南京郵電大學
Nanjing University of Posts and Telecommunications

Exploration of Optimizing Federated Learning on Client Selection with Reinforcement Learning

Reported by Zhouli Fan   Time：2024.04.18

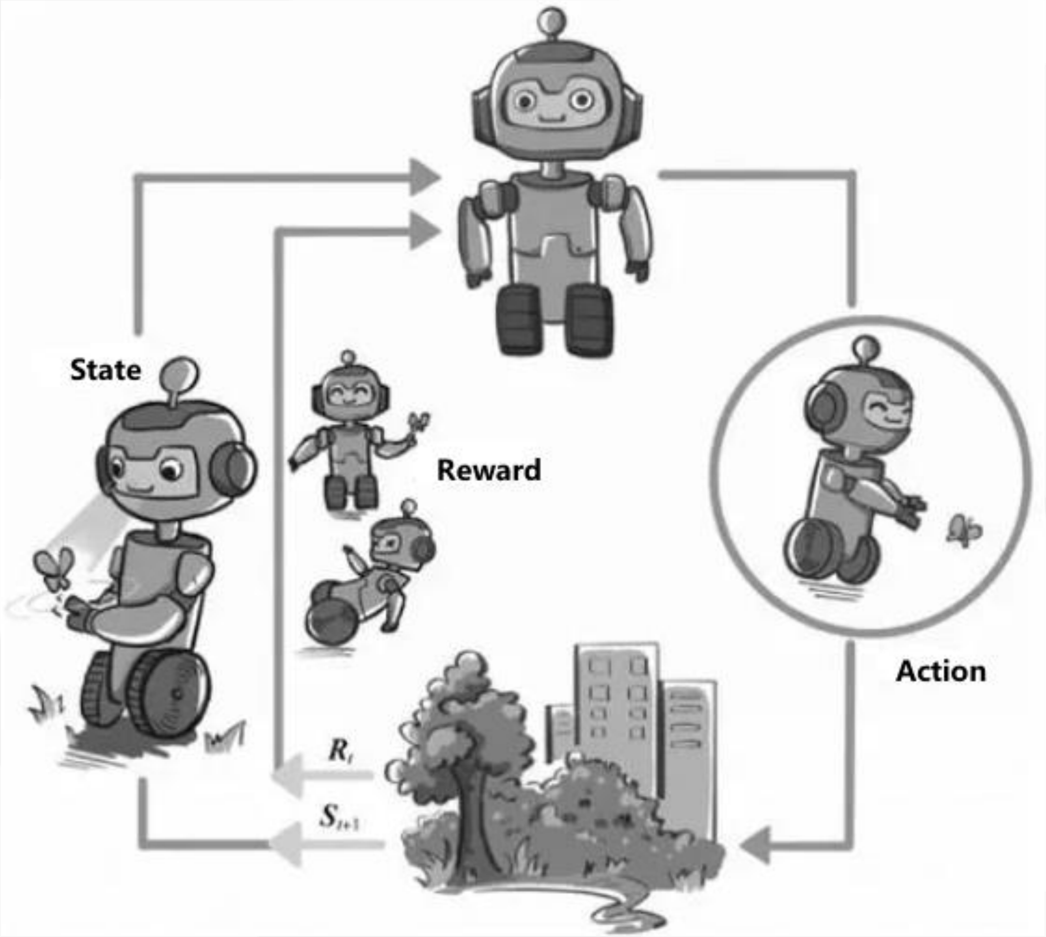
Catalogue

Reinforcement Learning

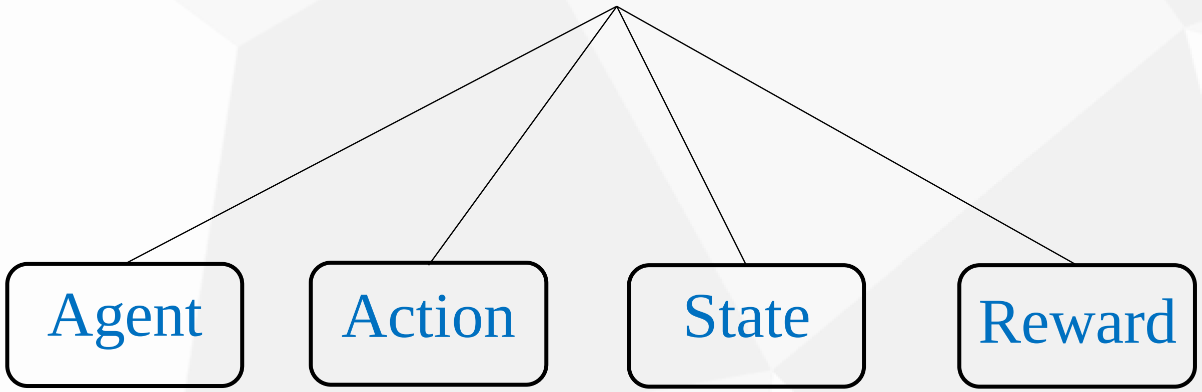
Introduction

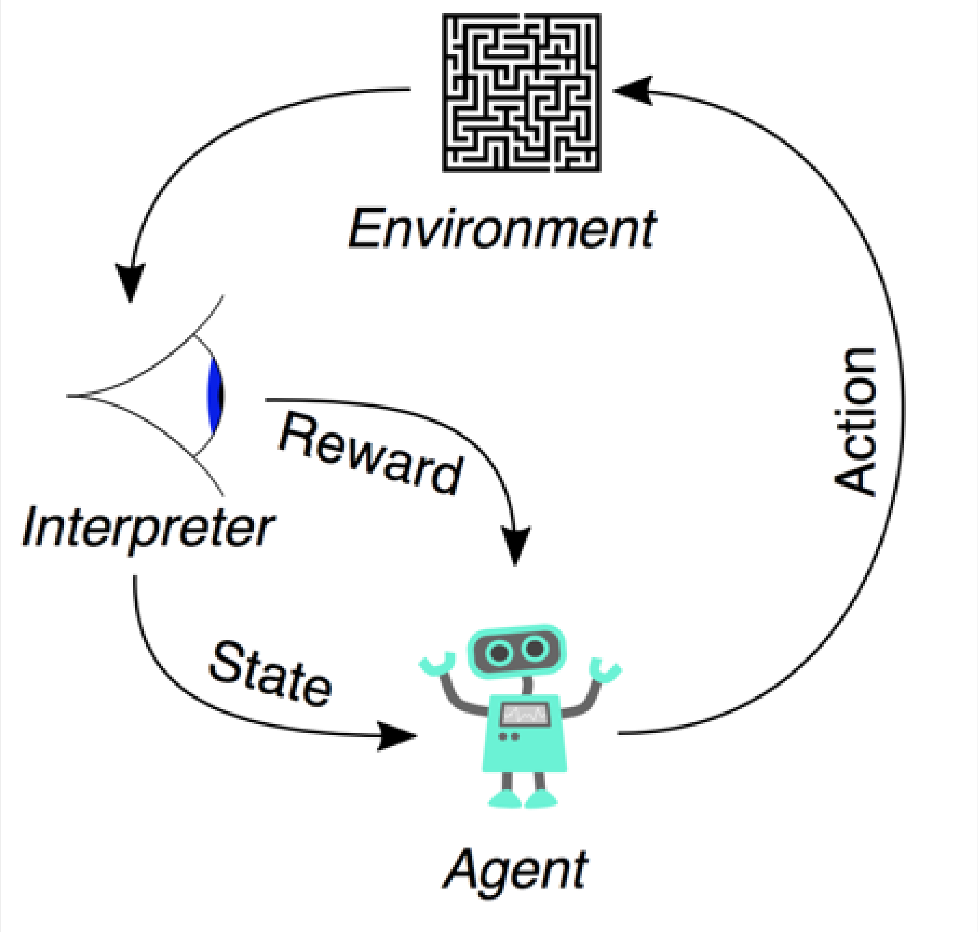
Reinforcement Learning (RL) :
a computational method in which machines achieve goals by interacting with the environment.

A round of interaction:
The machine making an action in a state of environment, and feedback the change and rewards to the next state.



Basic composition





Reinforcement Learning

Q-Learning

The key of the Q-Learning is **Q-Table**, Q-Table includes columns and rows, which contain a reward list for the best behavior of each state in a specific environment.
Q-Table can help agents understand which actions may bring rewards, which is used to update tuples (s, a, r, s’) as Q(s, a)

State

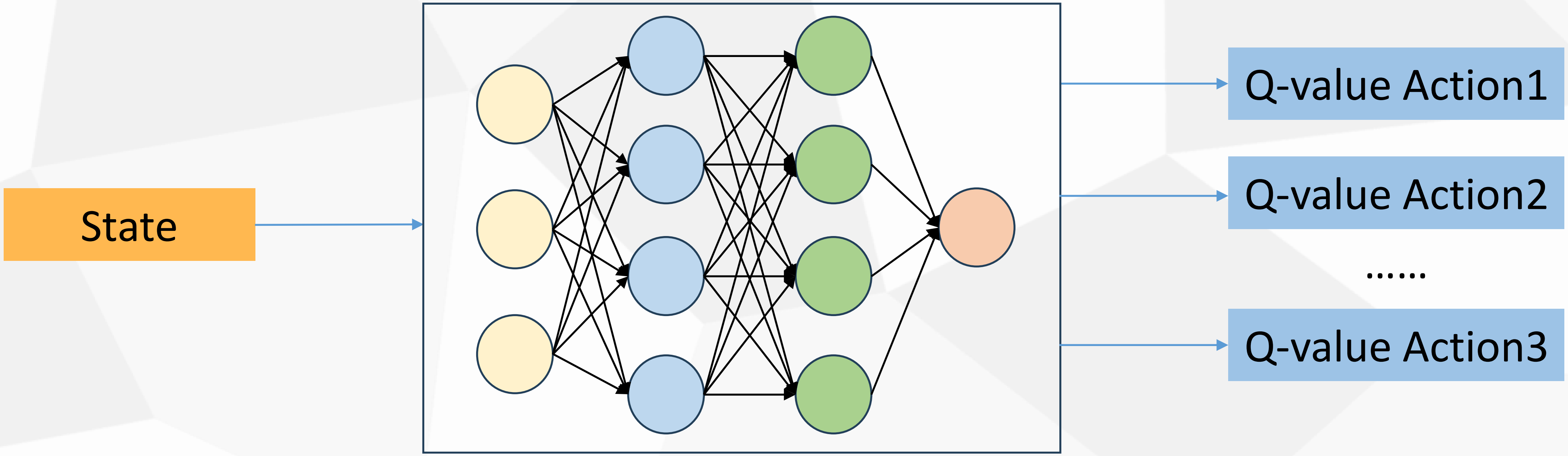
Action

Q-Table	
State-Action	**Values(Rewards)**
-	0
-	0
……	……
-	0

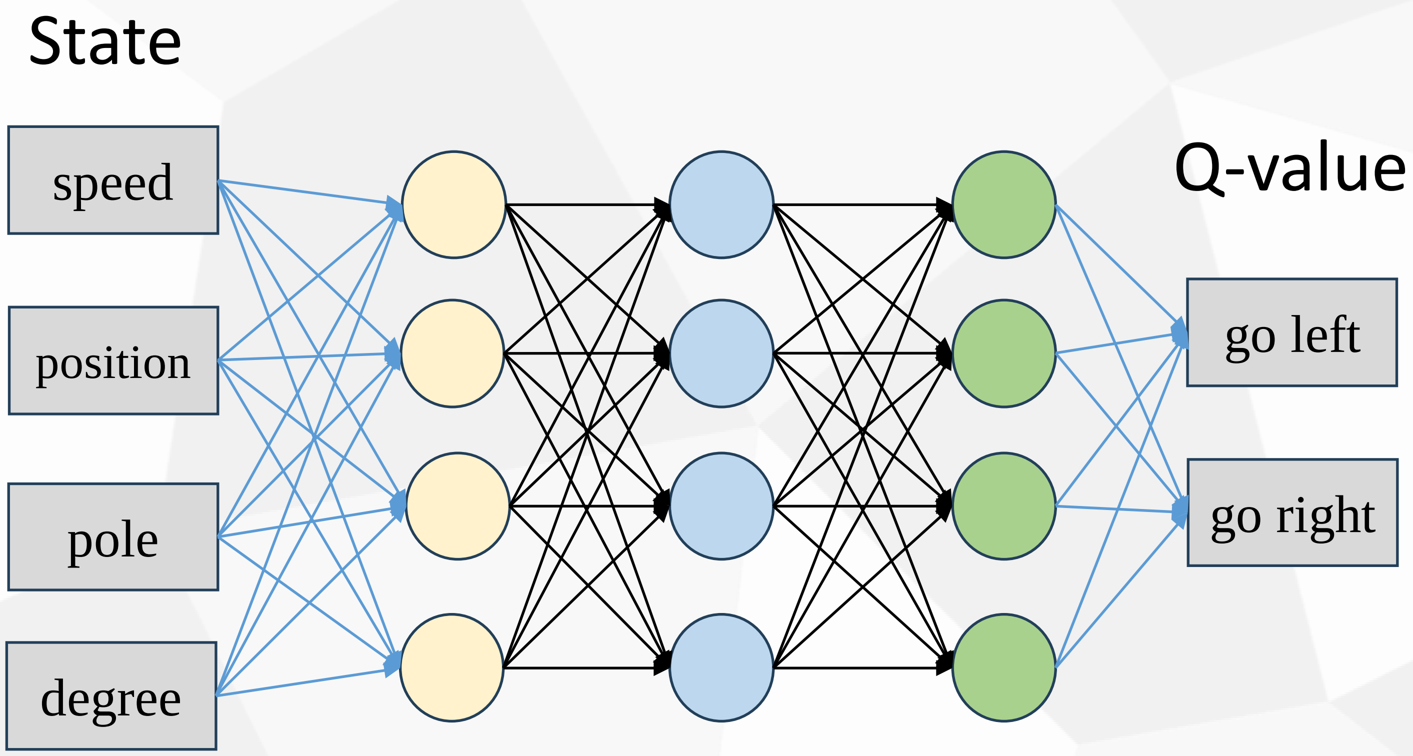
Q-value

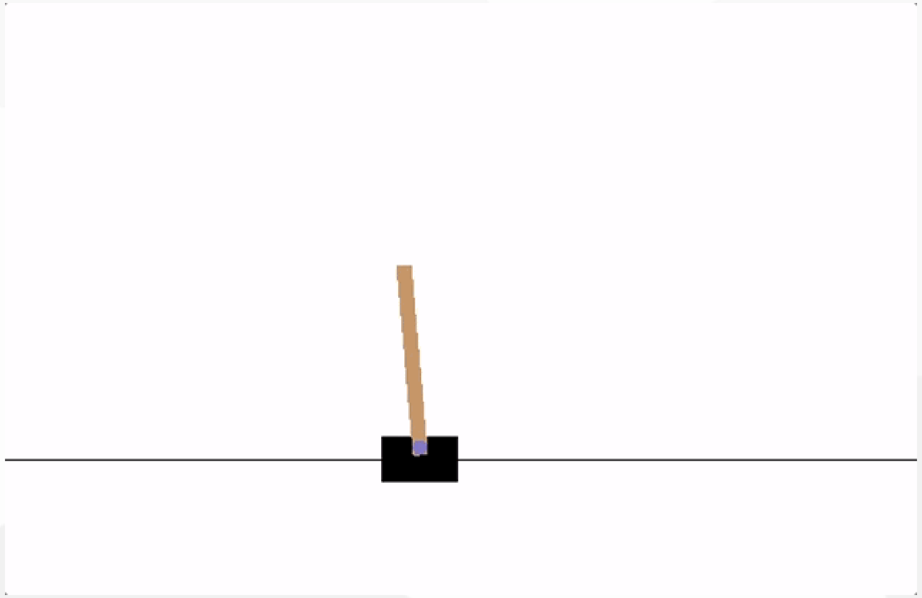
Deep Q-Learning

Deep Q-Learning(DQL) is used to solve large-scale problems that Q-Learning can not solve, using function fitting methods to estimate Q-values, which is an approximate method that uses **neural networks** to represent the action value function Q(s, a)



Deep Q-Learning example： Cartpole

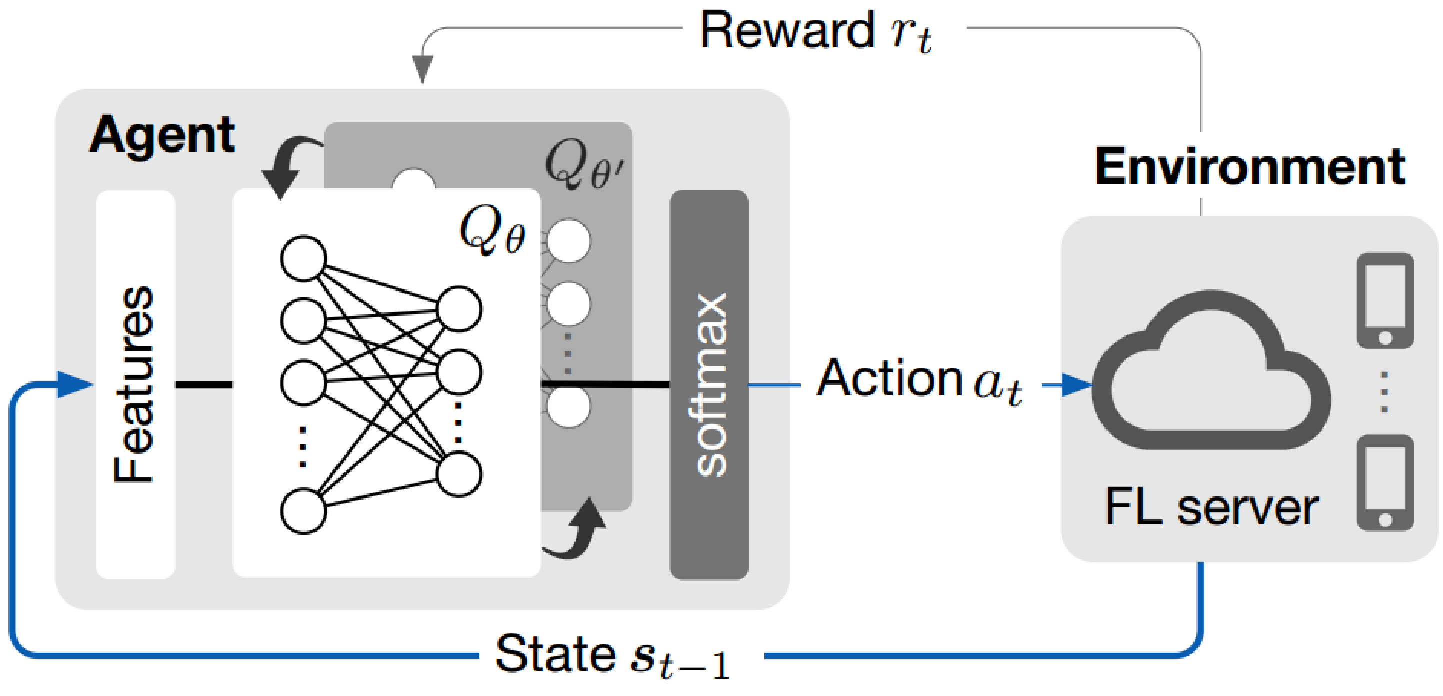




FAVOR

《Optimizing Federated Learning on Non-IID Data with Reinforcement Learning》

A experience-driven control framework that intelligently chooses the clients to participate in each round of federated learning.



DQN Agent interacting with the FL Server

DQN Design

- State

$s_t = (w_t, w_t^{(1)}, w_t^{(2)}, \ldots, w_t^{(N)})$

- Action

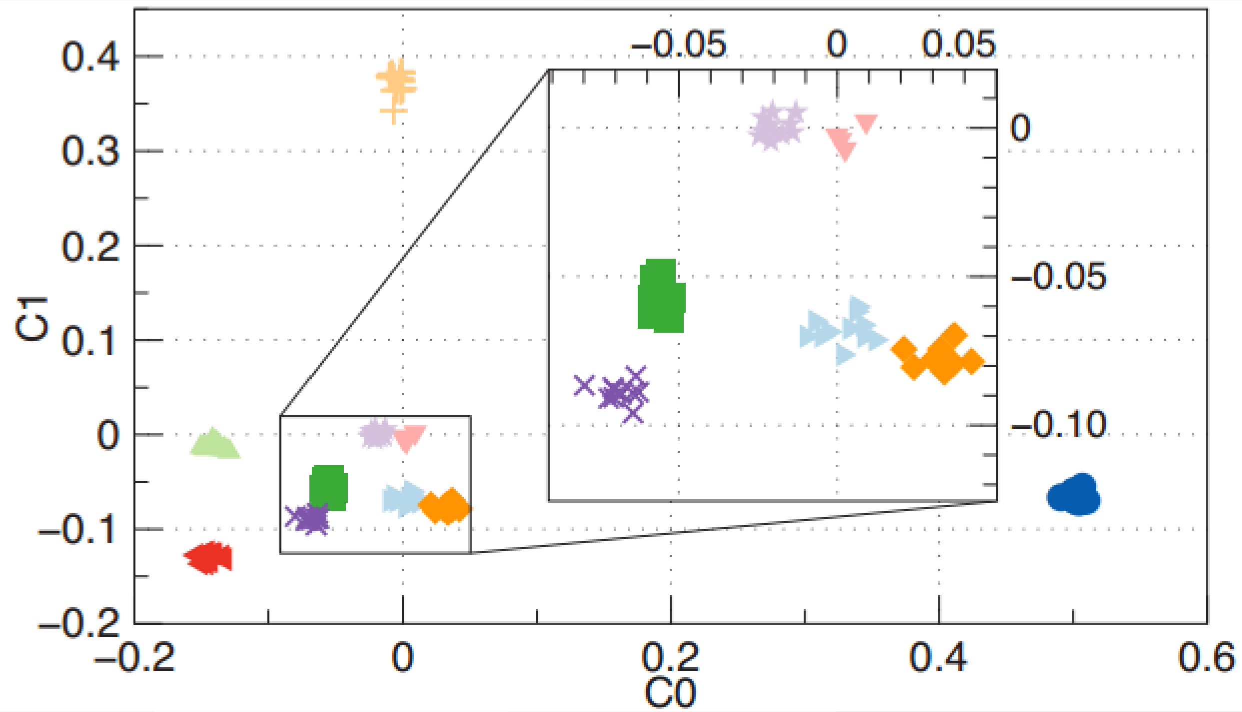
Select K clients of Top-K values of $Q^*(s_t, a)$

- Rewards

$r_t = C^{(w_t - \Omega)} - 1$

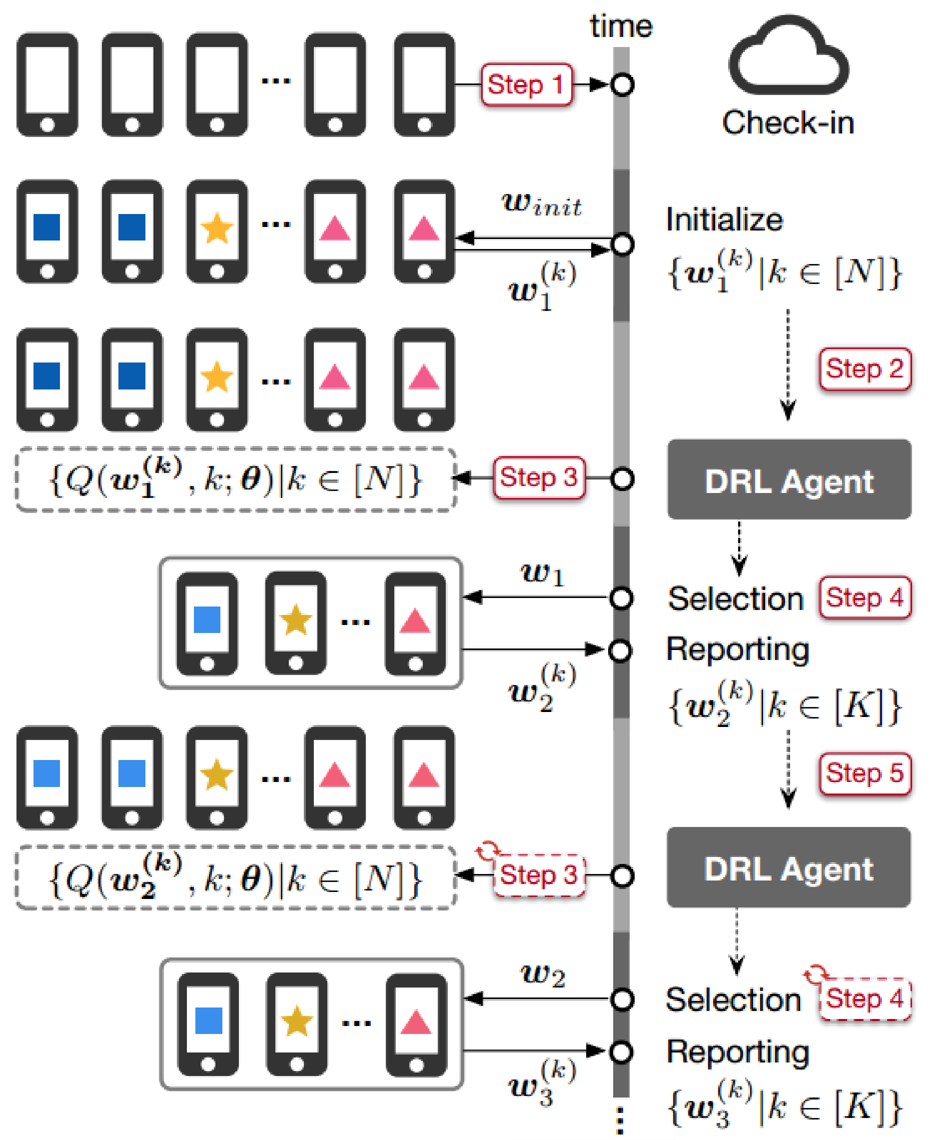
$R = \sum_{t=1}^{T} \gamma^{t-1} r_t = \sum_{t=1}^{T} \gamma^{t-1} (C^{(w_t - \Omega)} - 1)$

Principle component analysis (PCA)



PCA of CNN weights on the MNIST dataset

FAVOR



① all N clients check in with the FL Server

② each client download w_{init} and train one epoch to get { $w_t^{(k)}, k \in [N]$, t=1 }

③ DQN computes Q(s_t, a) for all clients

④ Selects K clients from top-K Q(s_t, a)

⑤ upload $\{ w_{t+1}^{(k)}, k \in [N] \}$ to the server

⑥ Move into round t+1 and repeat ③-⑥

《FLASH-RL: Federated Learning Addressing System and Static Heterogeneity using Reinforcement Learning》

FLASH-RL improves the State and rewards of DQN based on FAVOR which adds clients' CPU cores, CPU frequency, bandwidth, and size of dataset to the State, and makes the rewards more comprehensive.

Approach	Algorithm	Static Heterogeneity	System Heterogeneity	Number of Agents
FAVOR	DDQL	√	×	1
Adaptive CS in DRL	DDQL	×	√	1
Trust-Augment RL	DDQL	×	√	1
Multi-Agent RL	MARL	√	√	N

Comparison of STATE-OF-THE-ART RL client selection

Handling System Heterogeneity： Latency

- Total latency

$$l_t^k = T_t^{local_k} + T_t^{transmission_k}$$

- Local computing time

$$T_t^{local_k} = \frac{|D_{t,k}| g_k}{c_k f_{t,k}}$$

- Transmission time

$$T_t^{transmission_k} = \frac{\sigma(m_k)}{b_{t,k}}$$

Handling Static Heterogeneity: Reputation and Utility

- Reputation-based function

$$\psi_t^k = \lambda\left(A_t^k - A_{t-1}\right) + (1-\lambda)\psi_{t-1}^k$$

- Utility function

$$d(w_t^k, w_t) = \frac{1}{|w_t|}\sum_{j=1}^{|w_t|}\left|\frac{w_t^{k,j} - w_t^j}{w_t^j}\right|, \quad \zeta_t^{\mathrm{k}} = \begin{cases} e^{-\left|d(w_t^k, w_{\mathrm{t}})\right|}, & if\ P_t > P_{t-1} \\ 1 - e^{-\left|d(w_t^k, w_{\mathrm{t}})\right|}, & if\ P_t \le P_{t-1} \end{cases}$$

- Reputation-based utility function

$$\psi_t^k = \lambda\zeta_t^{\mathrm{k}} + (1-\lambda)\psi_{t-1}^k$$

Rewards by Combining System and Static Heterogeneity

- Reputation-based utility function with System and Static Heterogeneity

$$\psi_t^k = \lambda(\alpha_1 \zeta_t^{\mathrm{k}} - \alpha_1 \mathrm{l}_t^k) + (1 - \lambda)\psi_{t-1}^k$$

DQN Design

- State

$s_t = (s_t^1, s_t^2, s_t^3, \ldots, s_t^N),\ \ s_t^k = (w_t^k, n_t^k, c_k, f_{t,k}, b_{t,k})$

- Action

Select K clients of Top-K values of $Q^*(s_t,\ a)$

- Rewards

$$\psi_t^k = \lambda(\alpha_1 \zeta_t^{\mathrm{k}} - \alpha_1 \mathrm{l}_t^k) + (1-\lambda)\psi_{t-1}^k$$

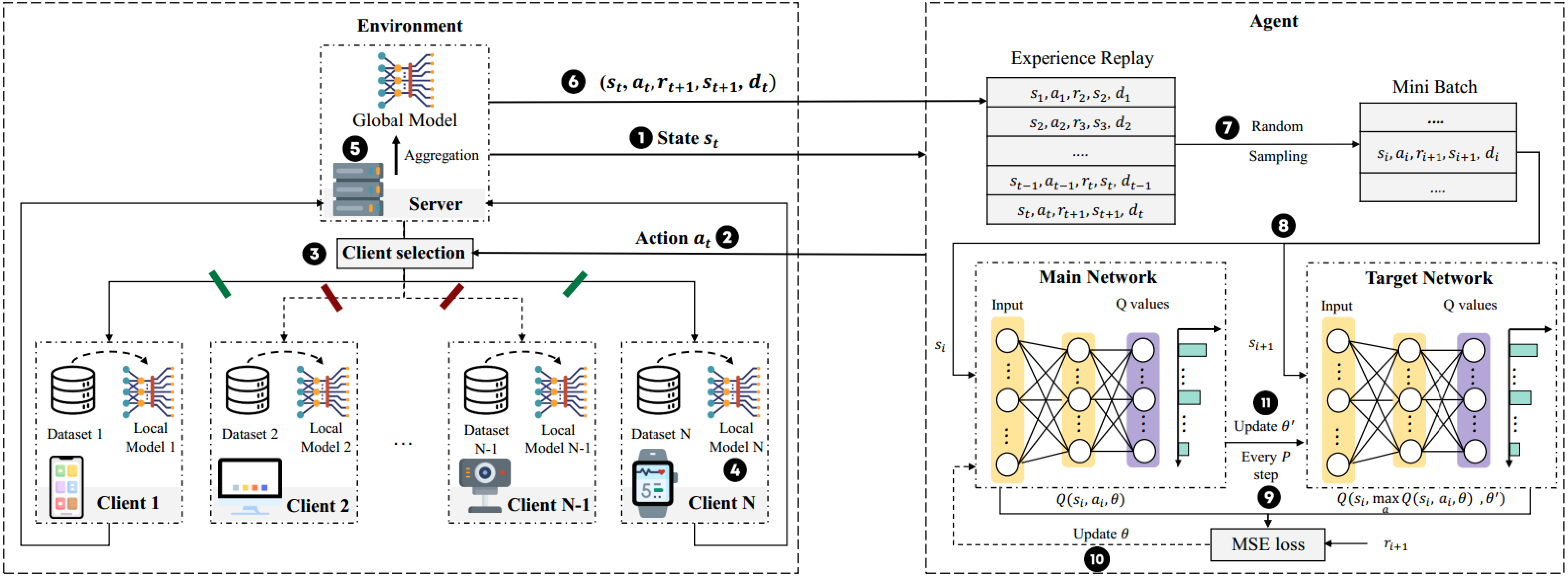
Edge Equipment Simulated Specifications

Name	CPU Frequency (MHZ)	Number of Cores
Hardware Spec. 1	921	128
Hardware Spec. 2	1300	256
Hardware Spec. 3	800	384
Hardware Spec. 4	1100	384
Hardware Spec. 5	1377	384
Hardware Spec. 6	350	4
Hardware Spec. 7	1500	4
Hardware Spec. 8	700	1
Hardware Spec. 9	3950	2
Hardware Spec. 10	4300	4
Hardware Spec. 11	4400	4
Hardware Spec. 12	4400	8

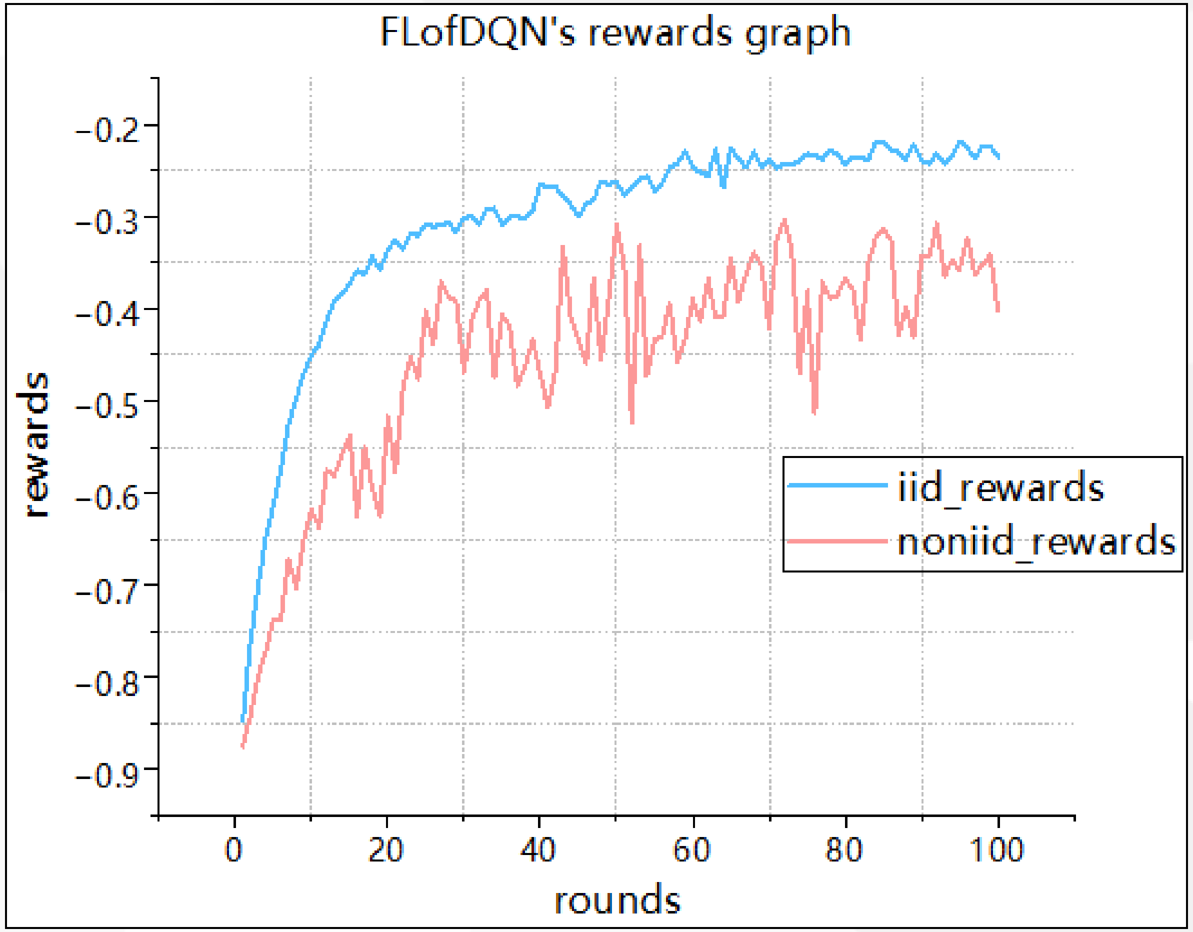
Transfer Protocols Specifications

Name	Bandwidth (Mb/s)
Wi-Fi 1	6
Wi-Fi 3	33
Wi-Fi 4	336
Fast Ethernet	100

FLASH

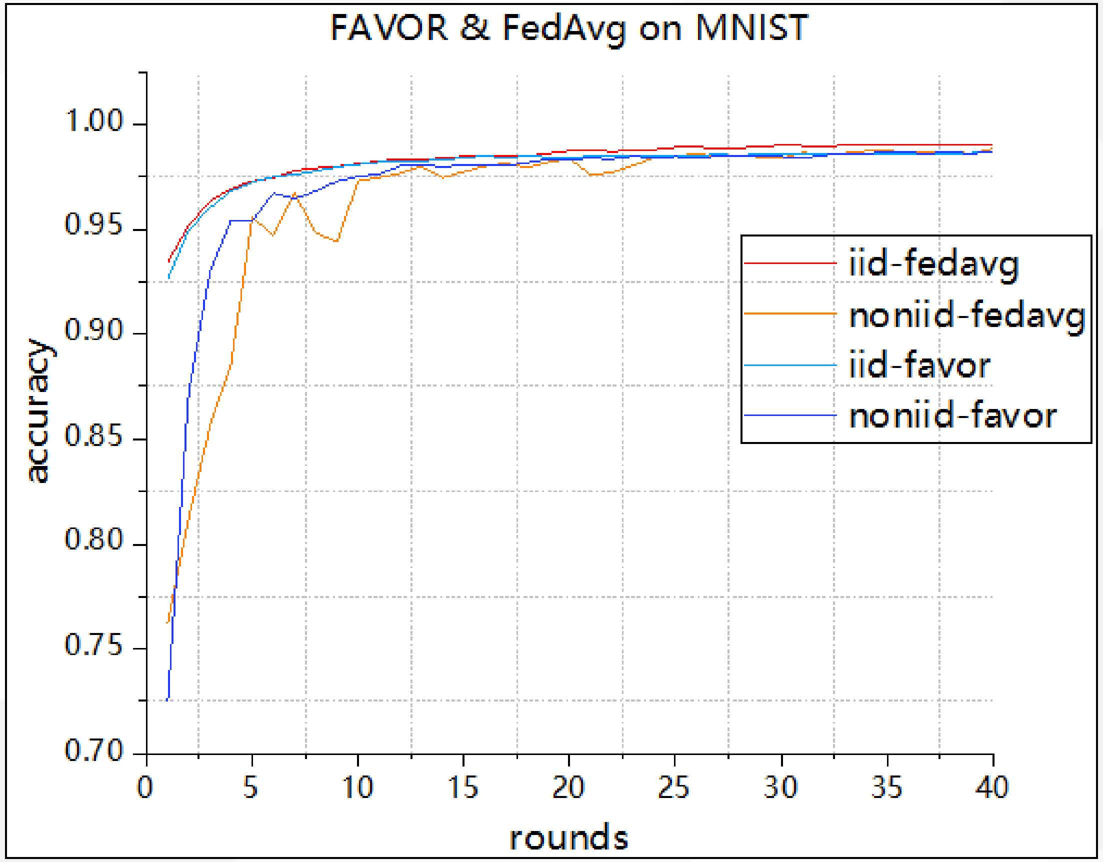

Environment
Global Model
Aggregation
Server
Client selection
Dataset 1
Local Model 1
Client 1
Dataset 2
Local Model 2
Client 2
Dataset N-1
Local Model N-1
Client N-1
Dataset N
Local Model N
Client N
$(s_t, a_t, r_{t+1}, s_{t+1}, d_t)$
State s_t
Action a_t
Agent
Experience Replay
s_1, a_1, r_2, s_2, d_1
s_2, a_2, r_3, s_3, d_2
....
$s_{t-1}, a_{t-1}, r_t, s_t, d_{t-1}$
$s_t, a_t, r_{t+1}, s_{t+1}, d_t$
Random
Sampling
Mini Batch
....
$s_i, a_i, r_{i+1}, s_{i+1}, d_i$
....
Main Network
Input
Q values
Target Network
Input
Q values
s_i
s_{i+1}
Update θ'
Every P step
$Q(s_i, a_i, \theta)$
$Q(s_i, \max_a Q(s_i, a_i, \theta), \theta')$
Update θ
MSE loss
r_{i+1}

Experiment

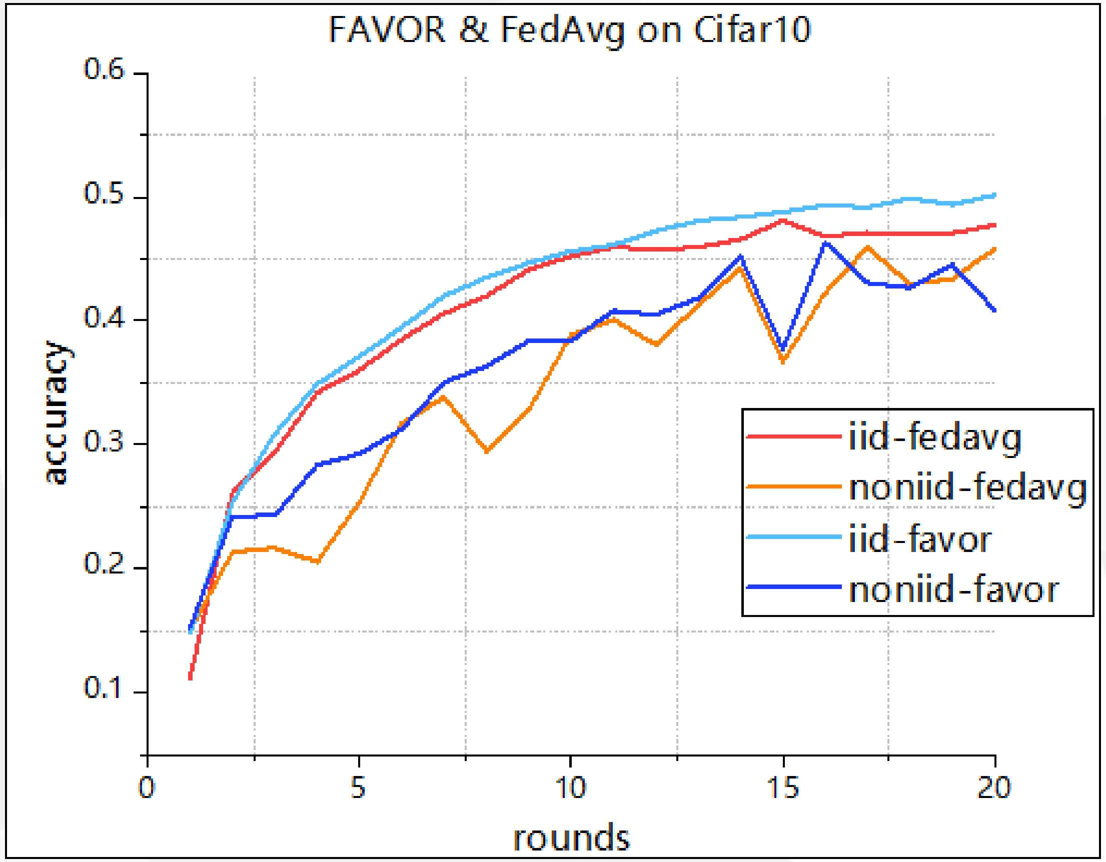


DQN’S Rewards with CIFAR10

Experiment

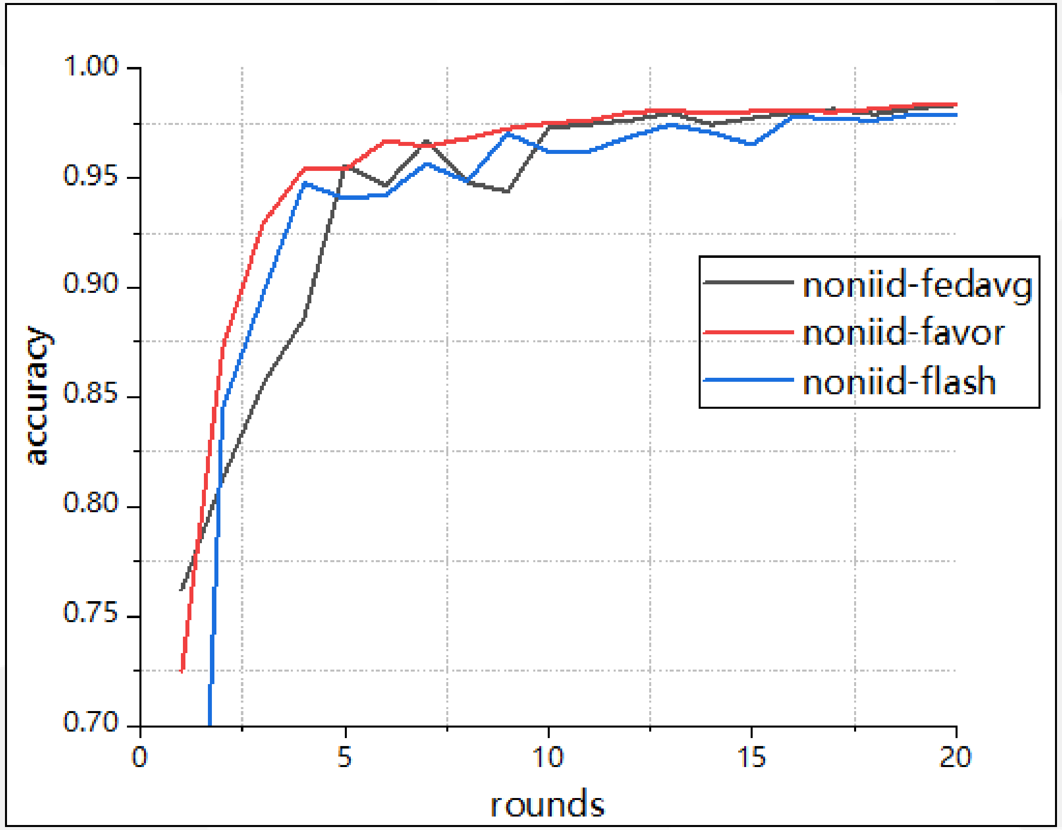


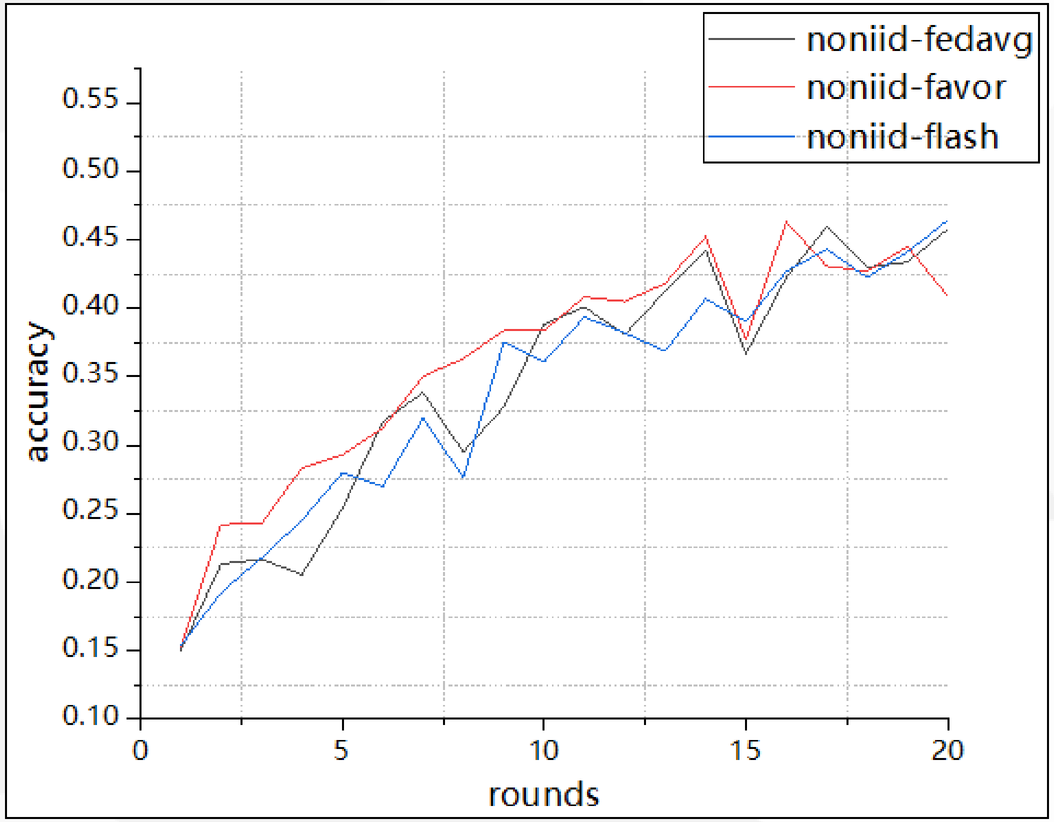
Comparative experiments on MNIST



Comparative experiments on CIFAR10

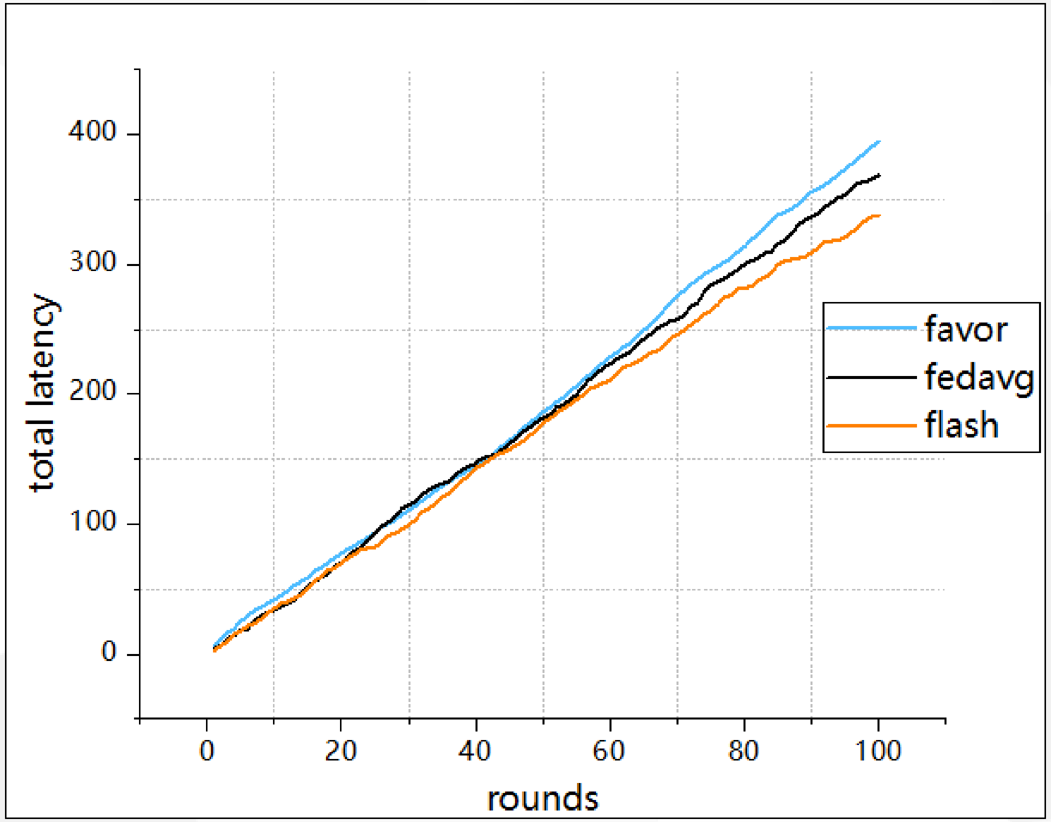
Experiment

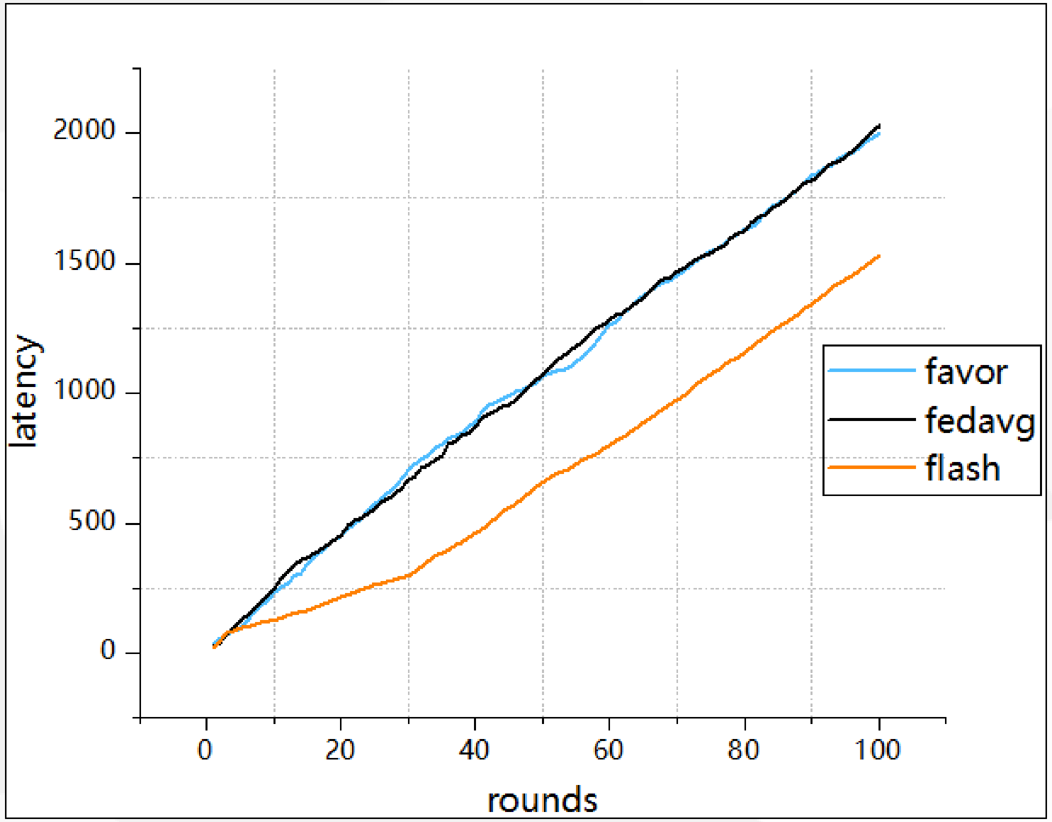




Comparative **accuracy** on MNSIT/CIFAR10 with three methods

Experiment





Comparative **Latency** on CIFAR10 /MNIST with three methods

Future Work

- Continue reading some of the methods mentioned in the FLASH-RL

- Think and consider the next innovation direction
  - Rewards function, action selection, RL strategies
  - Environmental innovation

南京邮電大學
Nanjing University of Posts and Telecommunications

Thanks