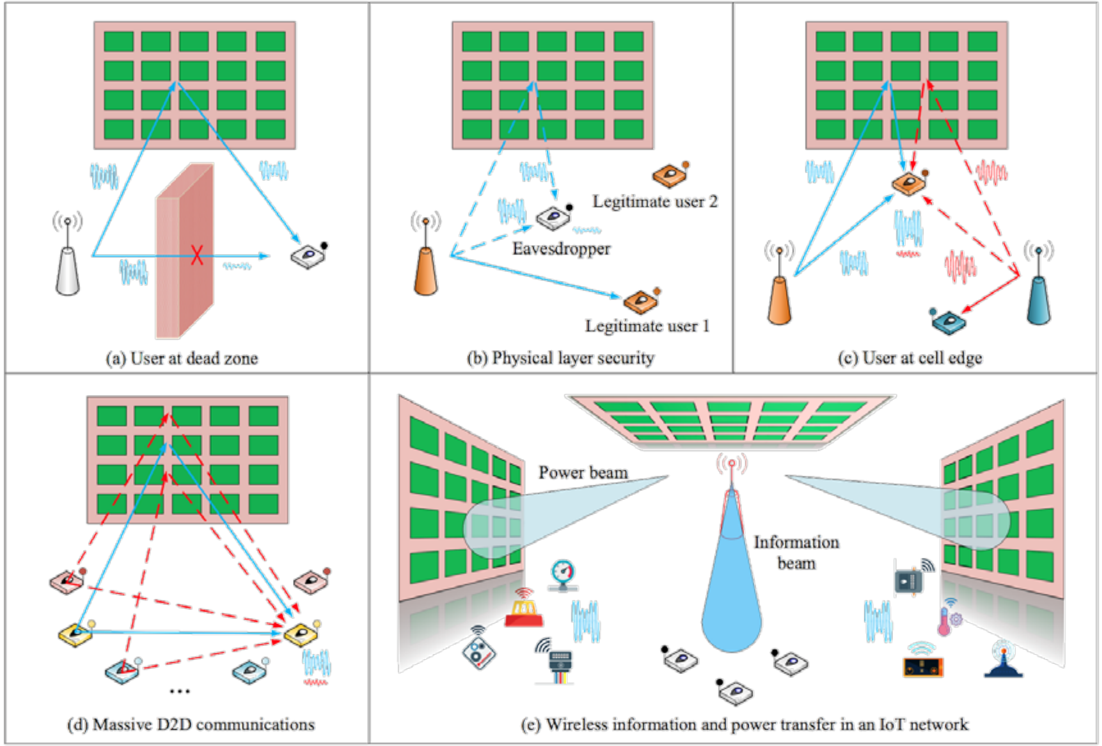
Fig. credit: Wu

1. Outer layer: a large number of metallic patches (elements) are printed on a dielectric substrate to directly interact with incident signals.

2. Second layer: a copper plate is used to avoid the signal energy leakage.

3. Inner layer: a control circuit board for adjusting the reflection amplitude/phase shift of each element, triggered by a smart controller attached to the IRS.

Intelligent reflecting surface meet wireless networks



intelligent reflecting surface meets wireless network:

- over-the-air computation
- edge computing/caching
- wireless power transfer
- D2D communications
- massive MIMO
- NOMA
- mmWave
- …

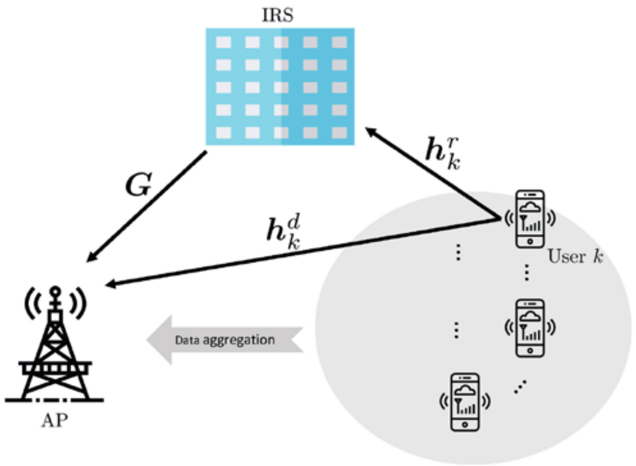
Fig. credit: Wu

IRS empowered AirComp

- Intelligent reflecting surface (IRS):
  - overcoming unfavorable signal propagation conditions
  - improving spectrum and energy efficiency
  - tuning phase shifts with M passive elements

$$\boldsymbol{\Theta} = \mathrm{diag}(\beta e^{j\theta_1}, \cdots, \beta e^{j\theta_M})$$

w.l.o.g. assuming $\beta = 1$



IRS aided AirComp system:
build controllable wireless environments to boost received signal power

Problem formulation

- Received signal at the AP: $\boldsymbol{y} = \sum_{k=1}^{K}(\boldsymbol{G}\boldsymbol{\Theta}\boldsymbol{h}_k^r + \boldsymbol{h}_k^d) b_k s_k + \boldsymbol{n}$

  w.l.o.g. suppose target function: $s := \sum_{k=1}^{K} s_k$

- Aggregation error:

$$\mathsf{MSE}(\boldsymbol{m}) = \frac{\sigma^2}{P_0} \max_k \frac{\|\boldsymbol{m}\|^2}{\|\boldsymbol{m}^{\mathsf{H}}(\boldsymbol{G}\boldsymbol{\Theta}\boldsymbol{h}_k^r + \boldsymbol{h}_k^d)\|^2}$$

$\boldsymbol{m}$ received beamforming vector

  - optimal transmitter scalar: $b_k = \sqrt{\eta}\frac{(\boldsymbol{m}^{\mathsf{H}}(\boldsymbol{G}\boldsymbol{\Theta}\boldsymbol{h}_k^r + \boldsymbol{h}_k^d))^{\mathsf{H}}}{\|\boldsymbol{m}^{\mathsf{H}}(\boldsymbol{G}\boldsymbol{\Theta}\boldsymbol{h}_k^r + \boldsymbol{h}_k^d)\|^2}$

- **Proposal:** joint design for AirComp transceivers and IRS phase shifts

$$\begin{aligned} \underset{\boldsymbol{m},\boldsymbol{\Theta}}{\text{minimize}} \quad & \left(\max_k \frac{\|\boldsymbol{m}\|^2}{\|\boldsymbol{m}^{\mathsf{H}}(\boldsymbol{G}\boldsymbol{\Theta}\boldsymbol{h}_k^r + \boldsymbol{h}_k^d)\|^2}\right) \\ \text{subject to} \quad & 0 \le \theta_n \le 2\pi, \forall n = 1, \cdots, N. \end{aligned}$$

$$\longrightarrow$$

$$\begin{aligned} \mathscr{P}: \quad \underset{\boldsymbol{m},\boldsymbol{\Theta}}{\text{minimize}} \quad & \|\boldsymbol{m}\|^2 \\ \text{subject to} \quad & \|\boldsymbol{m}^{\mathsf{H}}(\boldsymbol{G}\boldsymbol{\Theta}\boldsymbol{h}_k^r + \boldsymbol{h}_k^d)\|^2 \ge 1, \forall k, \\ & 0 \le \theta_n \le 2\pi, \forall n = 1, \cdots, N. \end{aligned}$$

Nonconvex bi-quadratic programming

- Nonconvex bi-quadratic programming problem

$$\begin{aligned} \mathscr{P}: \quad \underset{\boldsymbol{m}, \boldsymbol{\Theta}}{\text{minimize}} \quad & \|\boldsymbol{m}\|^2 \\ \text{subject to} \quad & \|\boldsymbol{m}^{\mathsf{H}}(\boldsymbol{G}\boldsymbol{\Theta}\boldsymbol{h}_k^r + \boldsymbol{h}_k^d)\|^2 \geq 1, \forall k, \\ & 0 \leq \theta_n \leq 2\pi, \forall n = 1, \cdots, N. \end{aligned}$$

- **Challenges:**
  - nonconvex quadratic constraints with respect to $\boldsymbol{m}$ and $\boldsymbol{\Theta}$
- **Solution:**
  - Alternating minimization for $\boldsymbol{m}$ and $\boldsymbol{\Theta}$
  - Matrix lifting to alternatively linearize nonconvex bi-quadratic constraints

An alternating DC framework

- **Goal:** updating receiver beamforming vector $\boldsymbol{m}$ with fixed IRS phase shifts $\boldsymbol{\Theta}$

$$\begin{aligned} \underset{\boldsymbol{m}}{\text{minimize}} \quad & \|\boldsymbol{m}\|^2 \\ \text{subject to} \quad & \|\boldsymbol{m}^{\mathsf{H}}(\boldsymbol{G}\boldsymbol{\Theta}\boldsymbol{h}_k^r + \boldsymbol{h}_k^d)\|^2 \geq 1, \forall k. \end{aligned}$$

matrix lifting $\boldsymbol{M} = \boldsymbol{m}\boldsymbol{m}^{\mathsf{H}}$

DC programming

$$\begin{aligned} \underset{\boldsymbol{M}}{\text{minimize}} \quad & \text{trace}(\boldsymbol{M}) \\ \text{subject to} \quad & \text{trace}(\boldsymbol{M}\boldsymbol{H}_k) \geq 1, \forall k, \\ & \boldsymbol{M} \succeq 0, \text{rank}(\boldsymbol{M}) = 1. \end{aligned}$$

$$\begin{aligned} \underset{\boldsymbol{M}}{\text{minimize}} \quad & \text{Tr}(\boldsymbol{M}) + \rho(\text{Tr}(\boldsymbol{M}) - \|\boldsymbol{M}\|_2) \\ \text{subject to} \quad & \text{Tr}(\boldsymbol{M}\boldsymbol{H}_k) \geq 1, \forall k, \\ & \boldsymbol{M} \succeq 0, \end{aligned}$$

DC representation $\text{rank}(\boldsymbol{M}) = 1 \Leftrightarrow \text{Tr}(\boldsymbol{M}) - \|\boldsymbol{M}\|_2 = 0$

An alternating DC framework

- **Goal:** updating phase shifts with fixed beamformer $\boldsymbol{v} = \mathrm{diag}(\boldsymbol{\Theta}) = [e^{j\theta_1}, \cdots, e^{j\theta_M}]^\mathsf{T}$

denoting $\boldsymbol{R}_k = \begin{bmatrix} \boldsymbol{a}_k \boldsymbol{a}_k^\mathsf{H}, & \boldsymbol{a}_k c_k \\ c_k^\mathsf{H} \boldsymbol{a}_k^\mathsf{H}, & 0 \end{bmatrix}, \tilde{\boldsymbol{v}} = \begin{bmatrix} \boldsymbol{v} \\ t \end{bmatrix}, \boldsymbol{a}_k^\mathsf{H} = \boldsymbol{m}^\mathsf{H} \boldsymbol{G} \mathrm{diag}(\boldsymbol{h}_k^r), c_k = \boldsymbol{m}^\mathsf{H} \boldsymbol{h}_k^d$

$$\begin{aligned} \text{find} \quad & \boldsymbol{v} \\ \text{subject to} \quad & |\boldsymbol{m}^\mathsf{H}(\boldsymbol{G}\mathrm{diag}(\boldsymbol{h}_k^r)\boldsymbol{v} + \boldsymbol{h}_k^d)|^2 \geq 1, \forall k, \\ & |v_n|^2 = 1, \forall v = 1, \cdots, N. \end{aligned}$$

$$\begin{aligned} \text{find} \quad & \boldsymbol{v} \\ \text{subject to} \quad & \tilde{\boldsymbol{v}}^\mathsf{H} \boldsymbol{R}_k \tilde{\boldsymbol{v}} + |c_k|^2 \geq 1, \forall k, \\ & |v_n|^2 = 1, \forall v = 1, \cdots, N, \end{aligned}$$

matrix lifting $\boldsymbol{V} = \tilde{\boldsymbol{v}} \tilde{\boldsymbol{v}}^\mathsf{H}$

$$\begin{aligned} \text{find} \quad & \boldsymbol{V} \\ \text{subject to} \quad & \mathrm{Tr}(\boldsymbol{R}_k \boldsymbol{V}) + |c_k|^2 \geq 1, \forall k, \\ & \boldsymbol{V}_{n,n} = 1, \forall n = 1, \cdots, N, \\ & \boldsymbol{V} \succeq 0, \quad \mathrm{rank}(\boldsymbol{V}) = 1. \end{aligned}$$

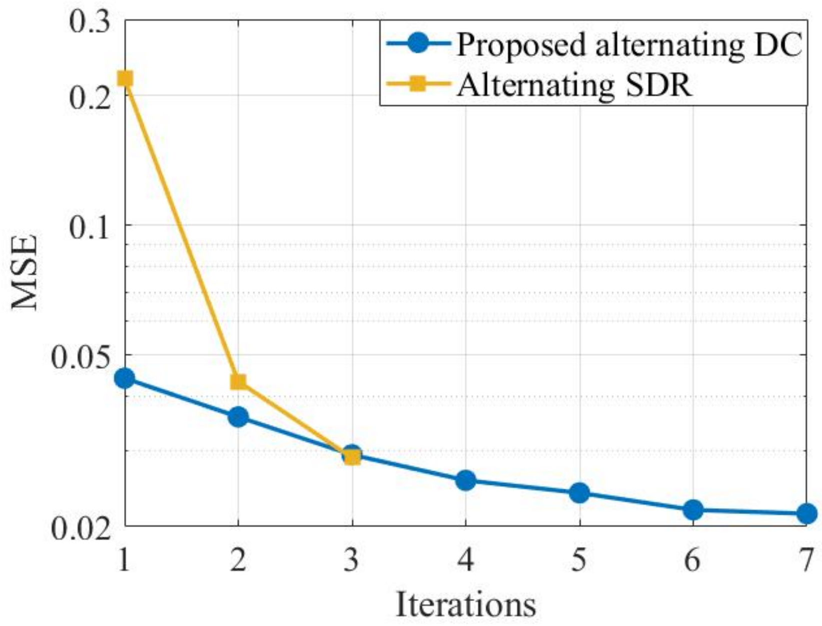
DC representation

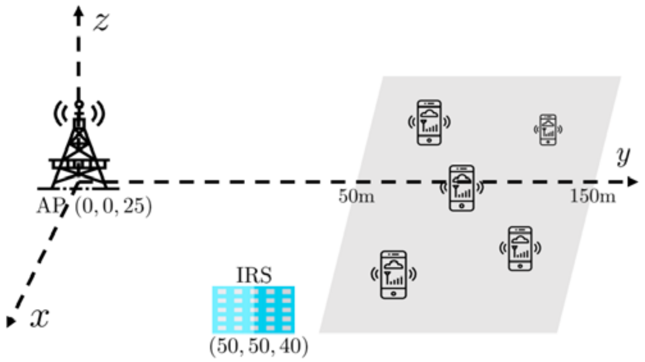
DC programming

$$\begin{aligned} \underset{\boldsymbol{V}}{\text{minimize}} \quad & \mathrm{Tr}(\boldsymbol{V}) - \|\boldsymbol{V}\|_2 \\ \text{subject to} \quad & \mathrm{Tr}(\boldsymbol{R}_k \boldsymbol{V}) + |c_k|^2 \geq 1, \forall k, \\ & \boldsymbol{V}_{n,n} = 1, \forall n = 1, \cdots, N, \\ & \boldsymbol{V} \succeq 0. \end{aligned}$$

Numerical results

- Convergence behaviors of the proposed alternating DC algorithm

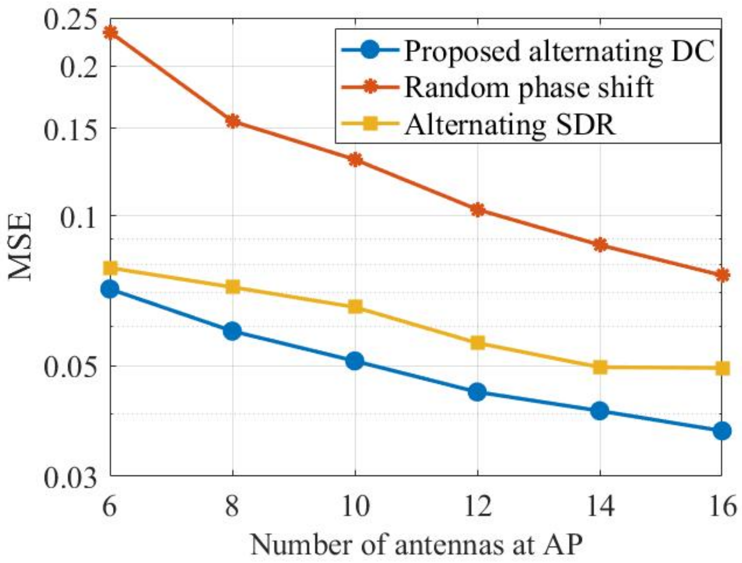


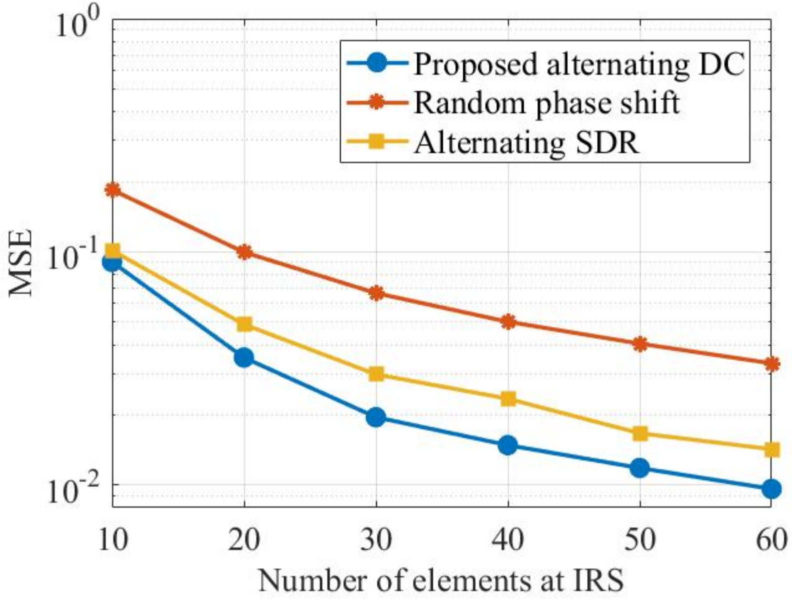


layout of AP, IRS and users

Numerical results

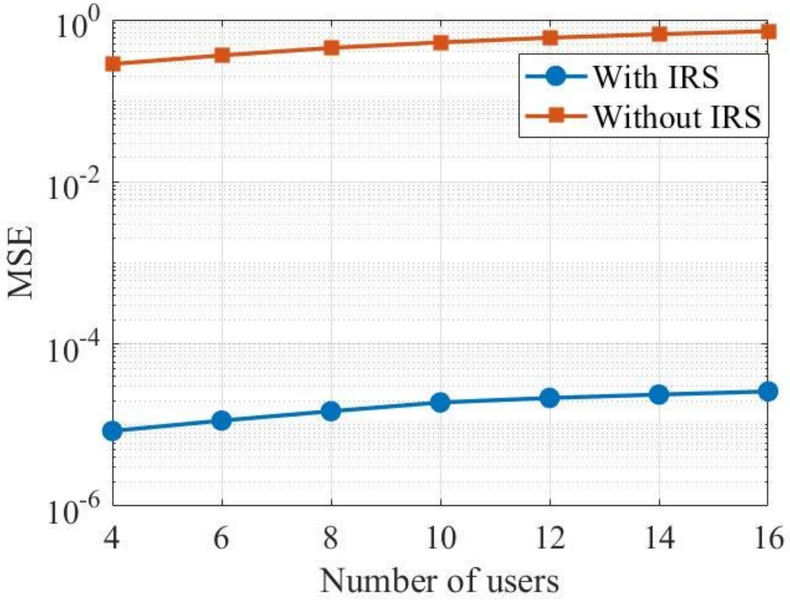
- Performance of different algorithms with different network settings





Numerical results

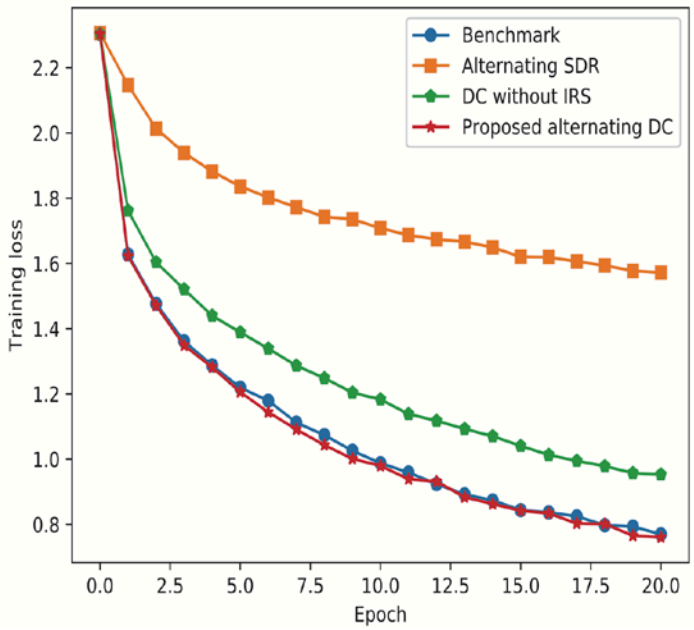
- The power of IRS for AirComp



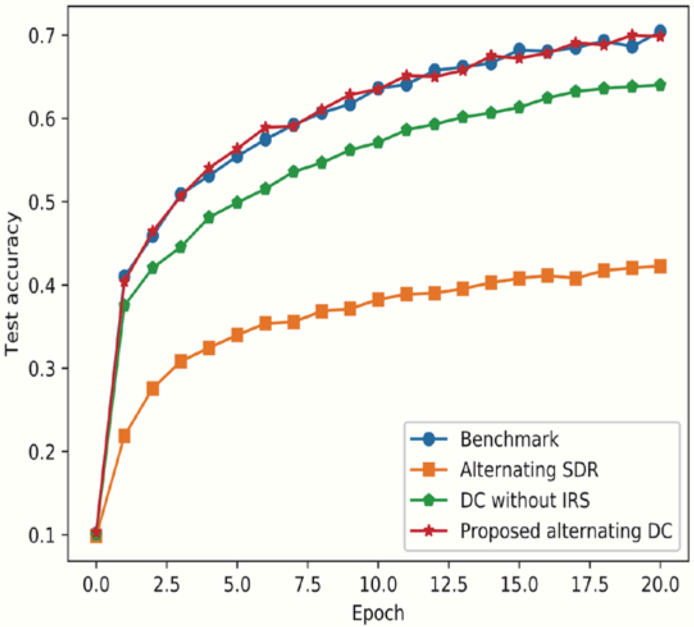
Insights: deploying IRS in AirComp system can significantly enhance the MSE performance for data aggregation

IRS empowered federated learning system

- The power of IRS for federated learning



training loss



prediction accuracy

Concluding remarks

- **Federated learning over "intelligent" wireless networks**
  - Federated learning via over-the-air computation
  - Over-the-air computation empowered by intelligent reflecting surface
- **Sparse and low-rank optimization framework**
  - Joint device selection and beamforming design for over-the-air computation
  - Joint phase shifts and transceiver design for IRS empowered AirComp
- **A unified DC programming framework**
  - DC representation for sparse and low-rank functions

Future directions

- **Federated learning**
  - ➢ stragglers, security, provable guarantees, …
- **Over-the-air computation**
  - ➢ channel uncertainty, synchronization, security, …
- **Sparse and low-rank optimization via DC programming**
  - ➢ optimality, scalability,…

To learn more…

- **Web:** http://shiyuanming.github.io/publicationstopic.html
- **Papers:**

- K. B. Letaief, W. Chen, **Y. Shi**, J. Zhang, and Y. Zhang, "The roadmap to 6G - AI empowered wireless networks," *IEEE Commun. Mag*., vol. 57, no. 8, pp. 84-90, Aug. 2019.
- J. Dong and **Y. Shi**, "Nonconvex demixing from bilinear measurements," *IEEE Trans. Signal Process*., vol. 66, no. 19, pp. 5152-5166, Oct., 2018.
- M. C. Tsakiris, L. Peng, A. Conca, L. Kneip, **Y. Shi**, and H. Choi, "An algebraic-geometric approach to shuffled linear regression," *IEEE Trans. Inf. Theory*., under major revision, 2019. https://arxiv.org/abs/1810.05440
- K. Yang, **Y. Shi**, and Z. Ding, "Data shuffling in wireless distributed computing via low-rank optimization," *IEEE Trans. Signal Process*., vol. 67, no. 12, pp. 3087-3099, Jun., 2019.
- K. Yang, **Y. Shi**, W. Yu, and Z. Ding, "Energy-efficient processing and robust wireless cooperative transmission for edge inference," *submitted*. https://arxiv.org/abs/1907.12475
- S. Hua, Y. Zhou, K. Yang, and **Y. Shi**, "Reconfigurable intelligent surface for green edge inference," *submitted*. https://arxiv.org/abs/1912.00820
- K. Yang, T. Jiang, **Y. Shi**, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun*., under minor revision, 2019. https://arxiv.org/abs/1812.11750
- T. Jiang and **Y. Shi**, "Over-the-air computation via intelligent reflecting surfaces," in *Proc. IEEE Global Commun. Conf. (Globecom)*, Waikoloa, Hawaii, USA, Dec. 2019. https://arxiv.org/abs/1904.12475

Thanks

http://shiyuanming.github.io/home.html