Introduction to Decentralized Optimization

Mingyi Hong

University Of Minnesota

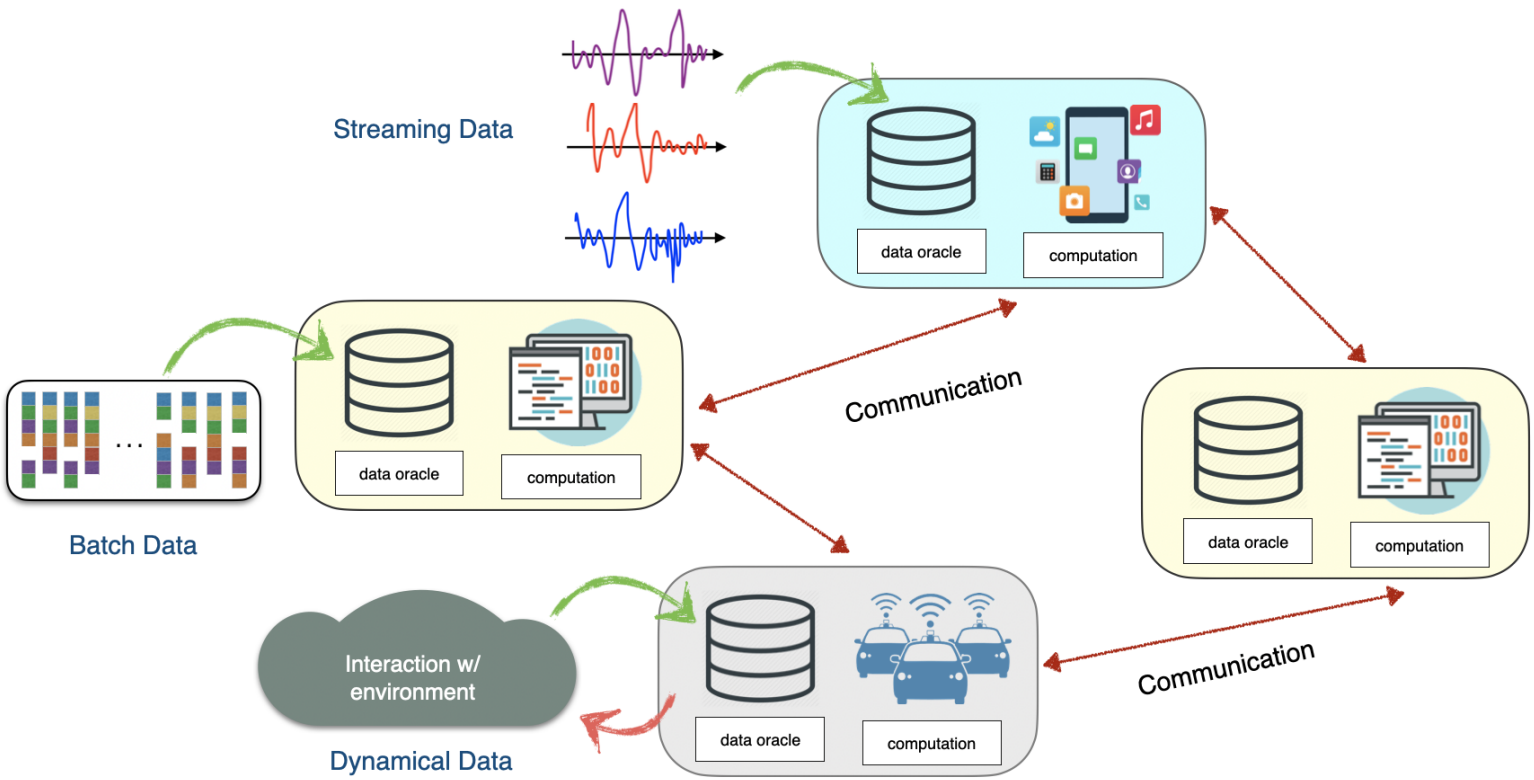
Outline

- Decentralized Optimization
- Overview of the Course
- Some Contemporary Applications

Decentralized Optimization

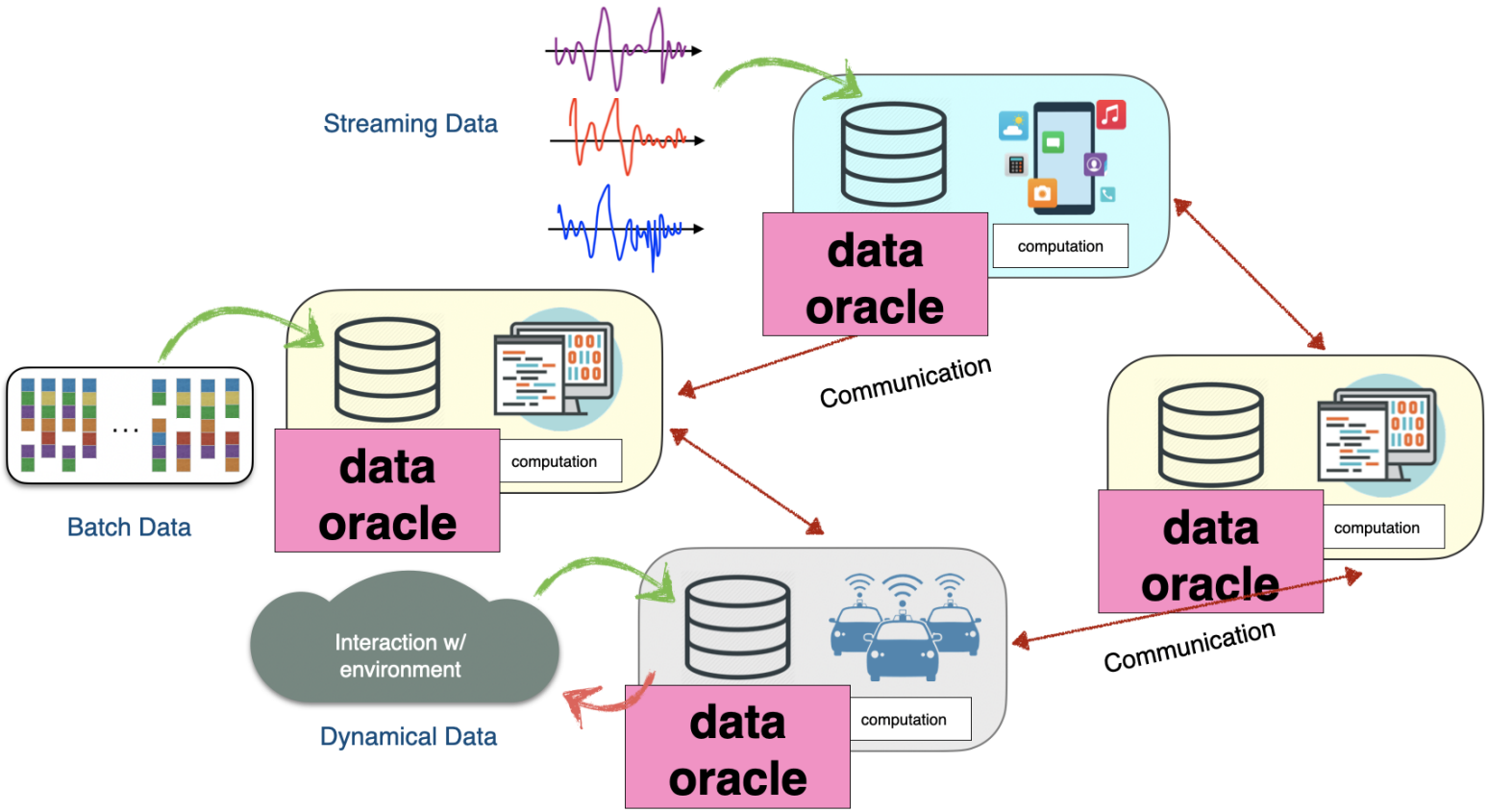
Some Key Elements

Figure 1.1: Key Elements



Some Key Elements

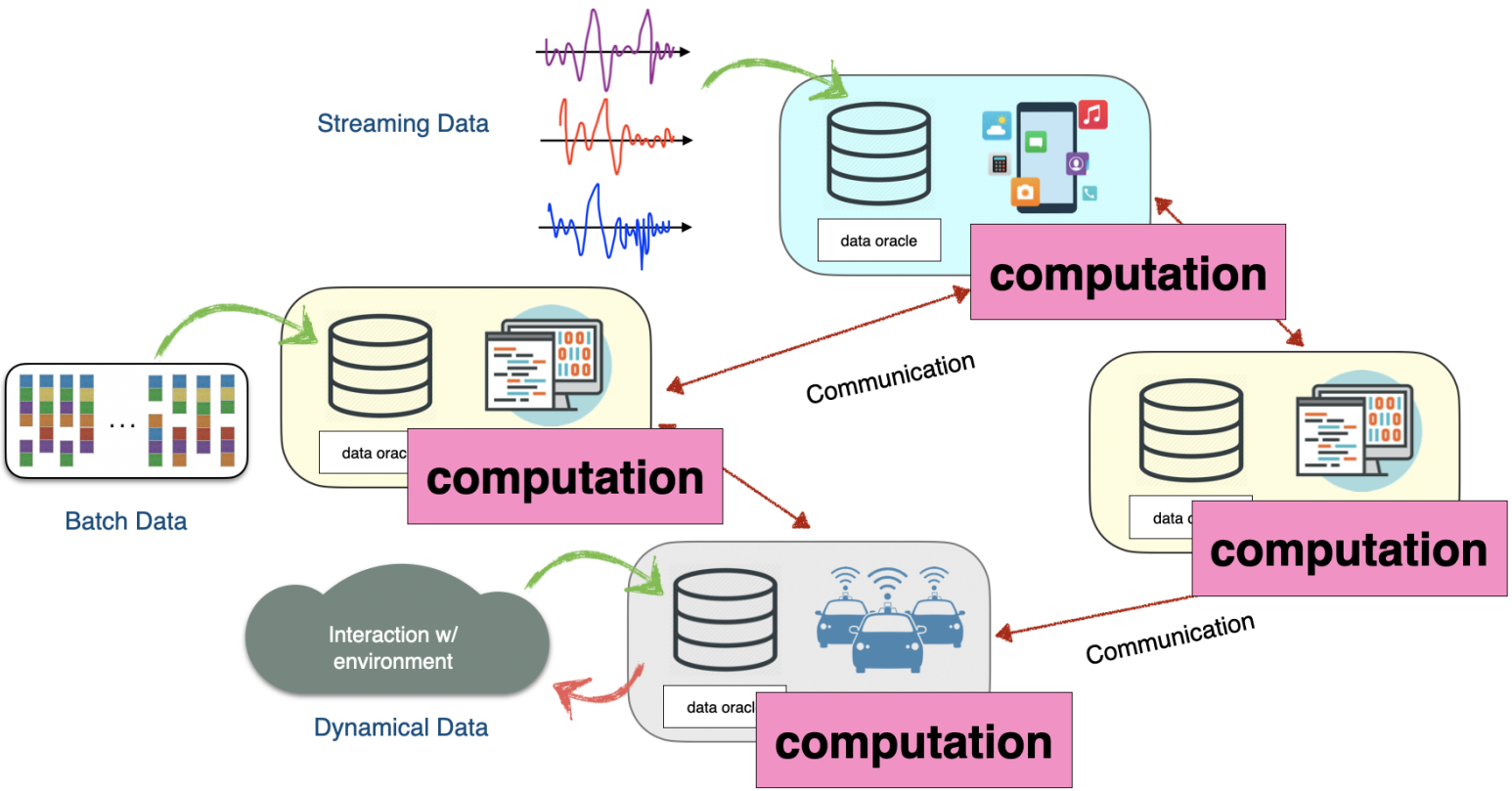
Figure 1.2: Key Elements



Data Oracle: Describing the data acquisition process.

Some Key Elements

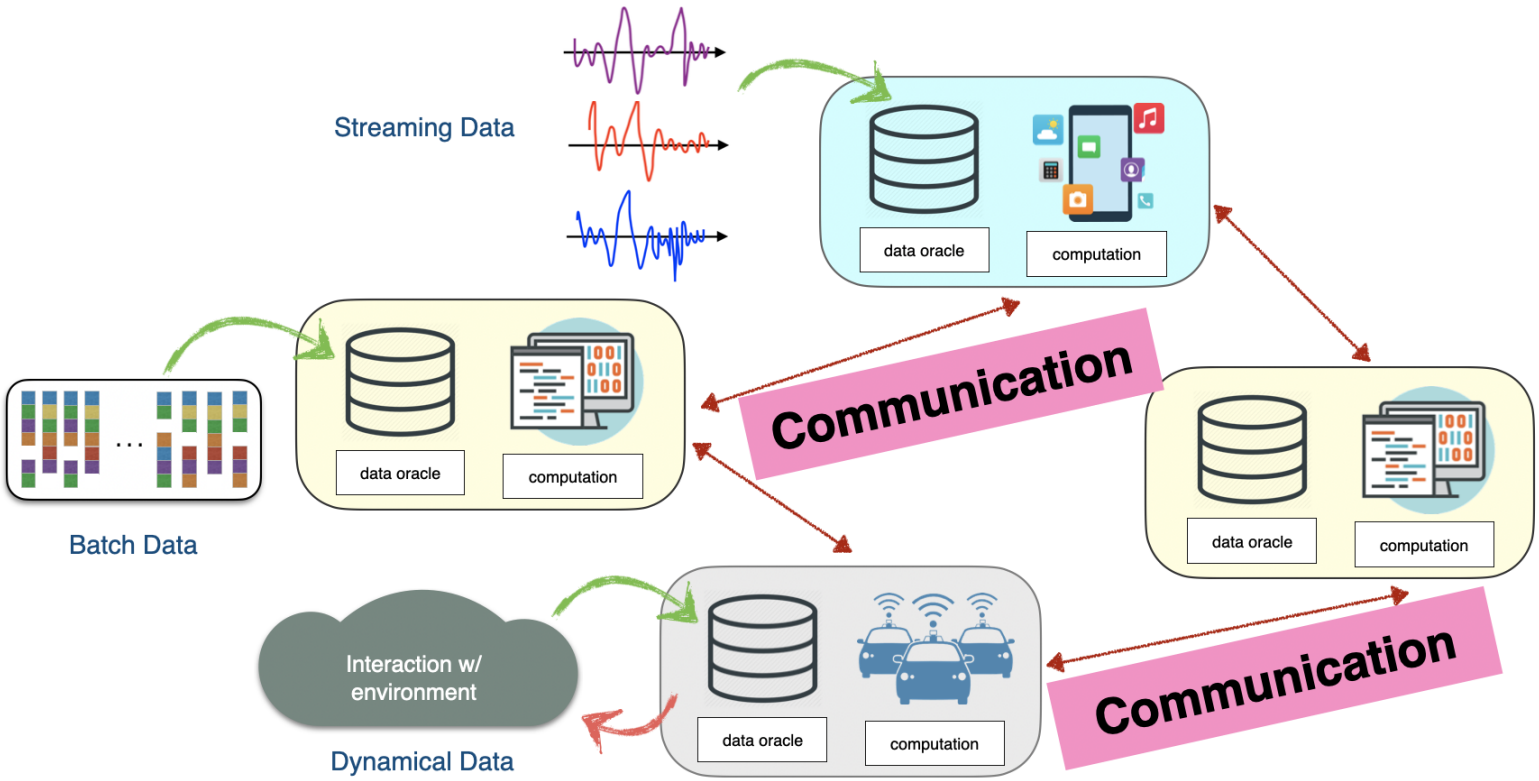
Figure 1.3: Key Elements



Computation Models: How the data is processed, which local problems to solved, etc...

Some Key Elements

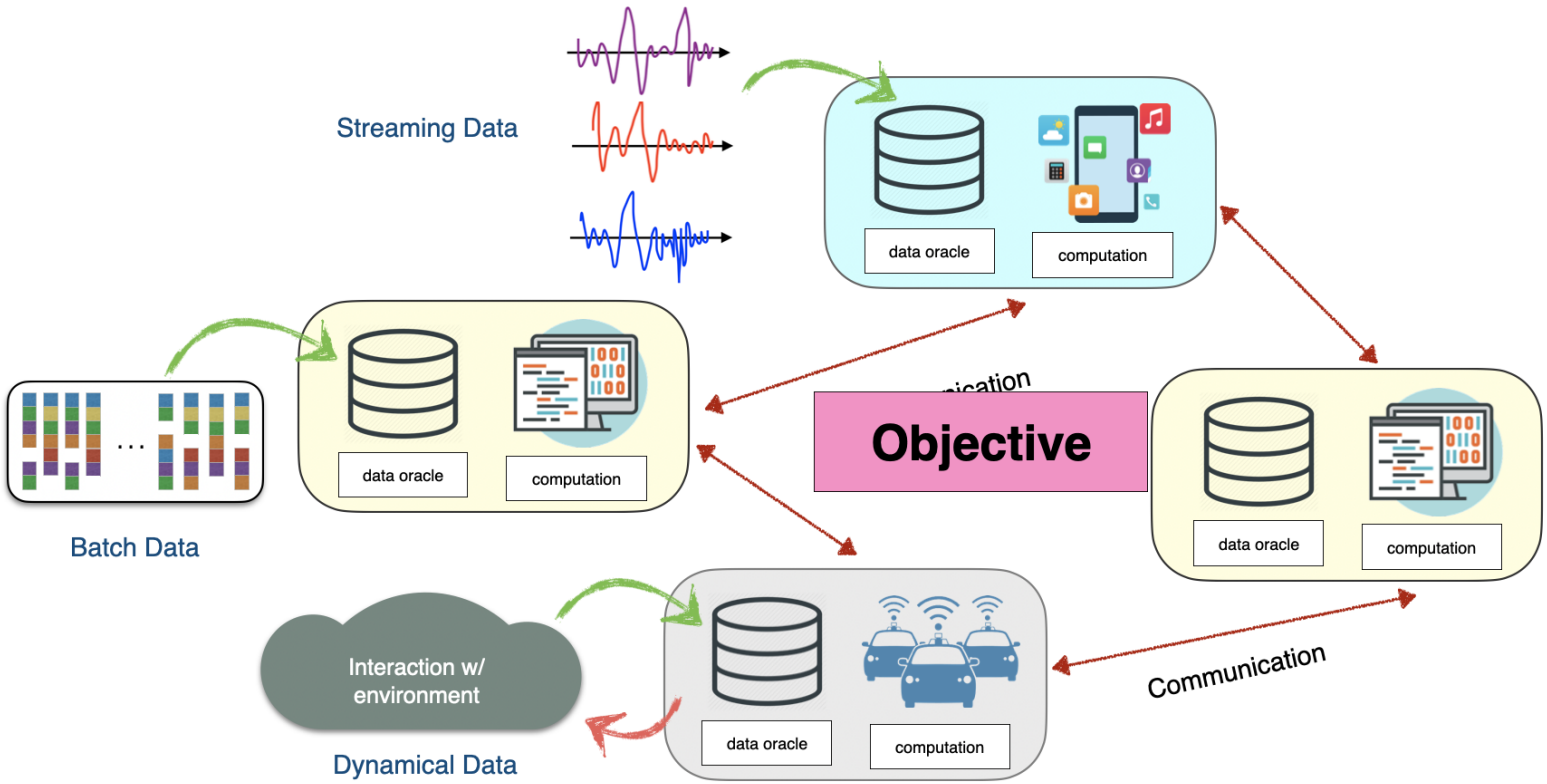
Figure 1.4: Key Elements



Communication: What information are exchanged, and how?

Some Key Elements

Figure 1.5: Key Elements



Objective: What kind of overall goal to achieve?

Key Research Questions

- How to provide mathematical models for different applications?
- How to characterize and model different components of decentralized optimization?
- How to design algorithms, and provide analysis
- ...

Course Outline

- This course will
  - Provide some basic modeling and analytical tools for classical decentralized optimization
  - Show you how to analyze and understand contemporary algorithms (say, after 2010)
  - Show you how to understand the intuition behind different algorithms
  - Show how to apply generic algorithms, and analytical tools, to solve specialized problems
  - Discuss current research trends and recent advances

Course Schedule

- Day 1:
  - Introduction
  - Preliminary on optimization and graph theory
- Day 2:
  - Average consensus problem
  - Decentralized gradient descent algorithm
- Day 3:
  - Modern methods for convex problems
  - Methods for non-convex problems

Course Outline

- Day 4:
  - Algorithms for stochastic problems
  - Recent advances and new topics
- Day 5:
  - Federated Learning
  - Other applications/ recent developments

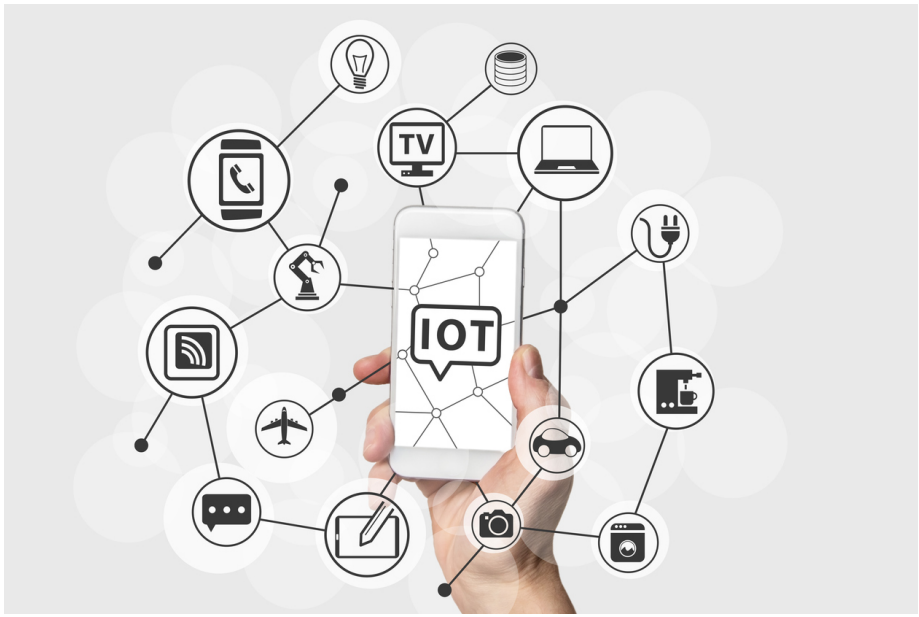
Logistics and Notes

- Because of the time difference, our lecture time is
  - **8:00-10:00** AM, **12:00-2:00** PM Beijing time
  - 50 $\rightarrow$ 55 min each session, 4 sessions a day
- I will distribute the slides, but some of the derivations will not be on slides; they will be derived in-class
- Also discussions will be made directly on slides
- This is a new set of slides; if you spot any typos or inaccuracies, please do let me know
- No office hours, but you are welcome to send me emails to discuss
- Mostly no homeworks, but will be useful to read related literatures

Applications

Application 1: Internet of Things

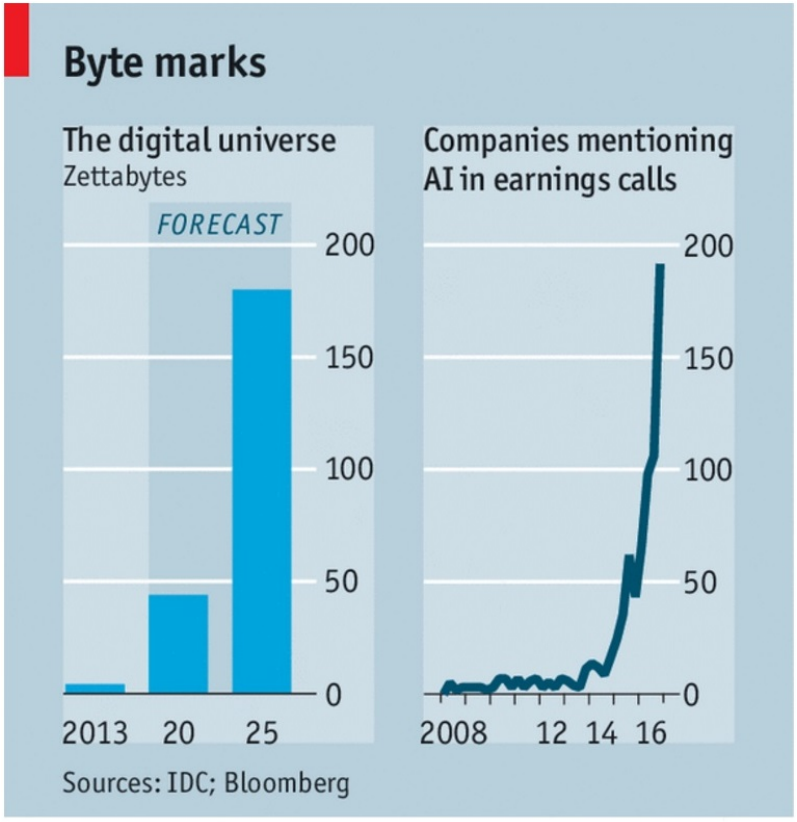
- We live in the era of the Internet of Things (IoT).
- # of networked objects such as sensors, actuators, and controllers has been growing exponentially
- Total # reach 212 billion globally by 2020 [Gantz et atl 12,Al-Fuqaha et al 15]



Application 1: Internet of Things

- A number of unique features of the IoT
  1. Explosive "distributed" data
  2. Increasingly powerful nodes
  3. Very complex tasks
- Poses new challenges and opportunities for the network/protocol design and performance analysis

Feature I - Explosive Distributed Data



Feature I - Explosive Distributed Data

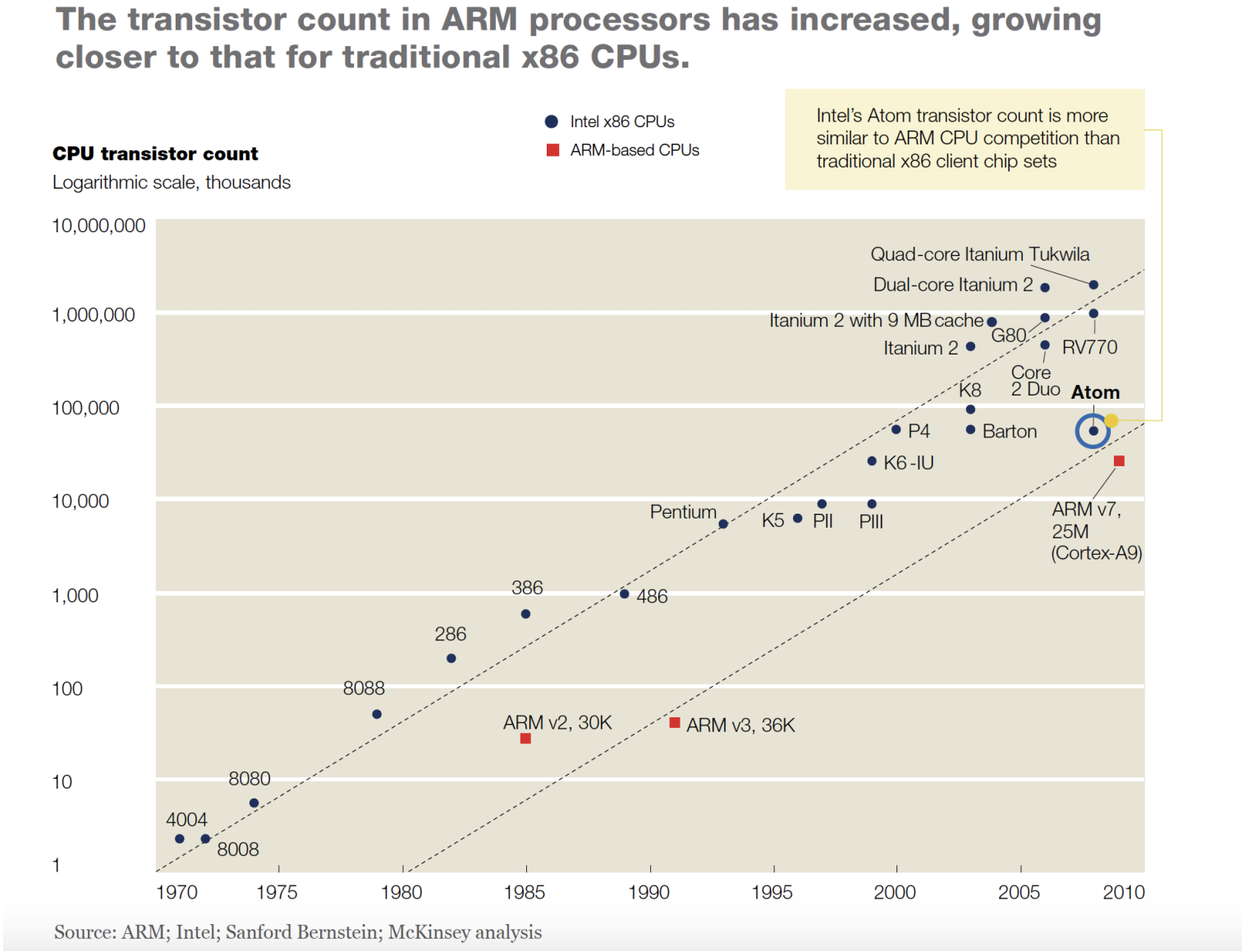
- Data volume is growing fast
- In 2025 the data created every year will reach 180 zettabyte (180 followed by 21 zeros)
- Collected and stored distributedly by IoT devices
- Pumping it all through a broadband Internet connection would take over 450m years

Feature I - Explosive Distributed Data

- Data are collected and located in geographically distributed locations

- Decision makers are geographically distributed
- **Question:** How to enable fast and distributed decision making/learning using decentralized data?

Increasing Node Capabilities



Application 2: Distributed Training of ML Models

- Distributed machine learning tasks [Balcan et al 15]
- Collaborative training deep neural networks, based on nodes' local data [Jiang et al 17] [Lian et al 17]
- High-dimensional problem, difficult and non-convex objectives

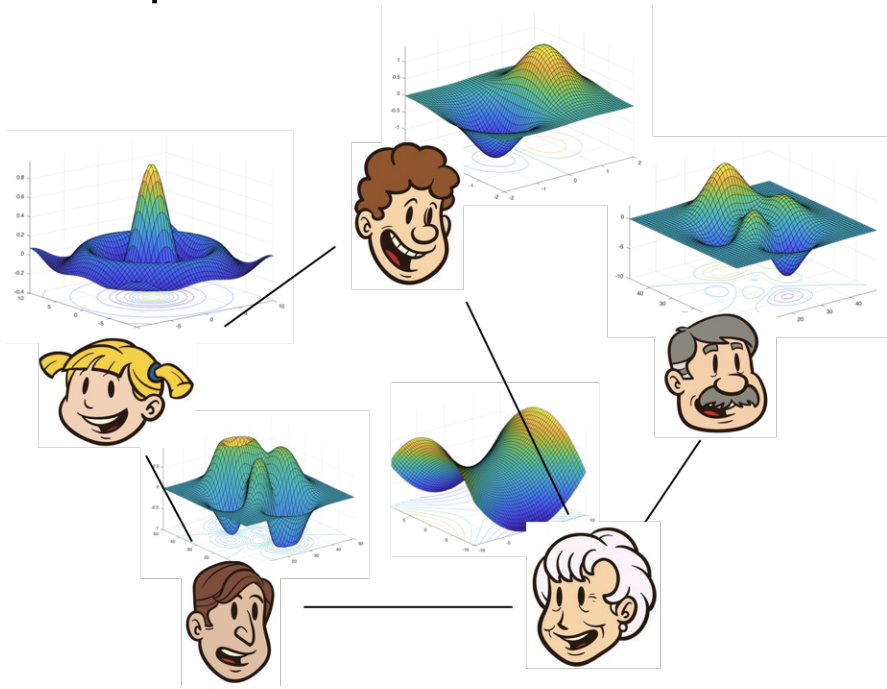
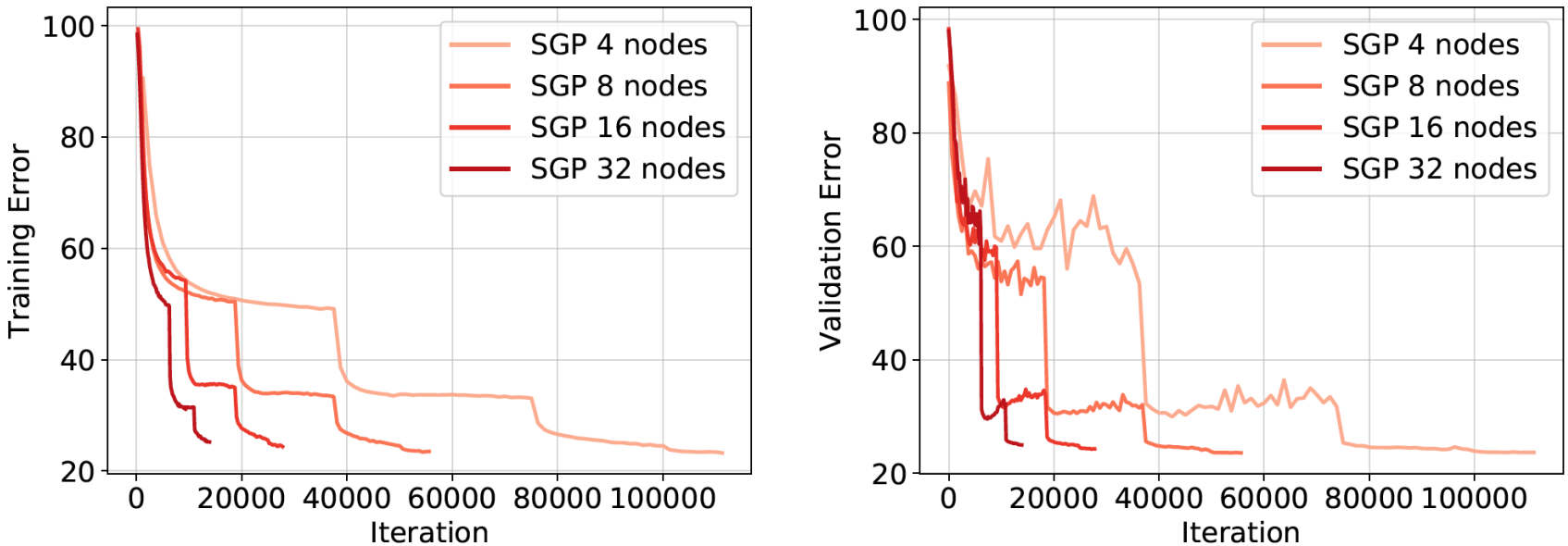


Figure 2.1: A distributed learning setting with distributed data.

Application 2: Distributed Training of ML Models

- Even data are centrally located, distributed training using multiple machines/GPUs can significantly outperform centralized training [Assran et al 19]

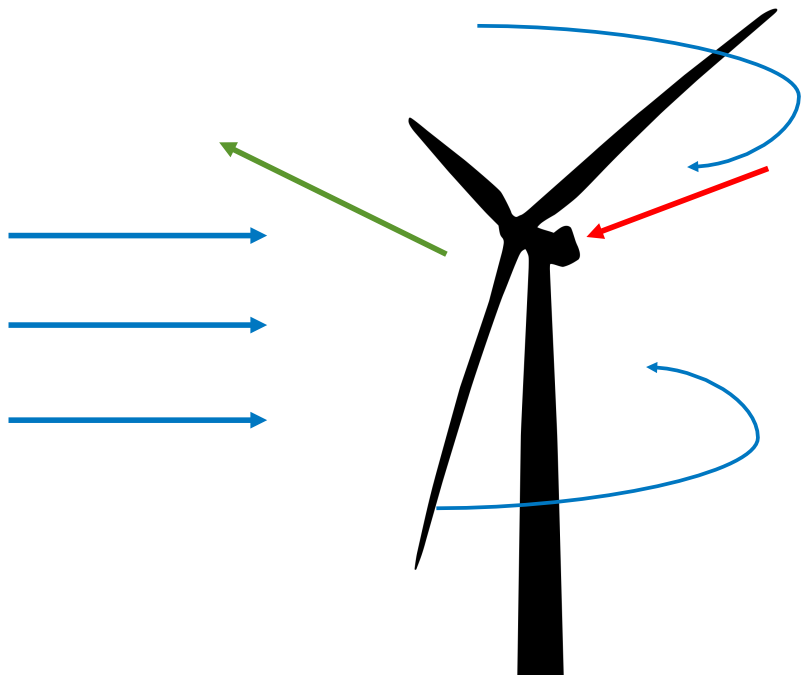


(a) Train (b) Validation

Figure 2.2: Distributed Training Improves Training/Testing Accuracy.

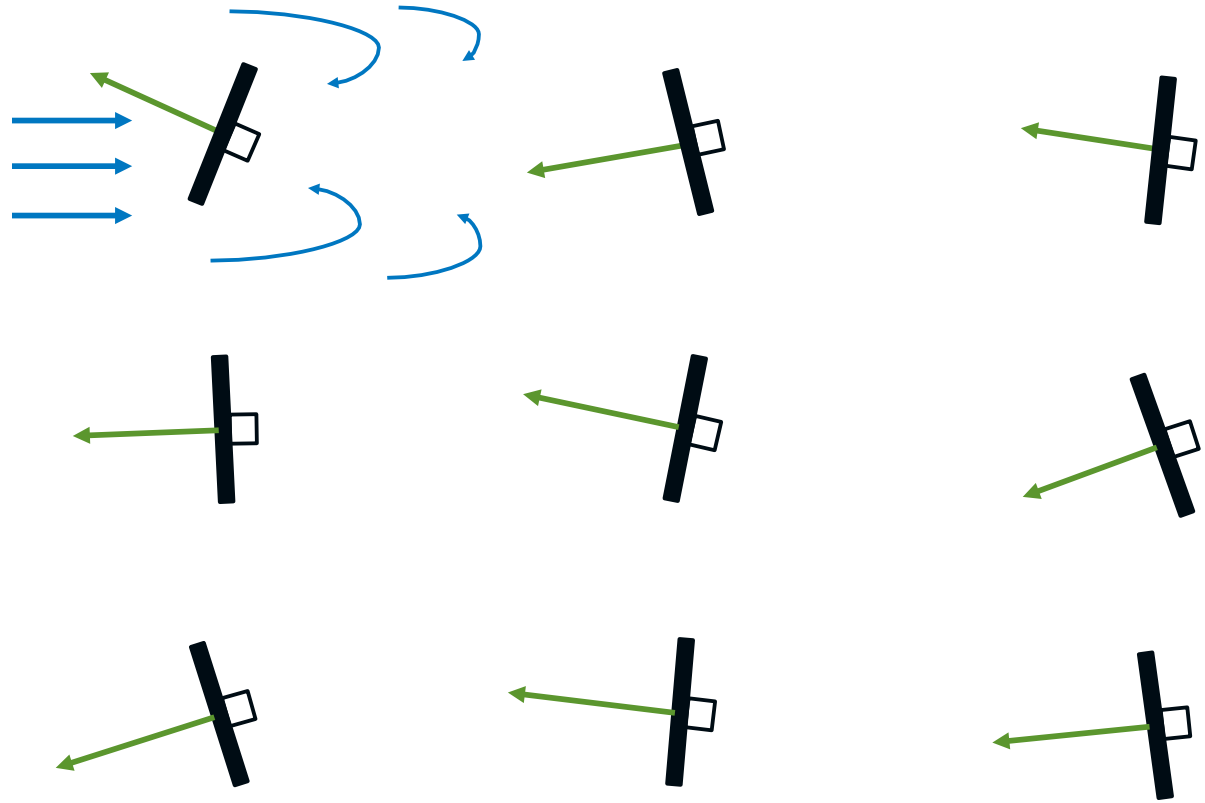
Application 3: Signal Processing, Power Systems

- Consider a wind turbine optimization problem (e.g., yaw angle)
- Traditionally optimized individually



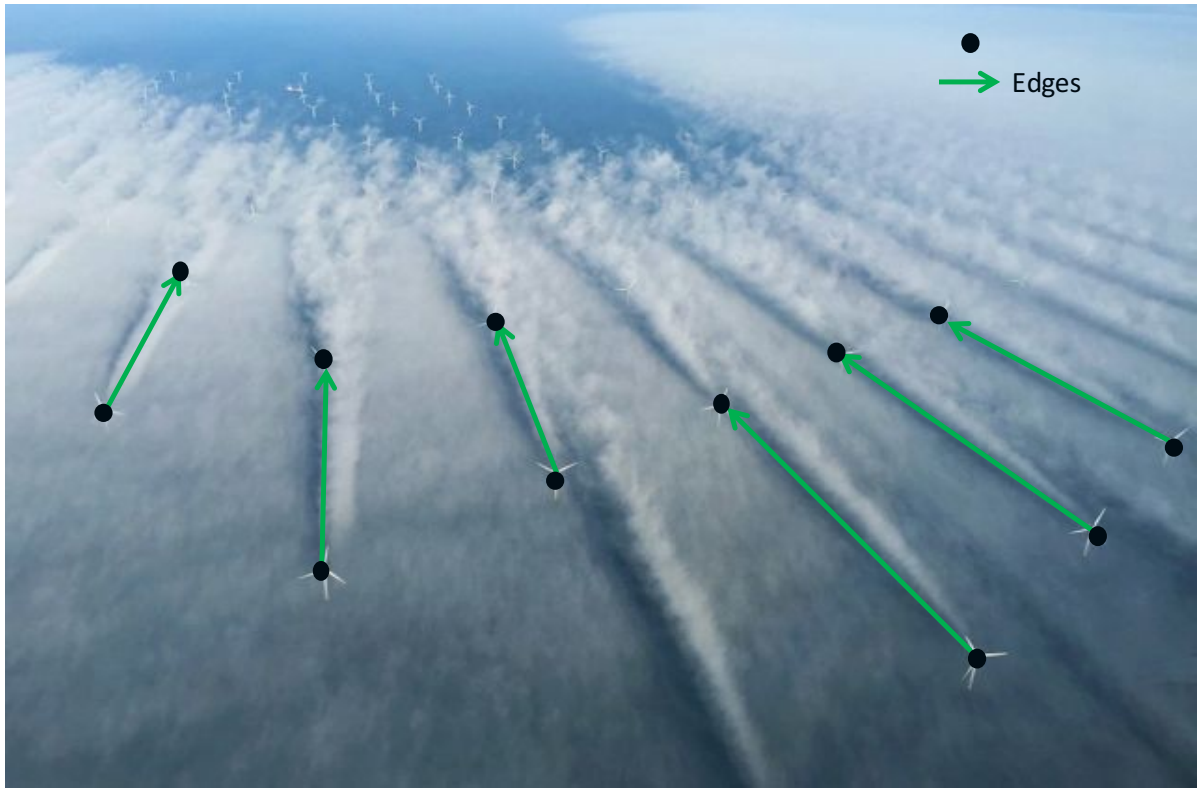
Application 3: Signal Processing, Power SystemsP

- But in typical setting, many wind turbine are collocated together



Application 3: Signal Processing, Power Systems

- But in typical setting, many wind turbine are collocated together



Application 3: Signal Processing, Power Systems

- Individual optimization can be highly suboptimal

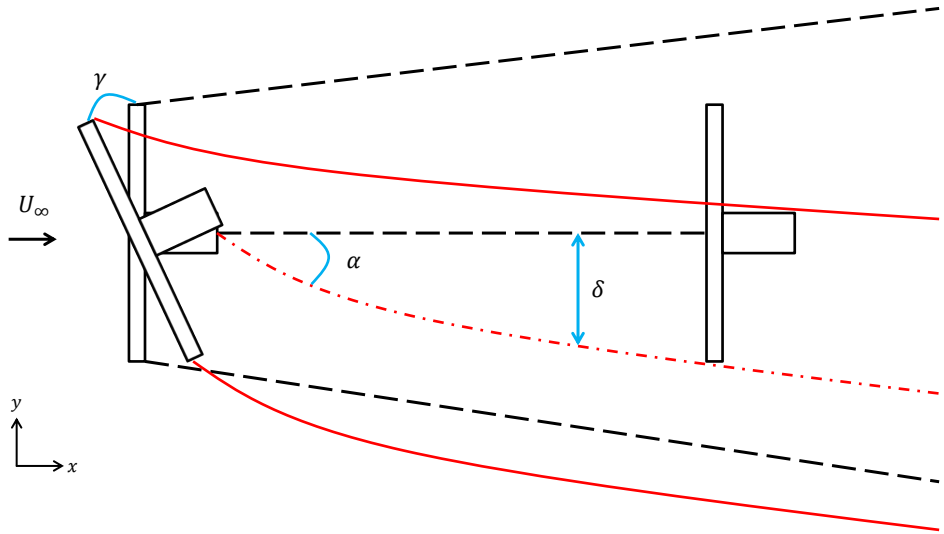


Figure 2.3: Two-turbine example of wake steering control

- No coordination between the up- and down stream turbines can cause
  1. Lose power - yaw misalignment
  2. Constantly yawing - noisy wind vane signal

Application 3: Signal Processing, Power Systems

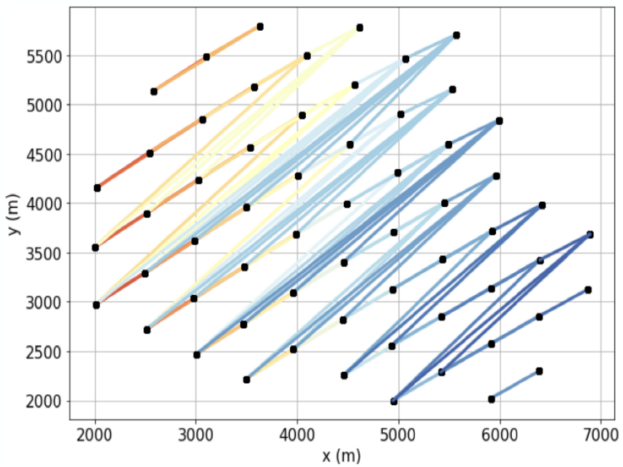
- **Solution**: Make the turbines connected
- Increased communication and coordination
  1. Consensus between turbines on direction
  2. Aligned with wind: increase power capture
  3. Robust wind direction measurement
  4. ....

Application 3: Signal Processing, Power Systems

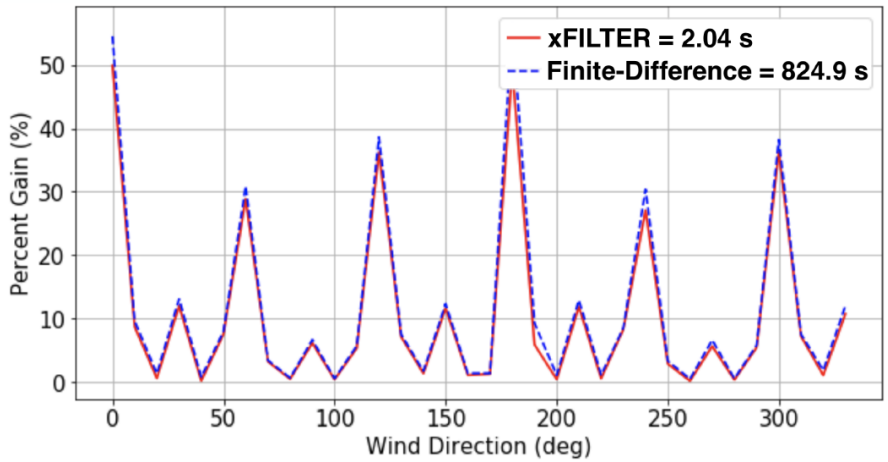
- **Challenge**: Individual optimization problem is complex
- The dynamics and power output of each turbine is a complicated function of parameters (such as yaw angles, wind speed, etc)
- Input/output relationships highly non-trivial (non-convex); they are usually simulated
    1. NREL's wind power simulation and modeling software (`https://www.nrel.gov/wind/data-tools.html`)

Application 3: Signal Processing, Power Systems

- Decentralized non-convex optimization significantly improvesthe practical performance [Annoni et al. 19]
- Princess Amalia Wind Farm [Fleming et al, 2016]
- Simulated 60 NREL's 5MW turbines [Jonkman et al, 2009]



Graph structure of wind farm



100X faster than centralized method

Application 4: Federated Learning (FL)

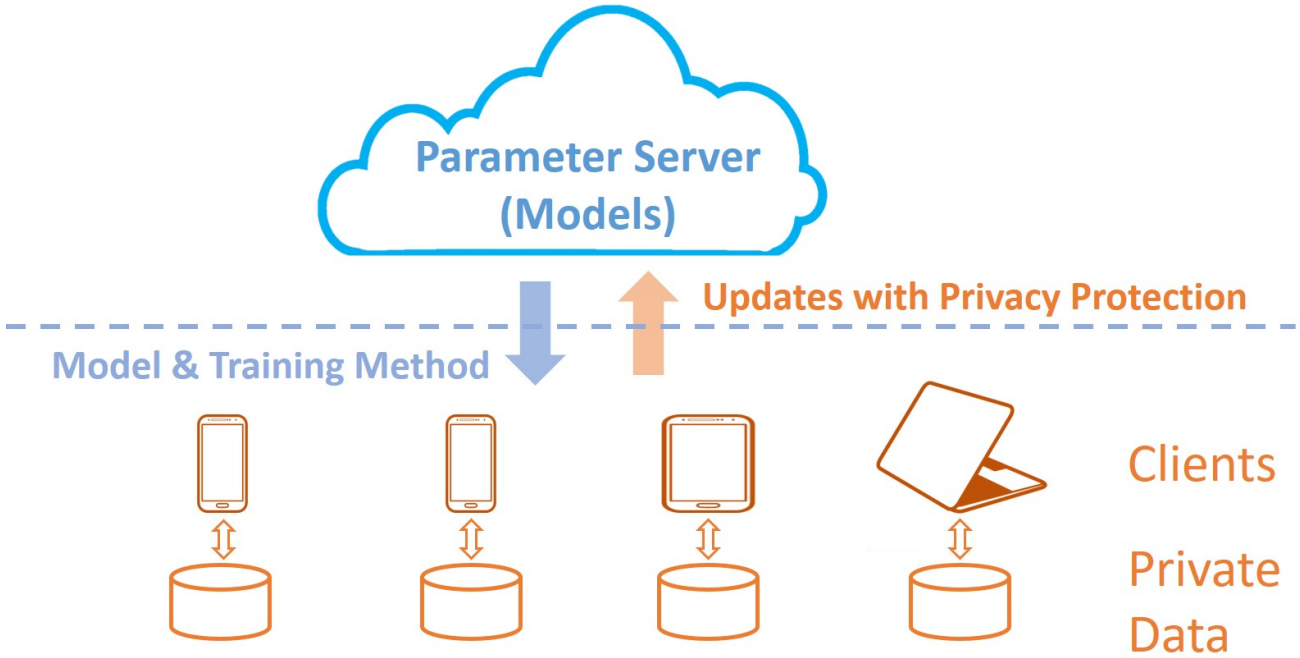
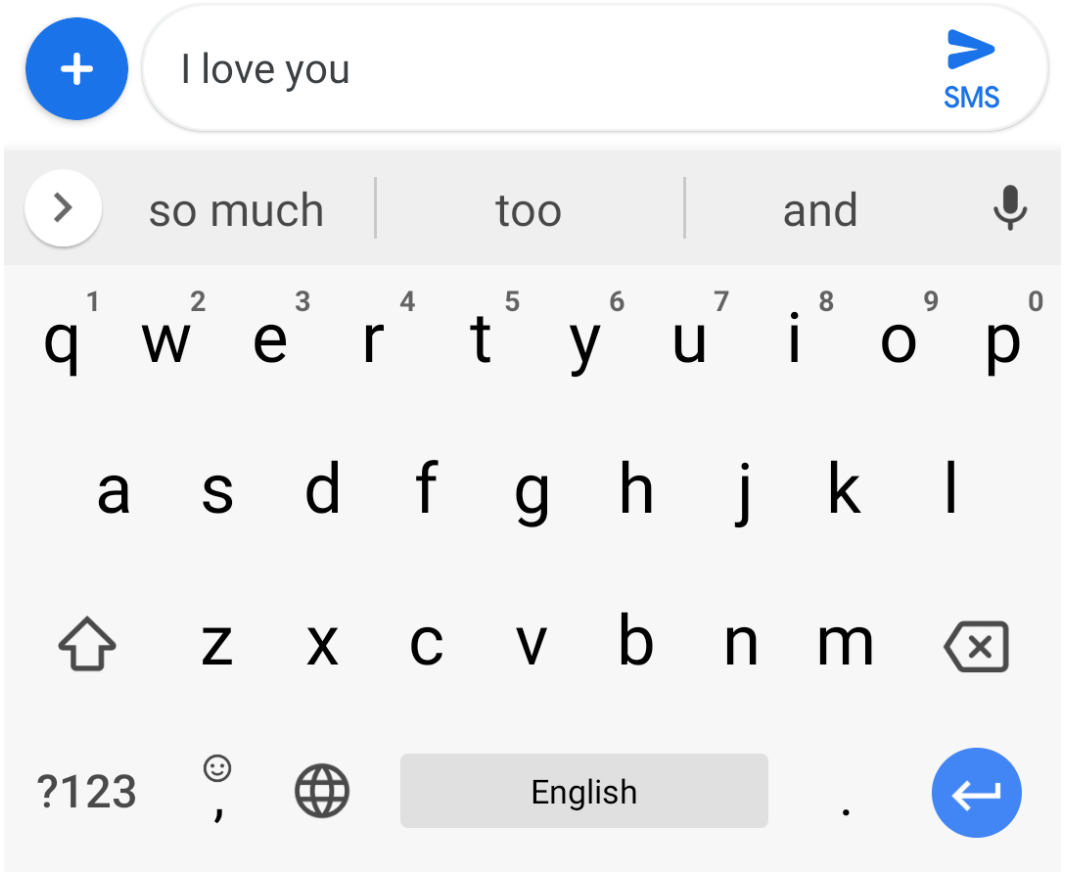


Figure 2.4: System structure of federated learning

- Parameter server network
- Massivly distributed data
- Communication compression

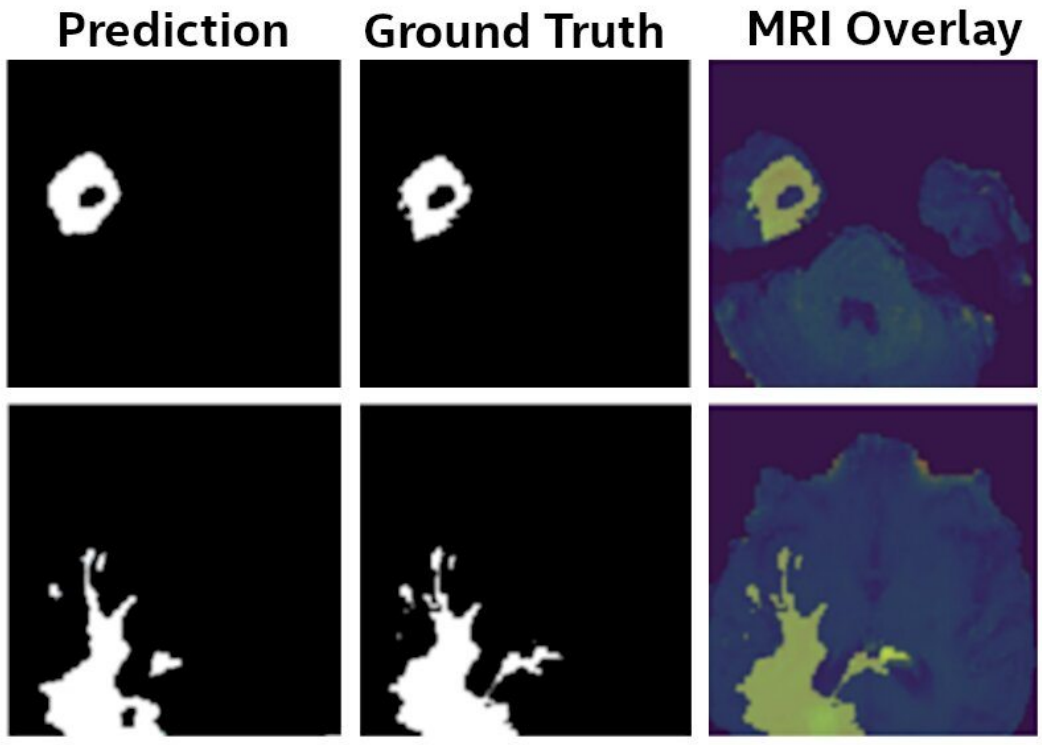
Application 4: Federated Learning (FL)

Figure 2.5: Keyboard Prediction



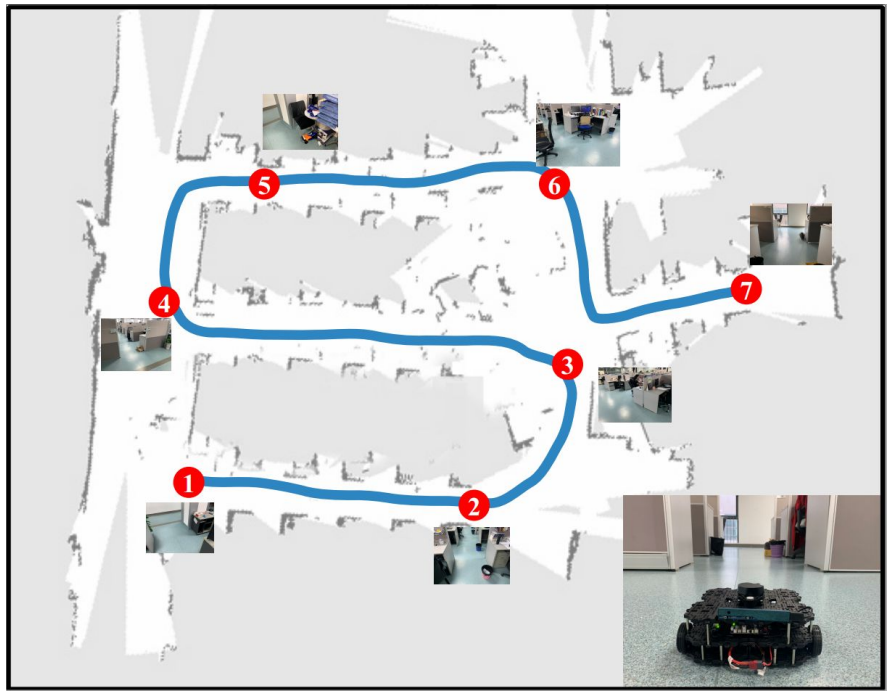
Application 4: Federated Learning (FL)

Figure 2.6: Medical Imaging



Application 4: Federated Learning (FL)

Figure 2.7: Robot Control



Generic Formulation

Generally speaking, decentralized/distribution optimization problem can be formulated into the following optimization problem:

$$\begin{aligned} \text{minimize}_{\boldsymbol{x},\boldsymbol{y}} \quad & \sum_{i=1}^{m} f_i(x, y_i) \\ \text{subject to} \quad & x \in X,\ y_i \in Y \end{aligned}$$

- Each $f_i : \mathbb{R}^n \to \mathbb{R}$ is a local loss function only available to agent i; this loss function is related to local data (i.e., in the ML training example, each f_i is related to local training data)
- x is the variables shared among all the agents (i.e., in the FL application, this is the model to be trained to predict the next entry)
- Each y_i is a local variable that is only relevant to agent i (i.e., in the power system application, this is the yawn angle for each individual system)

Generic Formulation

Key questions to be asked:

- How to allow the agents to jointly optimize the objective without sharing the data
- What kind of solution can we get, and how fast can we obtain these solutions
- What if the connectivity among the agents are changing all the time?
- How to reduce communication burden?
- What is the data is streaming into each agent?
- ....