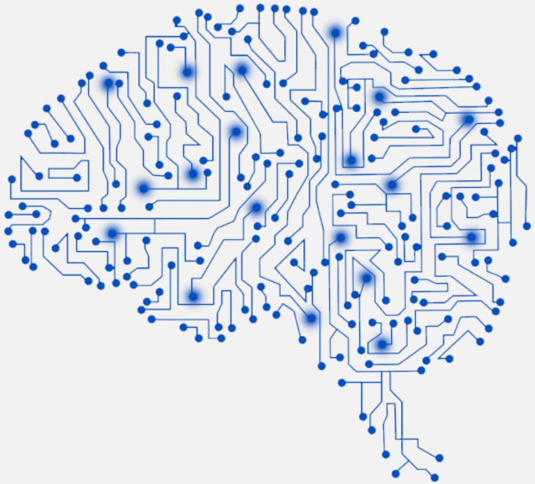


Deep Reinforcement Learning

Prof. Joongheon Kim
Korea University, Seoul, Korea
https://joongheon.github.io
joongheon@korea.ac.kr

Lecture Roadmap

Introduction and Preliminaries

Deep Reinforcement Learning Theory

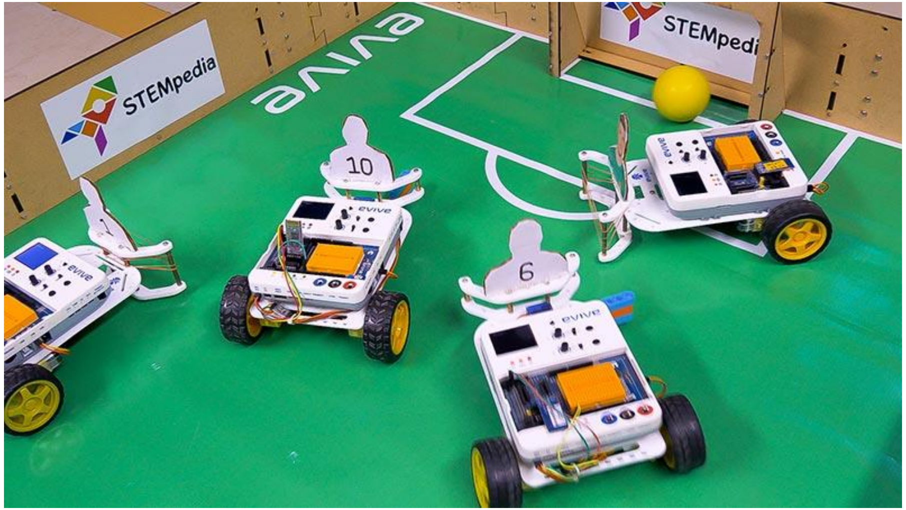
Deep Reinforcement Learning Implementation

Imitation Learning and Autonomous Driving

- Brief History and Successes
  - Minsky's PhD thesis (1954): Stochastic Neural-Analog **Reinforcement** Computer
  - Analogies with animal learning and psychology
  - Job-shop scheduling for NASA space missions (Zhang and Dietterich, 1997)
  - Robotic soccer (Stone and Veloso, 1998) – part of the world-champion approach

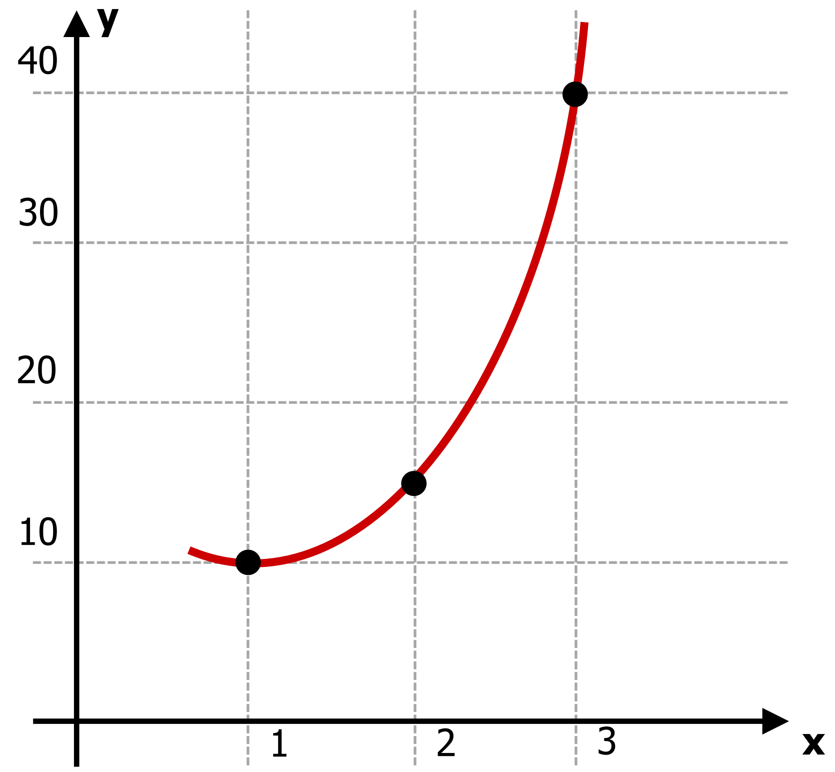
- When RL can be used?
  - Find the (approximated) **optimal action sequence** for **expected reward maximization** **(not for single optimal solution)**
  - Define **actions** and **rewards**. These are all we need to do.

- Action Sequence (also called **Policy**, later in this presentation)!

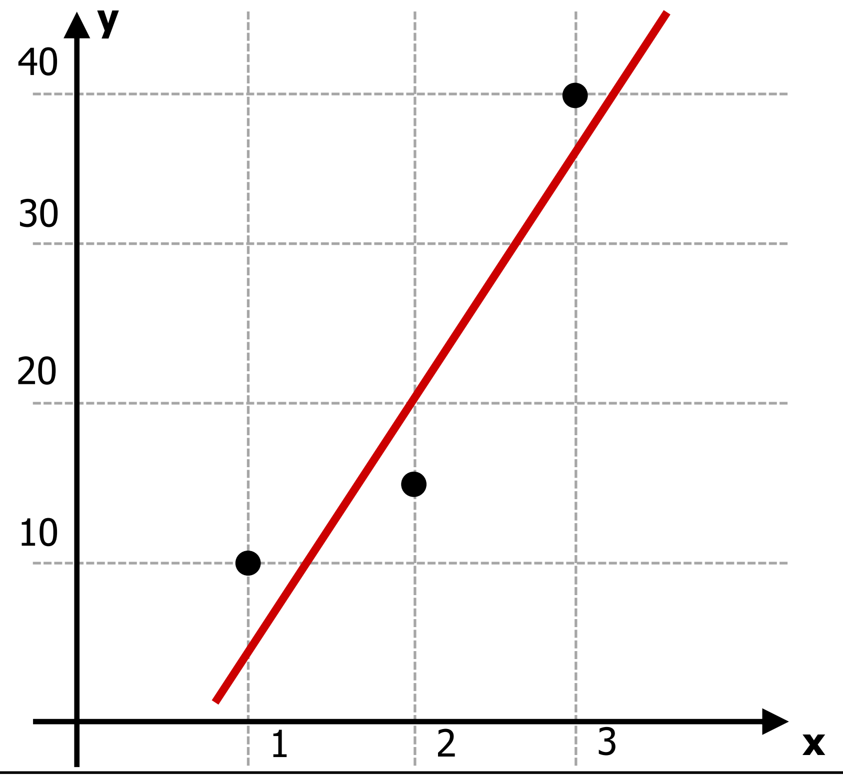


Interpolation vs. Linear Regression

Interpolation

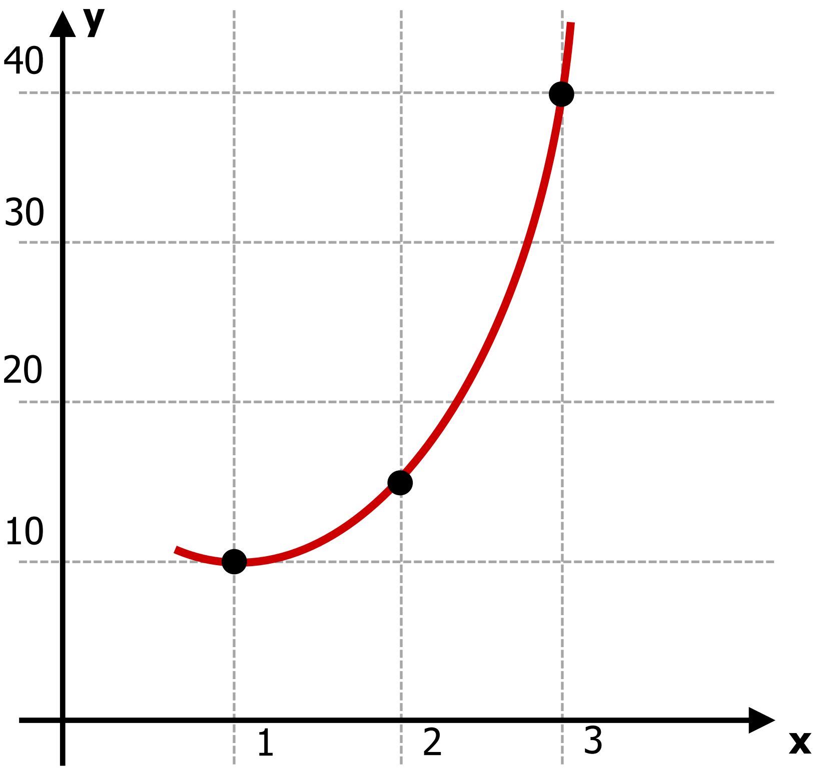


Linear Regression



Interpolation vs. Linear Regression

Interpolation



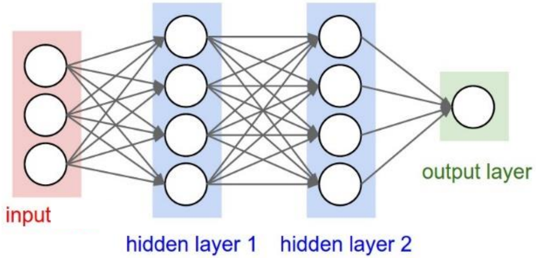
Interpolation with Polynomials

$$y = a_2 x^2 + a_1 x^1 + a_0$$

where three points are given.
→ Unique coefficients (a_0, a_1, a_2) can be calculated.

Is this related to
Neural Network Training?

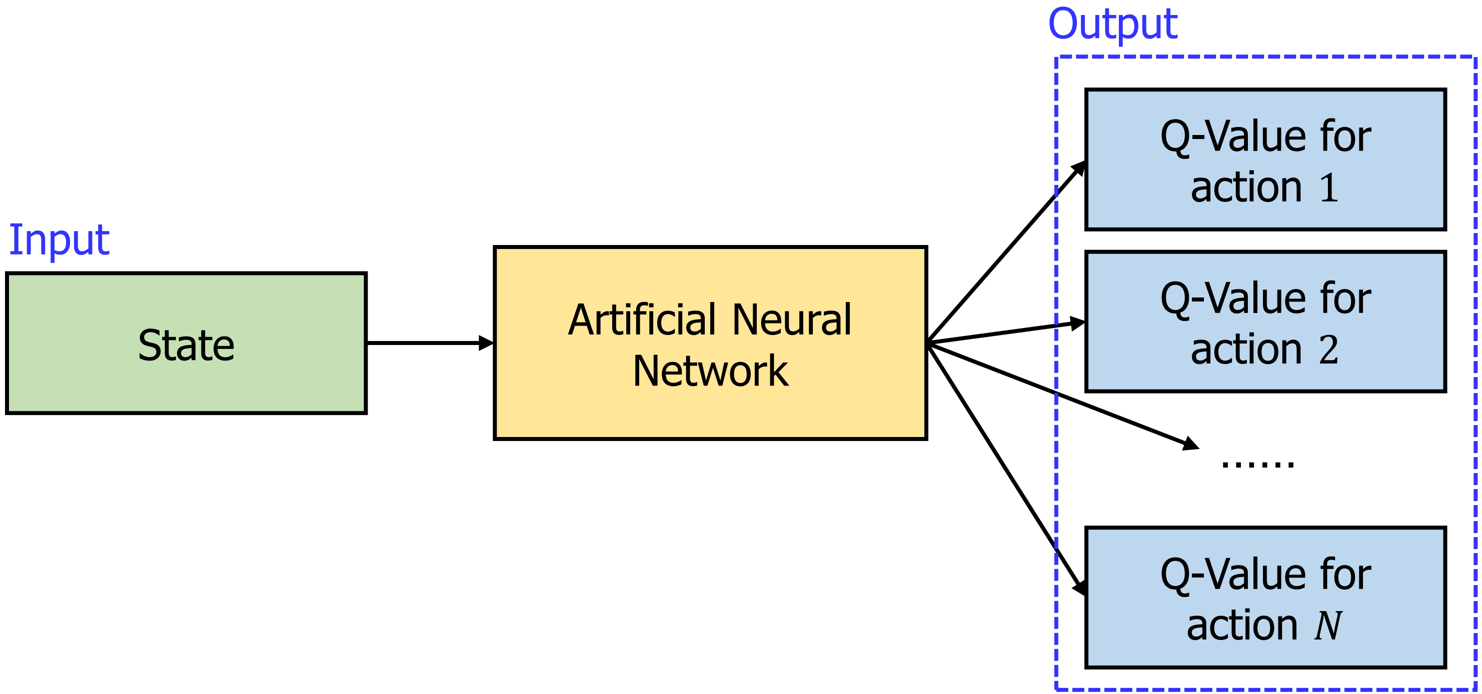
Interpolation and Neural Network Training



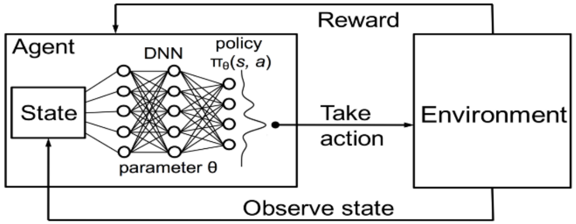
$$Y = a(a(a(X \cdot W_1 + b_1) \cdot W_2 + b_2) \cdot W_o + b_o)$$

where training data/labels (X: data, Y: labels) are given.
→ Find $W_1, b_1, W_2, b_2, W_o, b_o$
→ This is the mathematical meaning of neural network training.
→ **Function Approximation**
→ The most well-known function approximation with neural network: **Deep Reinforcement Learning**

- It is inefficient to make the Q-table for each state-action pair.
  → ANN is used to **approximate the Q-function**.



Deep Reinforcement Learning for Vehicles

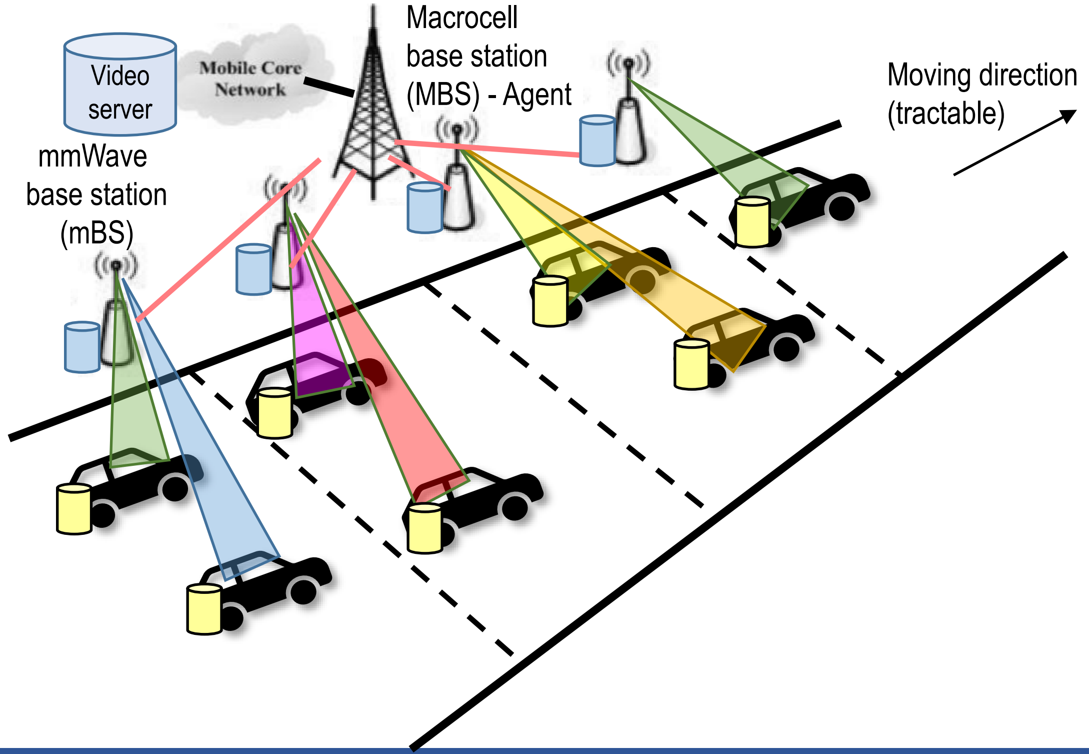


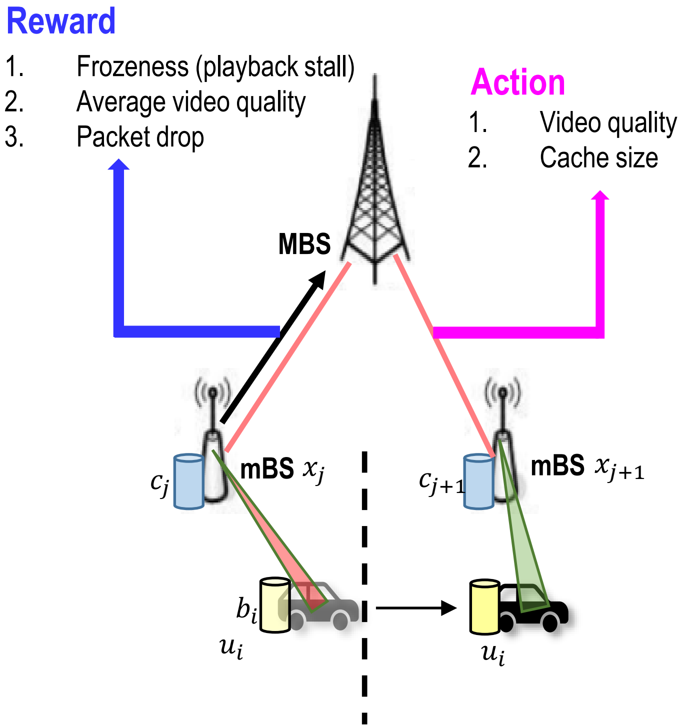
Deep Reinforcement Learning

Proactive Automotive/Vehicular Caching in I2V Infra

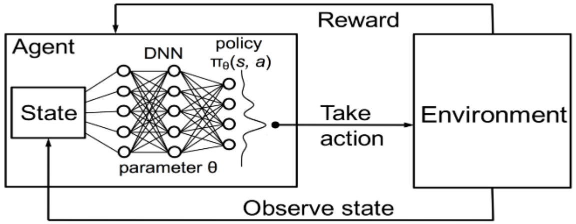
Multi-UAV Coordination for 5G Network Coverage Extension

- **Proactive Automotive/Vehicular Caching in I2V Infra**
  - DDPG Modeling (Rewards and Actions)





Deep Reinforcement Learning for Vehicles

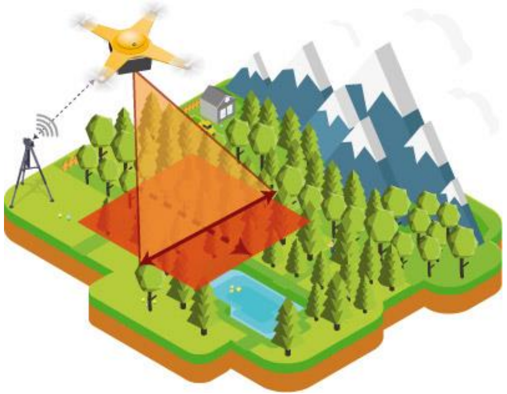


Deep Reinforcement Learning

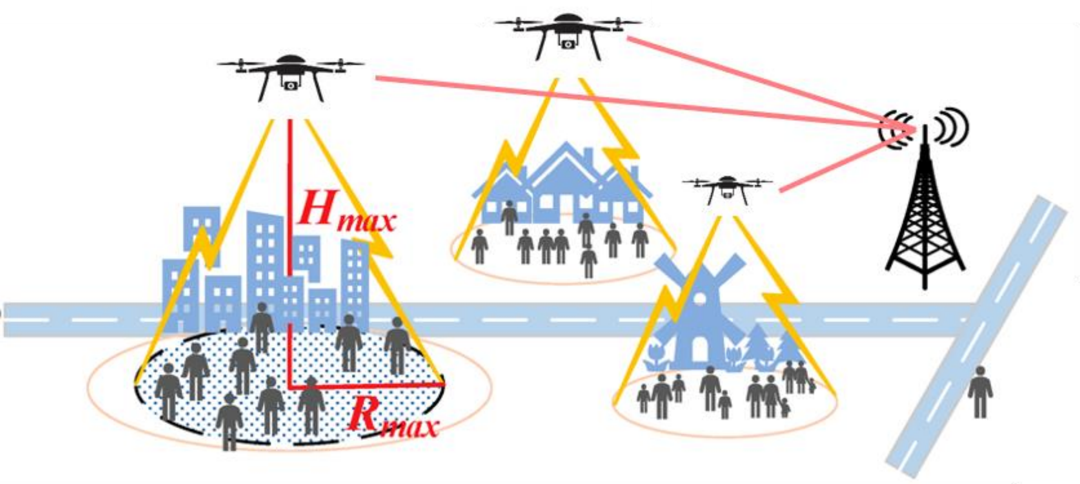
Proactive Automotive/Vehicular Caching in I2X Infra

Multi-UAV Coordination for 5G Network Coverage Extension

- **[2] Multi-UAV Coordination for 5G Network Coverage Extension**
  - Motivation
    - **NLOS and blockage effects** are major components which can make impacts on the performance of 5G millimeter-wave wireless technologies.
    - **Deploying UAV-assisted wireless networks** can be an effective solution to mitigate this issue as it enables **more LOS communications.**
  - Proposed Solution
    - **Reinforcement learning** based **multi-agent navigation** algorithm.



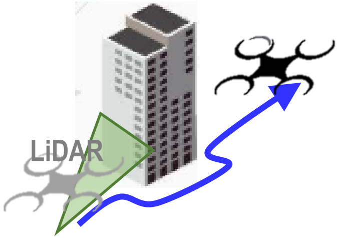
- **[2] Multi-UAV Coordination for 5G Network Coverage Extension**
  - System Model



Obstacle Avoidance (in Complex Environment)

- **LiDAR**

→ A remote sensing technology that uses rapid laser pulses to locate nearby obstacles.

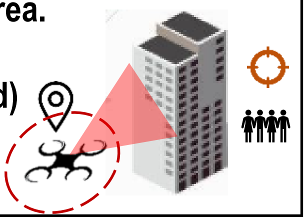


- **Path planning**

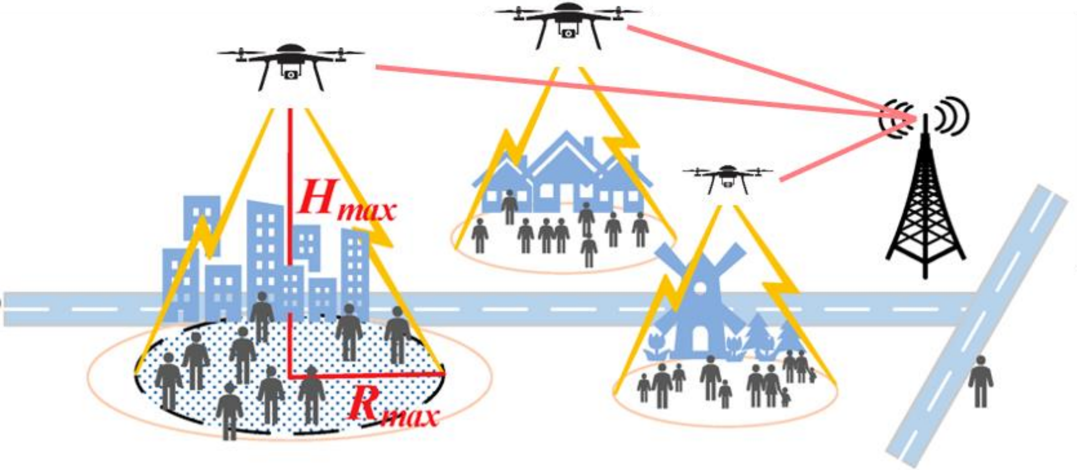
→ Drone sets a new path to avoid obstacles.

State

- **Location of drone.**
- **Location of shaded area.**
- **LiDAR (360 degree & forward)**



- **[2] Multi-UAV Coordination for 5G Network Coverage Extension**
  - System Model



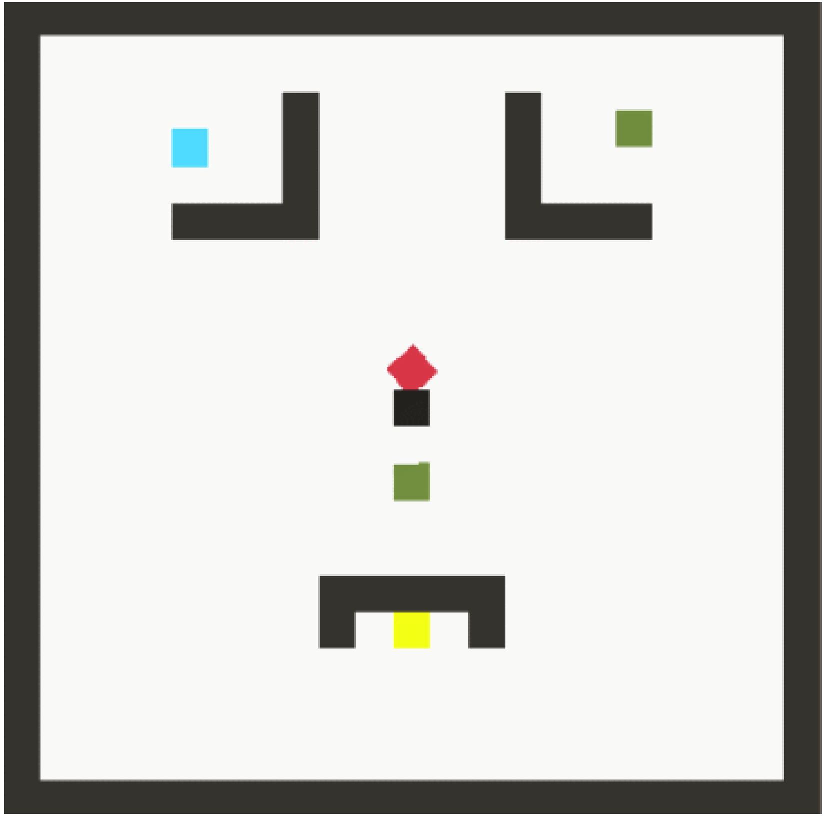
Reward

1. Collision with obstacles **(negative)**
2. Collision with drones **(negative)**
3. Arrive at destination **(positive)**
4. LOS communication **(positive)**

Action

1. Rotation (Left, Right)
2. Move forward

- **[2] Multi-UAV Coordination for 5G Network Coverage Extension**
  - Simple Demo (Bird View)



- ICML 2018 Tutorial
  - https://sites.google.com/view/icml2018-imitation-learning/

Imitation Learning Tutorial ICML 2018

- ICML 2019 Tutorial
  - https://slideslive.com/38917941/imitation-prediction-and-modelbased-reinforcement-learning-for-autonomous-driving

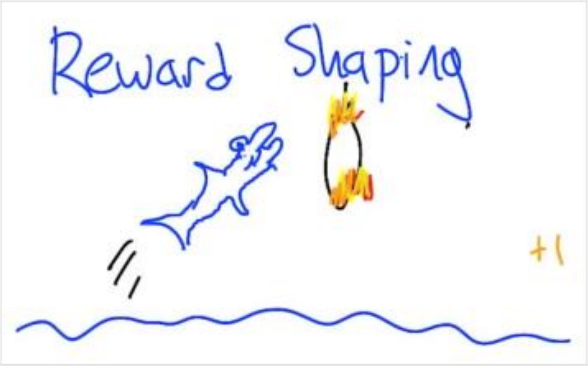
- Gameplay

Pro-Gamer

Trained Agent

The goal of Imitation Learning is to train a policy to mimic **the expert's demonstrations**

- Problems of RL



1. Reward Shaping
2. Safe Learning
3. Exploration process

Imitation Learning **handles with** these problems through the demonstration of the experts.

Inverse Reinforcement Learning (IRL)

Environment Model(MDP)

RL

Reward Function R → Reinforcement Learning $\arg\max_\pi \mathrm{E}[\sum_t \gamma^t R(s_t)|\pi]$ → Optimal Policy π

Environment Model(MDP)

IRL

Reward Function R ← Inverse Reinforcement Learning ← Optimal Policy π

Expert Policy

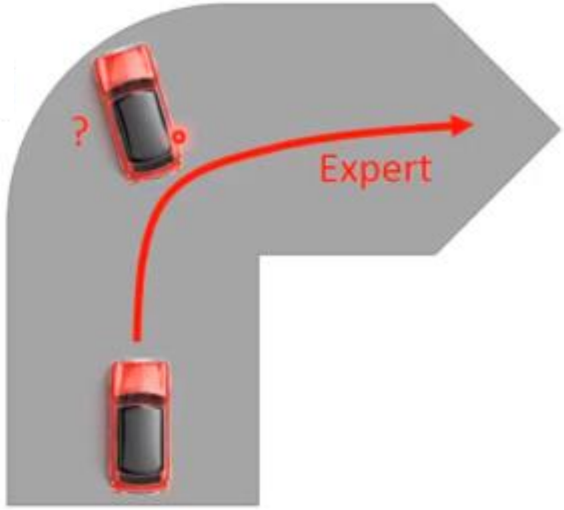
R that explains Expert Trajectories

Expert Trajectories $s_0, a_0, s_1, a_1, s_2, a_2, \cdots$

- **Behavior Cloning**
  - Define $P^* = P(s|\pi^*)$ (distribution of states visited by **expert**)
  - **Learning objective**

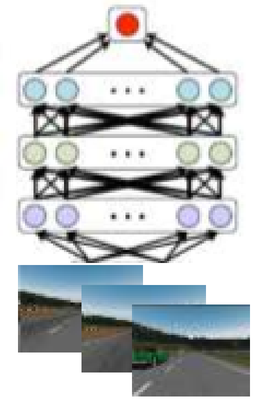
$$argmin_{\boldsymbol{\theta}} E_{(s,a_E)\sim P^*} L(a_E, \pi_{\boldsymbol{\theta}}(s))$$
$$L\left(a_E, \pi_{\boldsymbol{\theta}}(s)\right) = \left(a_E - \pi_{\boldsymbol{\theta}}(s)\right)^2$$

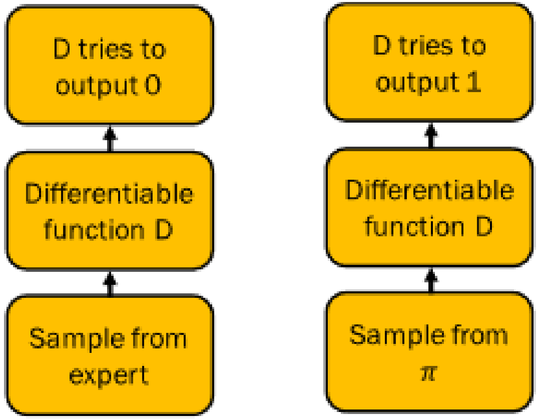
  - **Discussion**
    - Works well when P^* close to the distribution of states visited by $\pi_{\boldsymbol{\theta}}$
    - **Minimize 1-step deviation error** along the expert trajectories



- **Generative Adversarial Imitation Learning (GAIL)**, **NIPS 2016**
  - Generative adversarial imitation learning (GAIL) learns a policy that can imitate expert demonstration using **the adversarial network** from generative adversarial network (GAN).
  - **Learning Objective**

$$\boxed{argmin_{\theta}\ argmax_{\emptyset}\ E[\log(D_{\emptyset}(s,a)] + E[\log(1 - D_{\emptyset}(s,a))]}$$





Imitation Learning Applications: Starcraft2

- Starcraft2

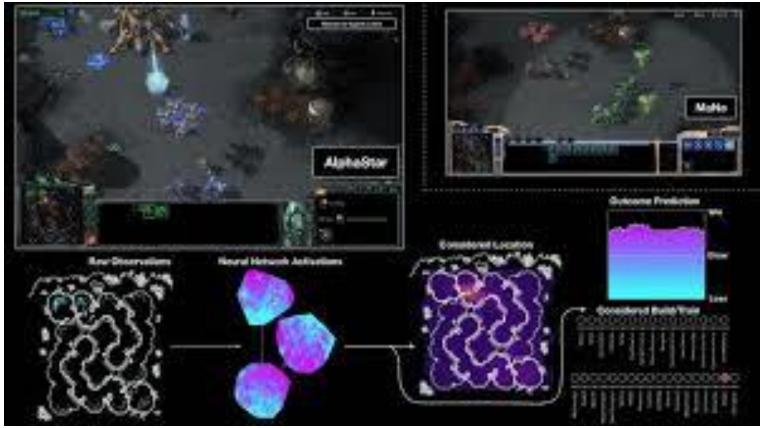
States: s = **minimap**, **screen**
Action: a = **select**, **drag**
Training set: $D = \{\tau \coloneqq (s, a)\}$ from expert
Goal: learn $\pi_\theta(s) \rightarrow$ a

States: s
Action: a
Policy: π_θ
- Policy maps states to actions : $\pi_\theta(s) \rightarrow$ a
- Distributions over actions : $\pi_\theta(s) \rightarrow P$(a)

State Dynamics: P(s′|s,a)
- Typically not known to policy
- Essentially the simulator/environment

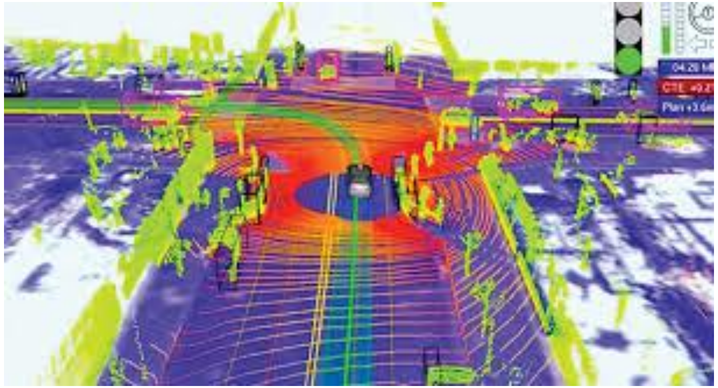
Rollout: sequentially execute $\pi_\theta(s_0)$ on initial state
- Produce trajectories $\boldsymbol{\tau}$

$\mathbf{P}(\boldsymbol{\tau}|\boldsymbol{\pi})$**:** distribution of trajectories induced by a policy
$\mathbf{P}(\mathbf{s}|\boldsymbol{\pi})$**:** distribution of states induced by a policy



- Autonomous Driving Control

States: s = **sensors**
Action: a = **steering wheel**, **brake**, …
Training set: $D = \{\tau \coloneqq (s, a)\}$ from expert
Goal: learn $\pi_\theta(s) \rightarrow$ a



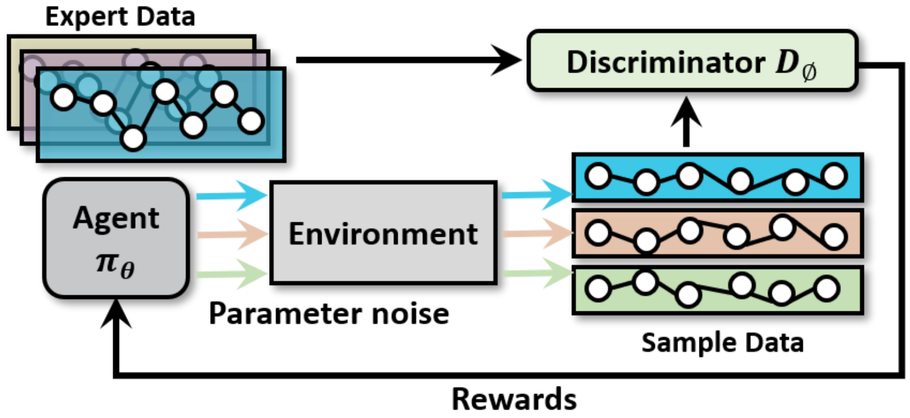
- Smartphone Security

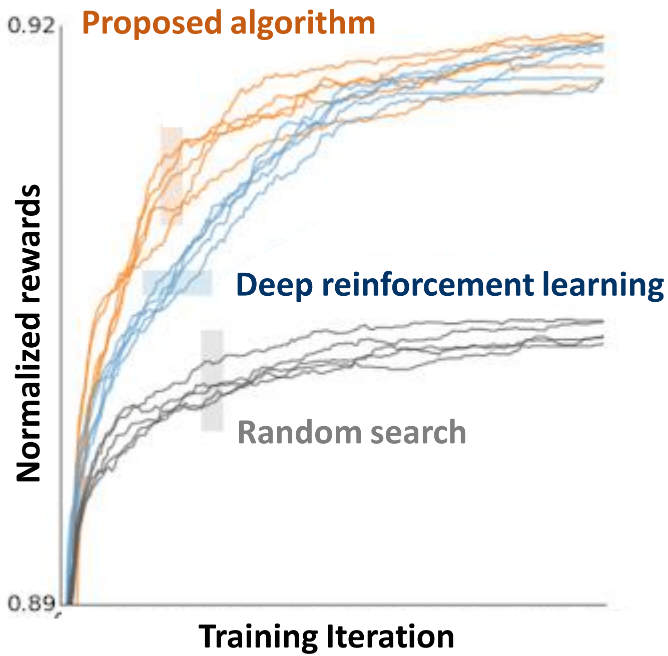
States: s = **apps**, …
Action: a = **use patterns**, …
Training set: $D = \{\tau \coloneqq (s, a)\}$ from expert
Goal: learn $\pi_\theta(s) \to$ a

- PPF/RFTN Injection Control in Medicine

States: s = **BIS**, **BP**, …
Action: a = **PPF**, **RFTN**, …
Training set: $D = \{\tau \coloneqq (s, a)\}$ from expert
Goal: learn $\pi_\theta(s) \rightarrow$ a

Autonomous Driving with Imitation Learning



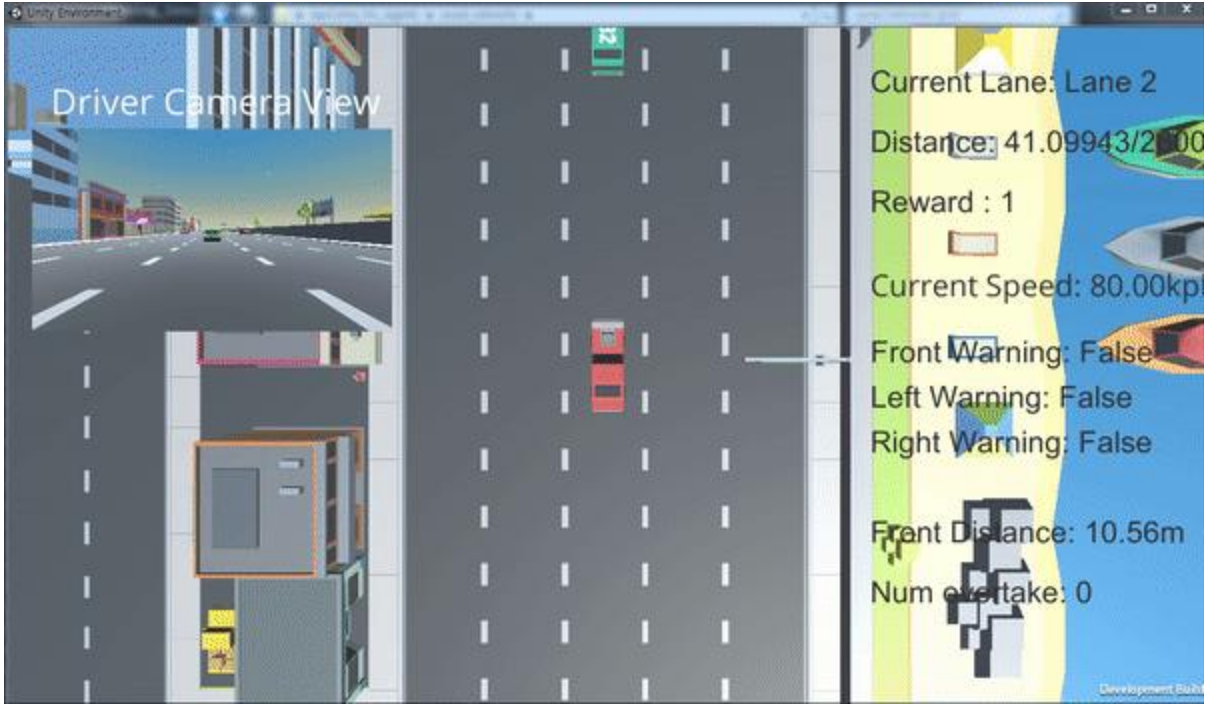


M. Shin and J. Kim, "**Adversarial Imitation Learning via Random Search in Lane Change Decision-Making**," *ICML 2019 Workshop on AI for Autonomous Driving*, 2019.

M. Shin and J. Kim, "**Randomized Adversarial Imitation Learning for Autonomous Driving**," *IJCAI*, 2019., (Acceptance Rate: 850/4752=17.89%)

Generative Adversarial Network (GAN) + **Random Search**
for Autonomous Driving

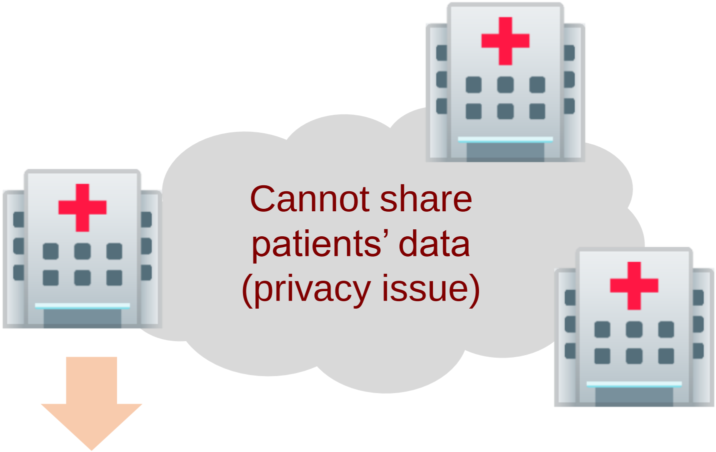
Autonomous Driving with Imitation Learning



Outline

- GAN Introduction
- Reinforcement Learning and Imitation Learning
- **Federated Learning**

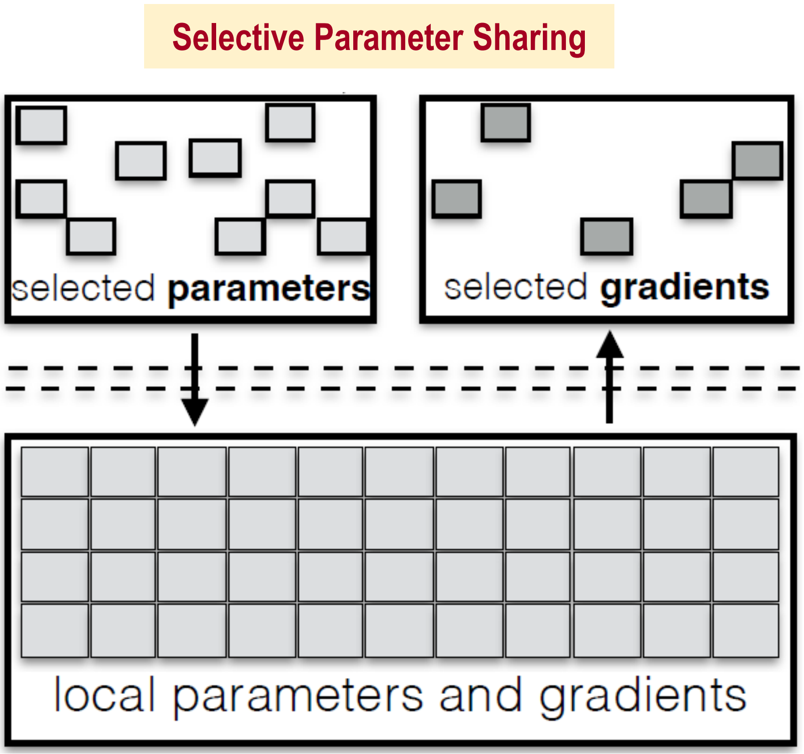
- **Motivation**
  - It's not possible to gather all data in a single hospital/medical-cloud for deep learning computation (due to patients' privacy).
    Then, following problems can occur:
    - **Overfitting** in each hospital
    - **Training Performance Degradation**
  - More serious problems can happen…



Goals

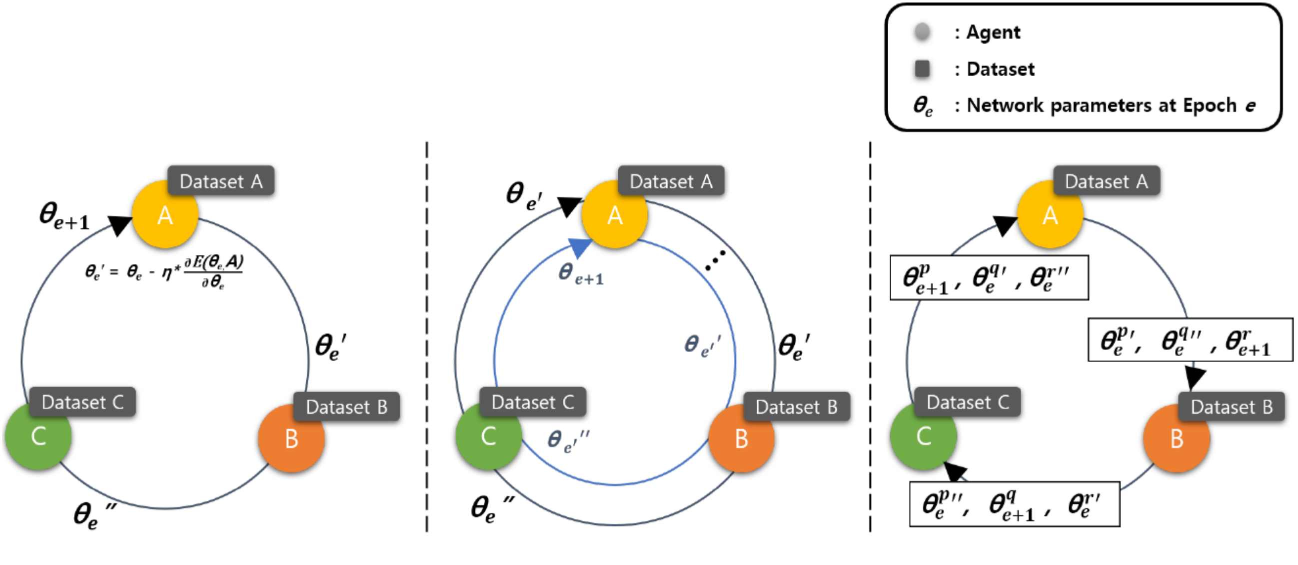
- Maintaining Deep Learning Computation Performance
- Prohibiting Duplicated Patients' Data

- **Collaborative Deep Learning**
  - How it works?
    - All clouds share the model at first.
    - Each cloud trains its own model (Data is not shared among clouds for privacy-preserving).
    - Each cloud shares weight values (not the data itself). → **Selective Parameter Sharing**
  - Disadvantages
    - Performance degradation
    - Synchronization (No network delays are assumed.)



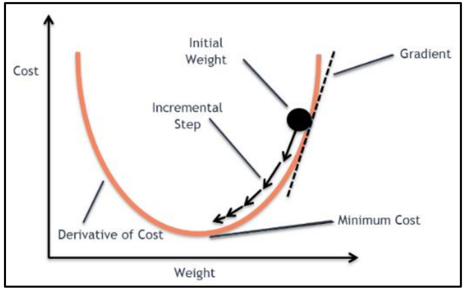
R. Shokri and V. Shmatikov, "Privacy-Preserving Deep Learning," *ACM CCS 2015*. (Citation: 500+)

- Cyclic Parameter Delivery for Distributed Privacy-Preserving Deep Learning



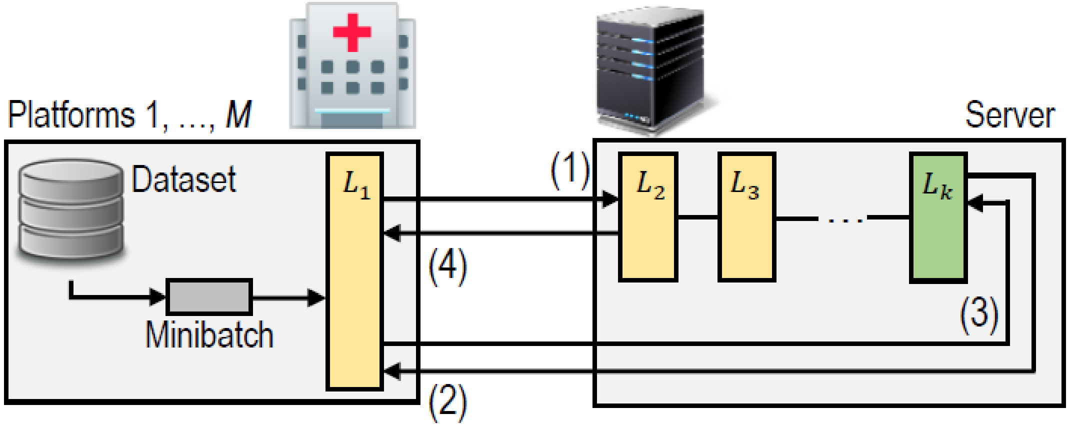
(a) (b) (c)

Cost Minimization



Additional Benefits
- Can Combat the Network Delay/Latency
- Can Combat the Network Data Imbalance

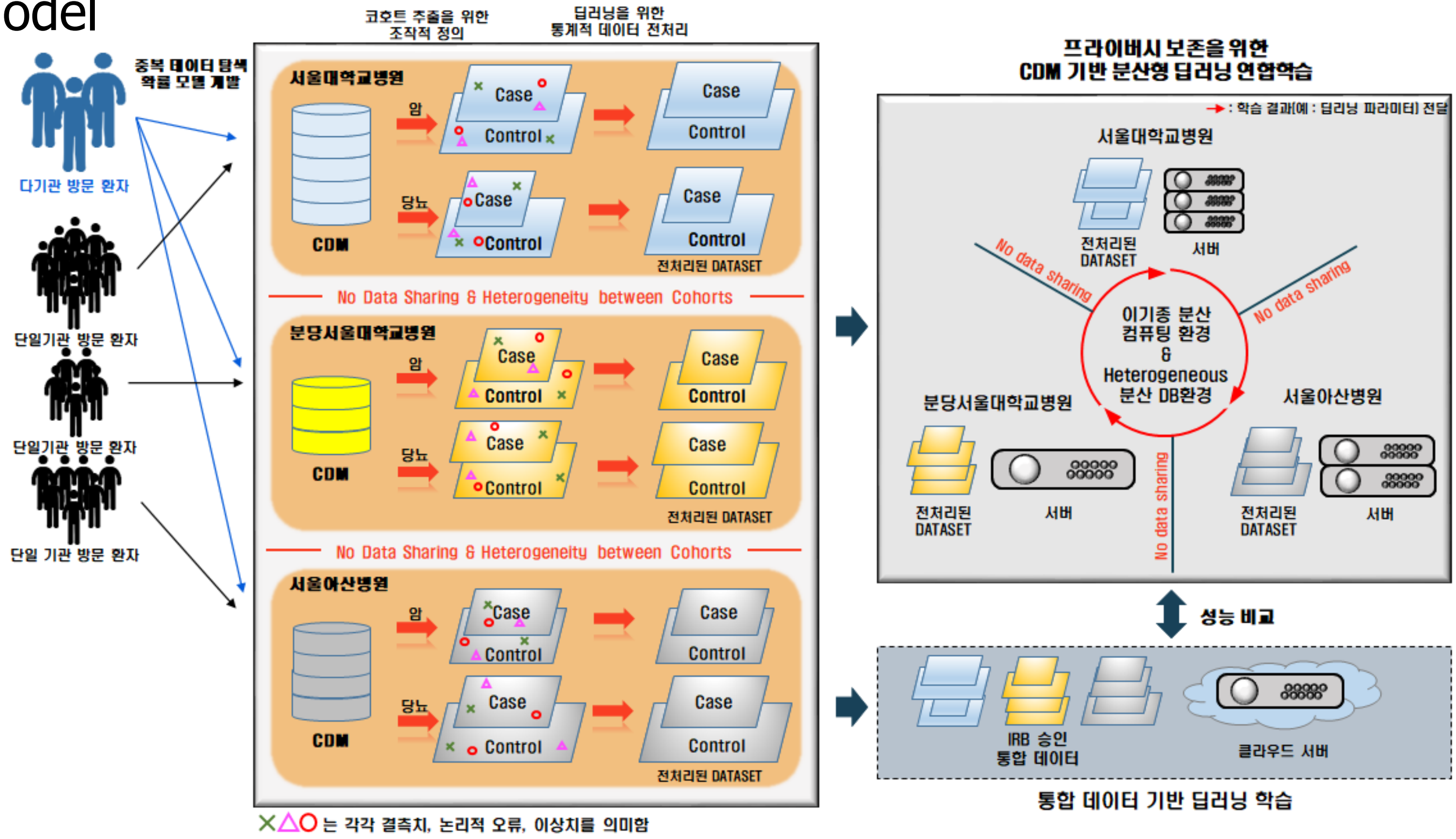
Distributed Federated Learning



- **Privacy-Preserving Distributed Deep Learning Computation**
  - Each platform has the first hidden layer of deep learning model (L_1)
  - Server has the other hidden layers and the output layer (L_2, …, L_{k-1}, L_k)
  - During training process the data is shared in the form of the results of L_1

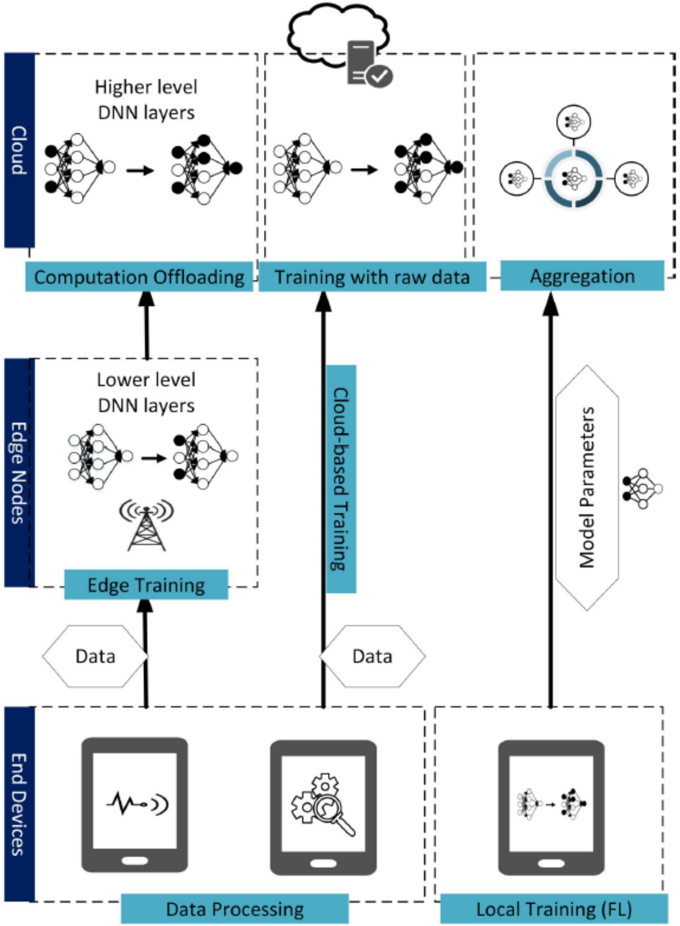
Privacy-Preserving Medical Deep Learning

- System Model



FL Applications to Networks (Paper #2)

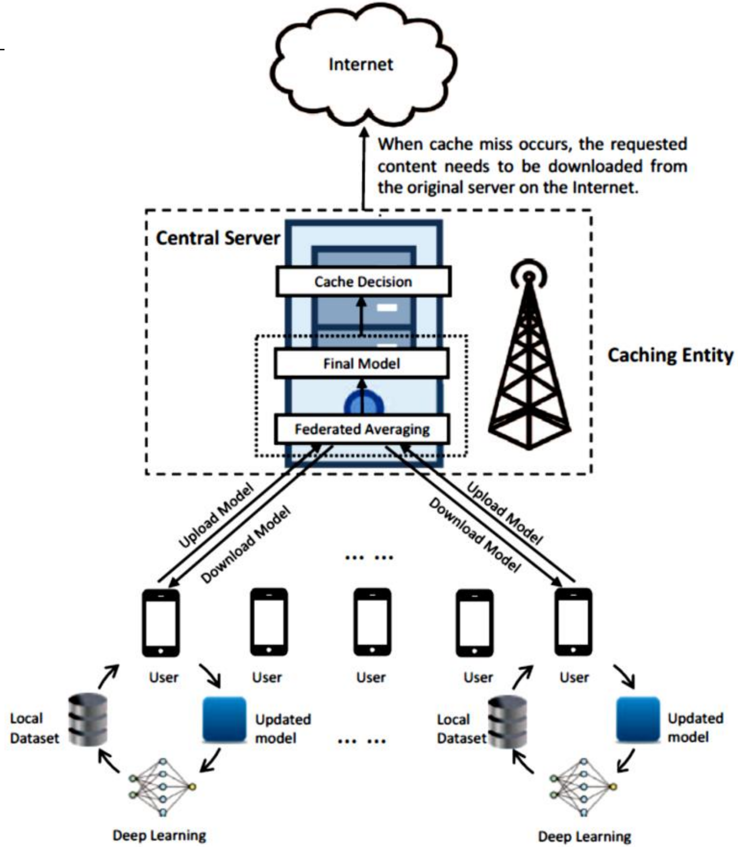
- Edge AI approach brings AI processing closer to where data is produced.
- FL allows training on devices where the data is produced.



W. Yang, *et. al.*, "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," *arXiv:1909.11875v1*, Sept. 2019.

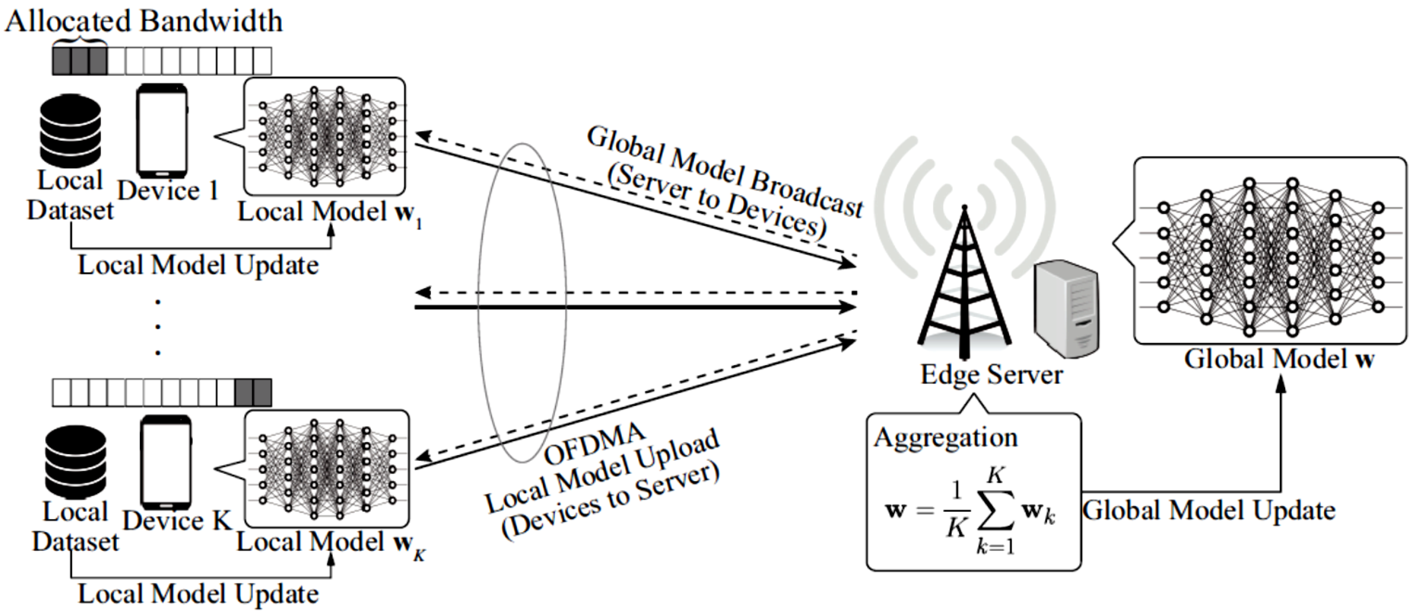
FL Applications to Networks (Paper #3)

- System Model



Z. Yu, J. Hu, G. Min, H. Lu, Z. Zhao, H. Wang, and N. Georgalas, "Federated Learning Based Proactive Content Caching in Edge Computing," in *Proc. of IEEE GLOBECOM*, Abu Dhabi, UAE, Dec. 2018.

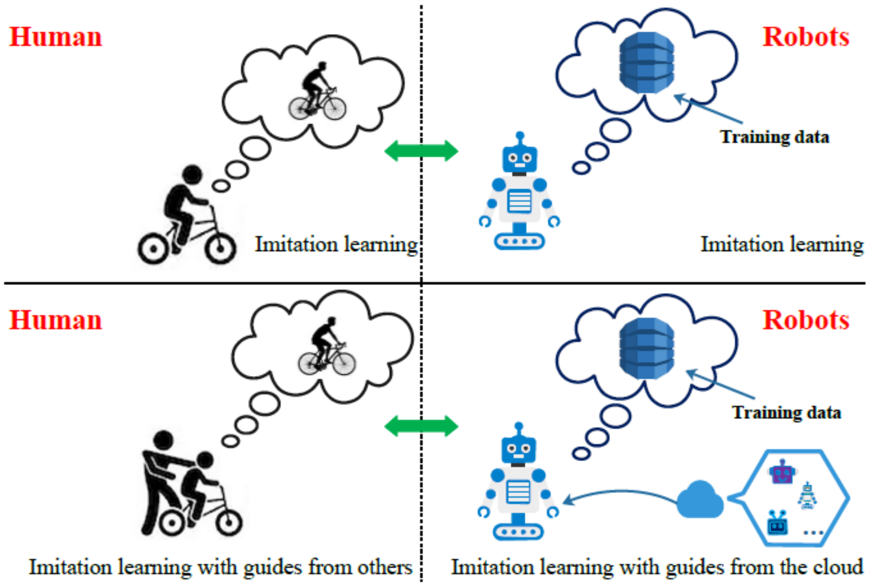
- Energy-Efficient FL System

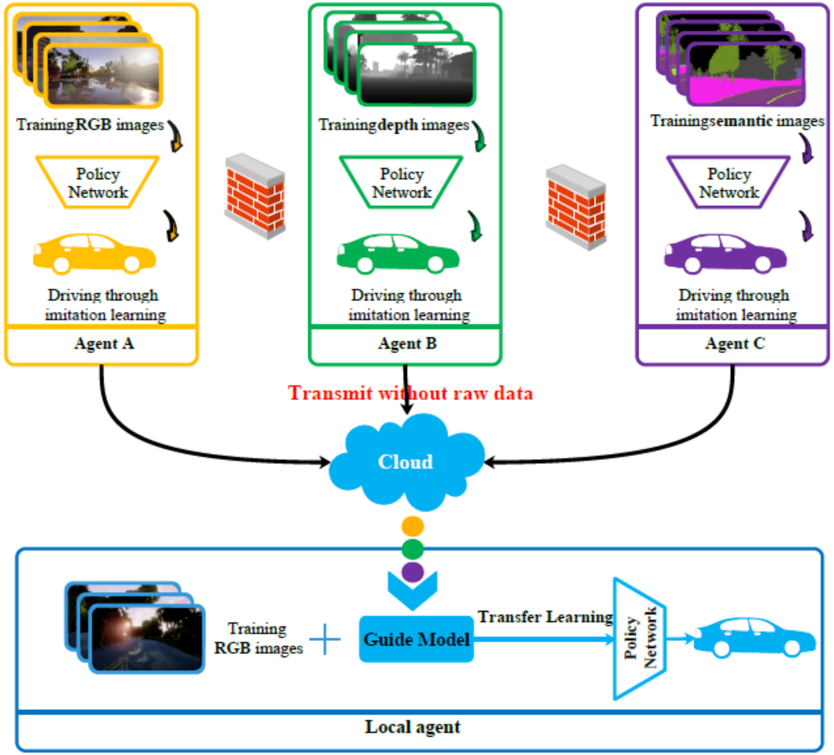


Q. Zeng, Y. Du, K. K. Leung, and K. Huang, "Energy-Efficient Radio Resource Allocation for Federated Edge Learning," *arXiv:1907.06040*, Jul. 2019.

FL Applications to Networks (Paper #6)

- Concept





B. Liu, L. Wang, M. Liu, and C.-Z. Xu, "Federated Imitation Learning: A Privacy Considered Imitation Learning Framework for Cloud Robotic Systems with Heterogeneous Sensor Data," *arXiv:1909.00895*, Sept. 2019.

Concluding Remarks and Q&A

- More questions?
  - joongheon@gmail.com
  - joongheon@cau.ac.kr
- More details?
  - https://sites.google.com/site/joongheonkim/
  - http://prof.cau.ac.kr/~joongheon