Inter-cell Interference Coordination for Small Cell Wireless Communications

Liu Yuchen

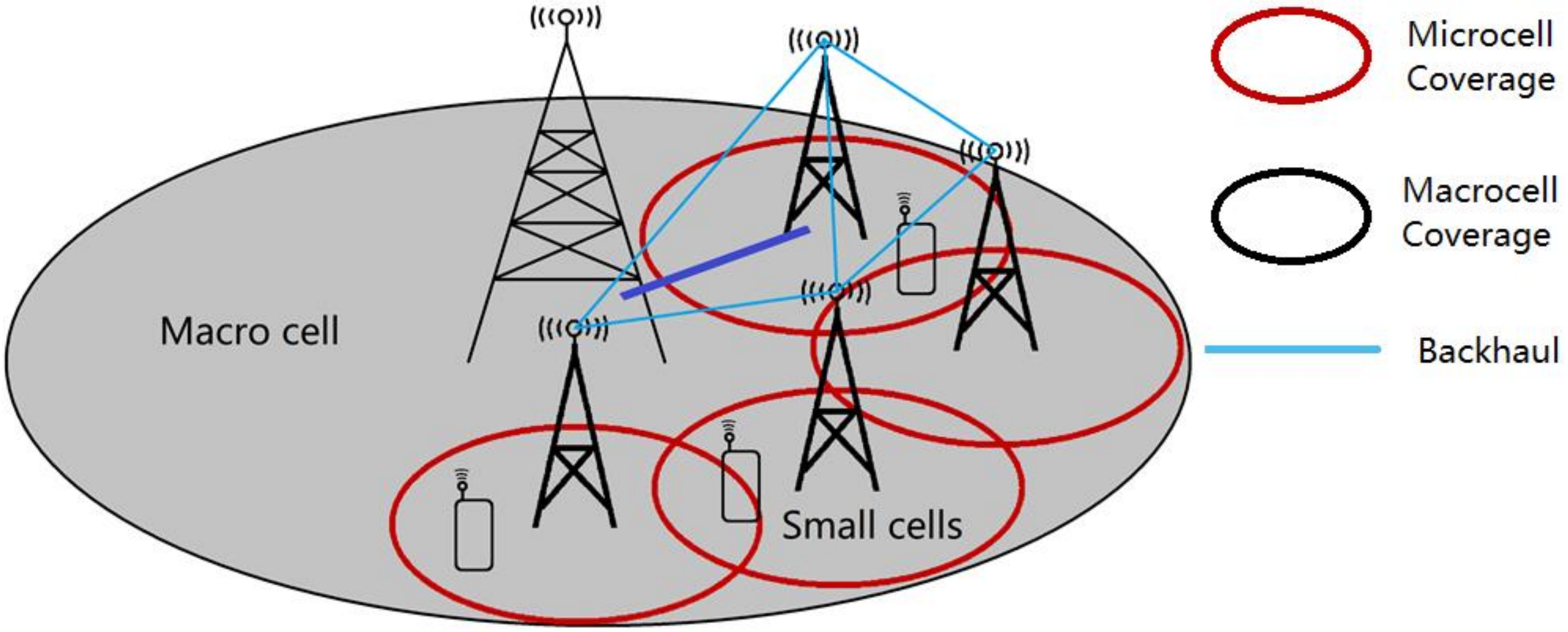
Fukawa Laboratory

Tokyo Institute of Technology

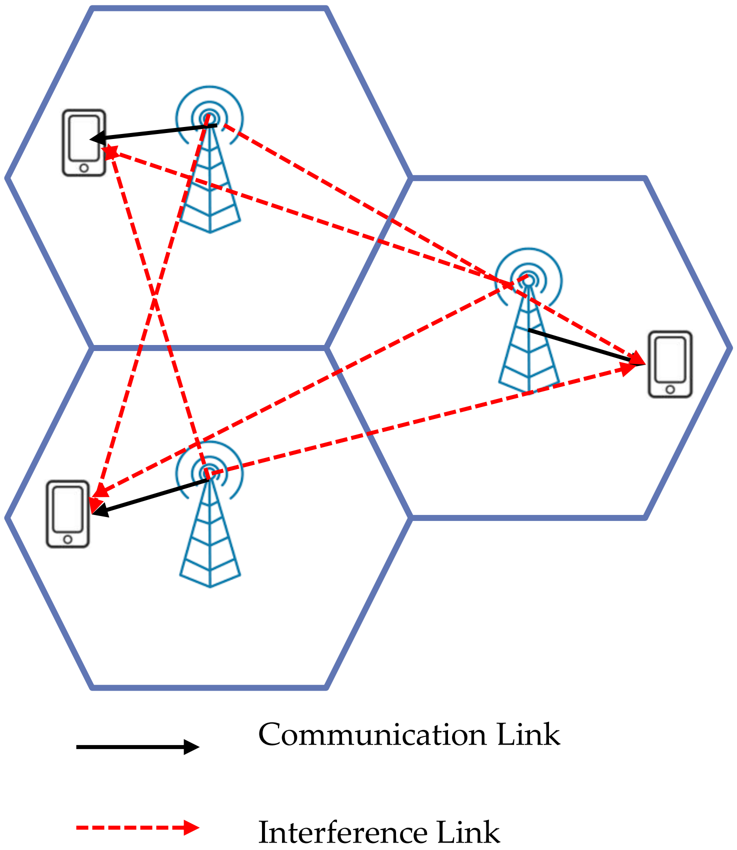
Contents

1. Research Background

- Increasing of carrier frequency in mobile communication system decrease size of cell, which leads to dense deployment of base station in urban area.
- High demand for data traffic also needs dense deployment of base station.
- Interference control is essential to small cell wireless communication system.



Inter-Cell Interference Coordination (ICIC)



- ICIC controls transmit power and beamforming to suppress Inter-Cell Interference (ICI)
- Appropriate transmit power increase system capacity, meanwhile reduce interference to other User Terminals (UT)
- Precoding vector provide high directive beamform which alleviates interference to other UTs

2. Signal Model and Problem Formulation

Channel

Angle of Arrival

$$\mathbf{H}_{i,j} = \frac{1}{\sqrt{\Lambda_{i,j}}} \sum_{l=1}^{L} h_{i,j,l}\mathbf{a}(\theta^{r}_{i,j,l})\mathbf{a}^{\mathrm{H}}(\theta^{t}_{i,j,l}) \qquad \Lambda_{i,j} = d^{\alpha}_{i,j}\eta_{i,j}$$

Angle of Departure

Transmitted signal

$$\mathbb{E}[s_k^{\mathrm{H}} s_k] = 1$$

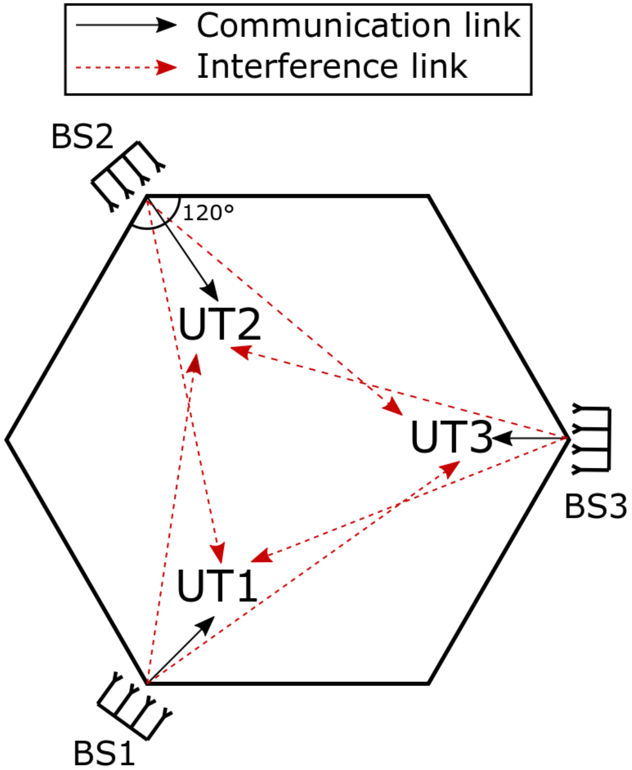
Received signal

$$\mathbf{r}_i = \mathbf{H}_{i,i}\mathbf{f}_i\sqrt{p_i}s_i + \sum_{j\neq i}^{K} \mathbf{H}_{i,j}\mathbf{f}_j\sqrt{p_j}s_j + \mathbf{n}_i$$

Equivalent channel

$$\mathbf{g}_{i,j} = \mathbf{H}_{i,j}\mathbf{f}_j \qquad \mathbb{E}[\mathbf{n}_i\mathbf{n}_i^{\mathrm{H}}] = \sigma_n^2\mathbf{I}_N$$

$$\hat{s}_i = \frac{\sqrt{p_i}\mathbf{g}_{i,i}^{\mathrm{H}}}{||\mathbf{g}_{i,i}||^2}\left\{\sqrt{p_i}\mathbf{g}_{i,i}s_i + \sum_{j=1,j\neq i}^{K}\sqrt{p_j}\mathbf{g}_{i,j}s_j + \mathbf{n}_i\right\}$$



$\mathbf{H}_{i,j}$: Channel between i-th UT and j-th BS
N: #antennas K: #links L: #path

- Signal to Interference and Noise power Ratio:

$$\mathrm{SINR}_i = \frac{p_i ||\mathbf{g}_{i,i}||^4}{\sigma_n^2 ||\mathbf{g}_{i,i}||^2 + \sum_{j=1, j\neq i}^{K} p_j |\mathbf{g}_{i,i}^{\mathrm{H}} \mathbf{g}_{i,j}|^2}$$

- Optimization problem: maximize the average capacity

$$C = \frac{1}{K} \sum_{i=1}^{K} \log_2(1 + \mathrm{SINR}_i)$$

$$\mathcal{P}: \quad \max \ \frac{1}{K} \sum_{i=1}^{K} \log_2(1 + \mathrm{SINR}_i)$$

$$s.t. \ \mathbf{f}_k \in \mathbb{F}, \forall k \in [1, K]$$

Predefined codebook for beamforming vector

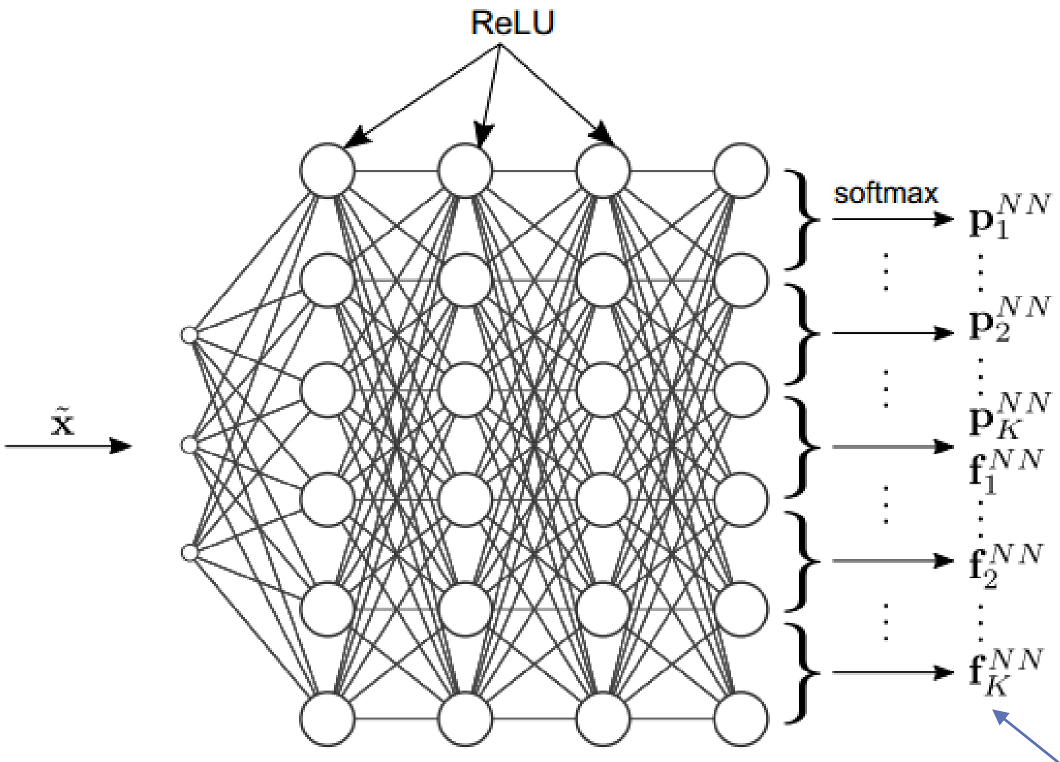
$$p_k \in \mathbb{P}, \forall k \in [1, K]$$

Predefined codebook for transmit power

3. Research on Topic
3.1 Traditional Approach

- **Greedy Search Algorithm (ES)**
  Explores every possible combination of all BS's transmit power and beamforming vector.
  Full CSI is necessary for the capacity computation where the CSI feed- back is conducted over the backhaul of unlimited capacity.
  Guaranteed to get the optimal solution and almost infeasible in most application.
- **Belief Propagation Algorithm[1]**
  Employing the BP power control scheme, the probabilities of P1, . . . , PK (possible of power level setting) are computed by iteratively exchanging messages between connected BSs and UTs.
  BP algorithm guarantees an approximate solution after number of iterations.
- **Maximum Power Algorithm**
  The maximum power transmission is considered to yield the lower bound performance of power control ICIC. In the algorithm, all BSs transmit the highest power level, P1 = . . . = PK = Pmax
- **Distributed Pricing Algorithm[2]**
  An interference payment must be paid by each link when updating transmit power. The transmit power level is updated to maximize a utility function.
  Price exchange protocol are needed in the DP algorithm.

3.2 Supervised Learning (LS)-based Approach



beamforming vector for *K*-th BS in one-hot representation

- Input data: Power of all the possible equivalent channels

$$\mathbf{x}_{i,j} = \left[||\mathbf{g}_{i,j}(\mathbb{F}_1)||^2, \ldots, ||\mathbf{g}_{i,j}(\mathbb{F}_{|\mathbb{F}|})||^2 \right]$$

$$\mathbf{x} = [\mathbf{x}_{1,1}, \ldots, \mathbf{x}_{1,K}, \ldots, \mathbf{x}_{K,K}]^{\mathrm{T}}$$

$$\tilde{\mathbf{x}} = \log_{10}\left(\frac{\mathbf{x}}{x_{max}}\right)$$

The greatest element in **x**

- Output data:
  one-hot vector indexing transmit power or beamforming vector

- Enhanced input data:
  Including the correlation terms improves the performance

$$\mathrm{SINR}_i = \frac{p_i ||\mathbf{g}_{i,i}||^4}{\sigma_n^2 ||\mathbf{g}_{i,i}||^2 + \sum_{j=1, j\neq i}^{K} p_j \underline{|\mathbf{g}_{i,i}^{\mathrm{H}} \mathbf{g}_{i,j}|^2}}$$

correlation terms

Weight Update of SL-based ICIC

- Regularization
  - L_2 regularization (minimize $\frac{\beta_{reg}}{2}|\mathbf{w}|^2$) NN weights
  - Dropout (randomly set outputs of nodes to 0 with possibility of *droprate*)

- Loss function

$$J_k^{\text{pow}} = -\frac{1}{|\mathbb{P}|}\sum_{r=1}^{|\mathbb{P}|}\left(p_{k,r}^{*}\log p_{k,r}^{NN}\right)$$

$$J_k^{\text{beam}} = -\frac{1}{|\mathbb{F}|}\sum_{t=1}^{|\mathbb{F}|}\left(f_{k,t}^{*}\log f_{k,t}^{NN}\right)$$

$$J = \frac{1}{K}\sum_{k=1}^{K} J_k^{\text{pow}} + \frac{1}{K}\sum_{k=1}^{K} J_k^{\text{beam}} + \frac{\beta_{reg}}{2}||\mathbf{w}||^2$$

$\mathbf{p}_k^*$ and $\mathbf{f}_k^*$: solutions from ES converted to one-hot vectors

$p_{k,r}^*$: the r-th element of $\mathbf{p}_k^*$
$f_{k,t}^*$: the t-th element of $\mathbf{f}_k^*$
$p_{k,t}^{NN}$: the r-th element of $\mathbf{p}_k^{NN}$
$f_{k,t}^{NN}$: the t-th element of $\mathbf{f}_k^{NN}$

- Training: steepest descent & back propagation

$$\mathbf{w} = \mathbf{w} - \alpha_l \frac{\partial J}{\partial \mathbf{w}}$$

learning rate

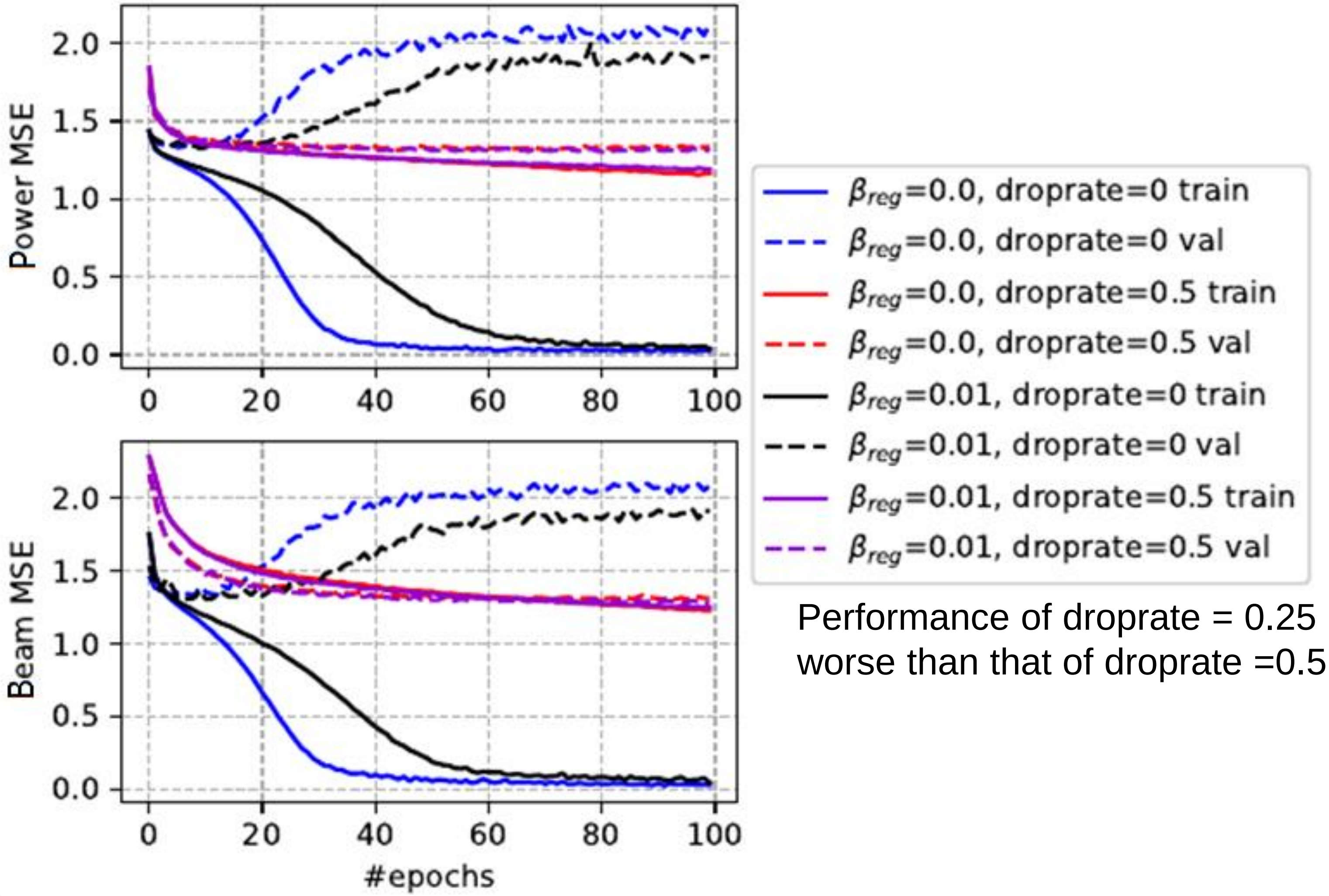
Needs Training data generated by ES !!!

Simulation Conditions (SL)

- Radio Network-	
Diameter of service area	30m
Path loss exponent (α)	4
Shadowing std. dev.	8 dB
K factor of Rician channel	10 dB
◇ Number of BS antenna (N)	4 (ULAs), 16 (URAs)
Number of UT antenna (M)	4 (ULAs)
Noise power	-100 dBm
◇ Number of beamforming patterns ($\lvert\mathbb{F}\rvert$)	4
◇ Transmitting power levels ($\mathbb{P}$)	$\{-10, -5, 0, 5, 10\}$ dBm
- Neural Network-	
◇ Initial learning rate	1×10^{-4}
Training set size	16000
Validation set size	1600
Minibatch size	16
◇ Number of hidden layer	4
◇ Number of nodes in each hidden layer	1024
Learning rate decay	0.96
◇ Dropout rate	0.5
◇ β_{reg}	1×10^{-2}

Simulation Results (SL) – Training Process

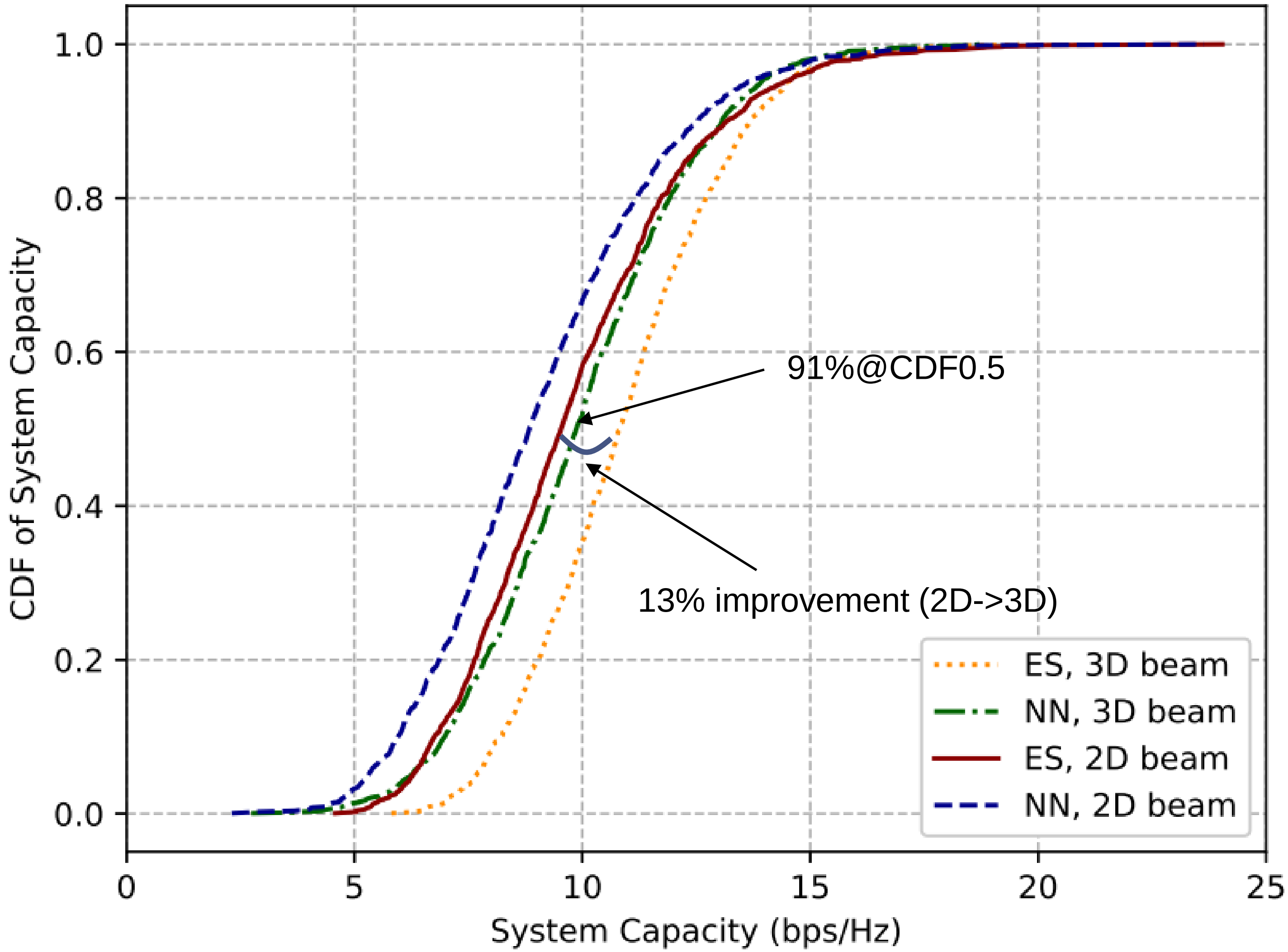
- Dropout has a stronger regularization effect and can suppress overfitting
- L_2 regularization slightly improves the performance
- Drop rate = 0.5 and β_{reg} = 0.01 are applied in following simulations



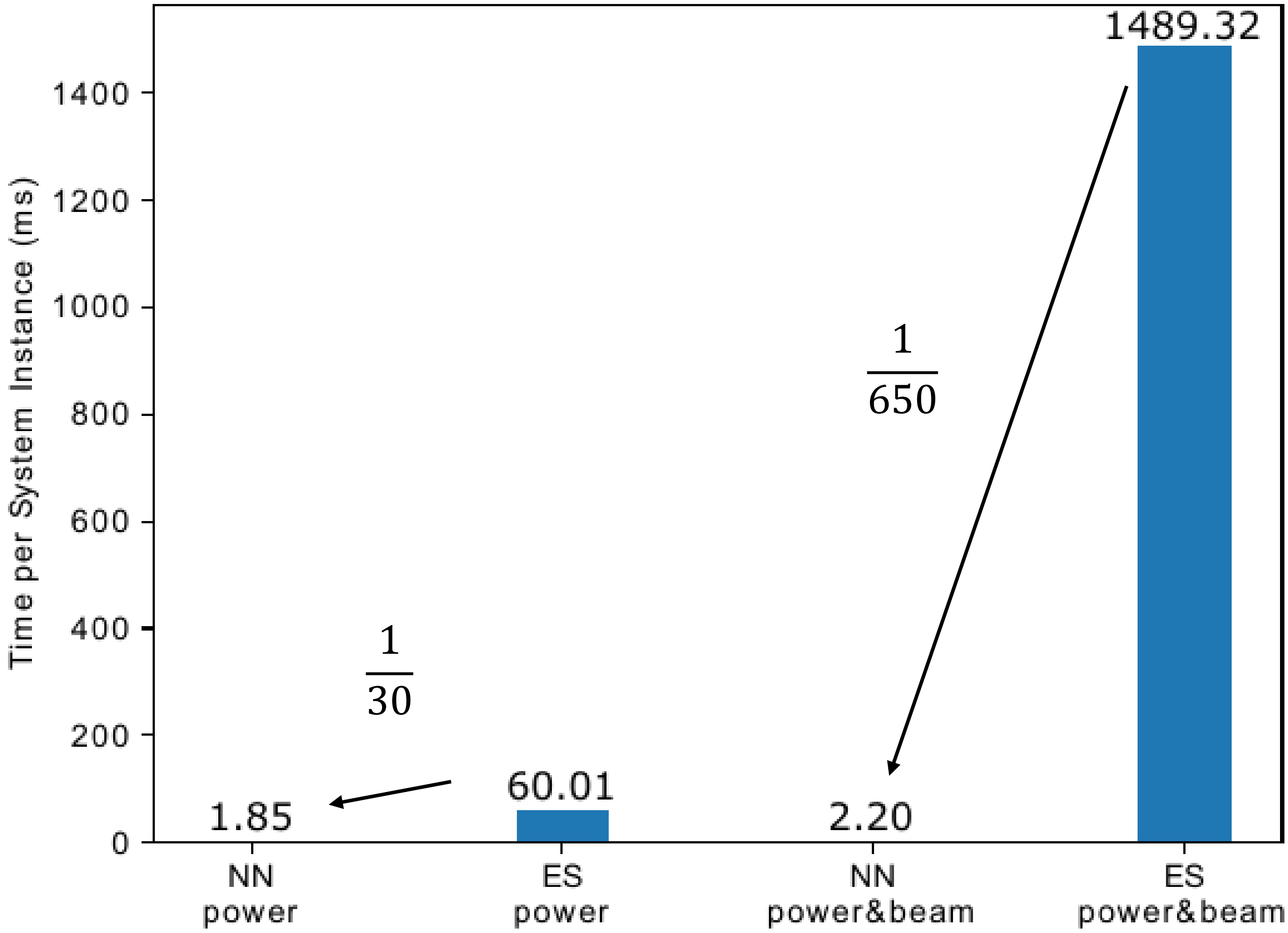
Performance of droprate = 0.25 is worse than that of droprate =0.5

Simulation Results (SL) – System Capacity (3D)

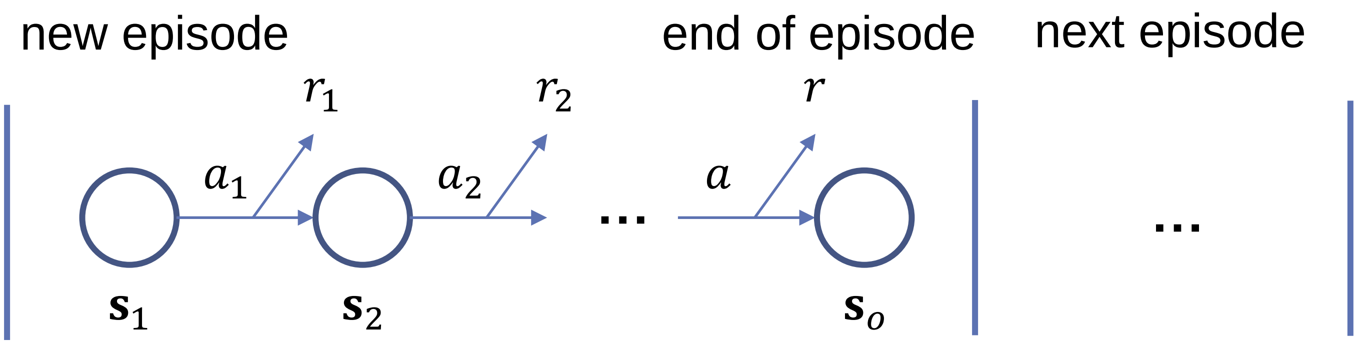
- We apply 4-by-4 URA with 8 predefined beamforming patterns, and the tilt angle of the URA 30°



Simulation Results (SL) – Running Time



3.3 Deep Q-Learning (DQL) Approach



$\mathbf{s}$: state
a : action
r : reward

- Policy: how the agent selects action given a state
- Q function $Q_\pi(\mathbf{s}, a)$: the expected cumulated reward that the agent obtains starting from state $\mathbf{s}$, taking action a and following policy π thereafter

Optimal Q function

- The RL problem is solved if we can obtain $Q^*_\pi(\mathbf{s}, a)$

$$a^* = \arg\max_a Q^*_\pi(\mathbf{s}, a)$$

Optimal action

- DQL trains Q-network $Q_\pi(\mathbf{s}, a; \mathbf{w}_z)$ (a NN) to approximate $Q^*_\pi(\mathbf{s}, a)$

NN weights

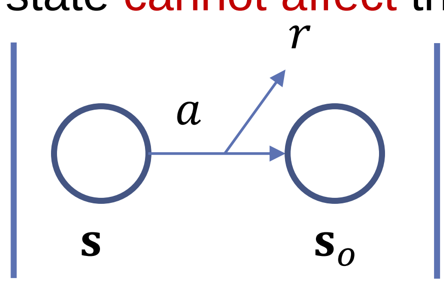
Form ICIC into RL Problem

- The state $\mathbf{s}$: power of all the possible equivalent channels

$$\mathbf{x}_{i,j} = \left[||\mathbf{g}_{i,j}(\mathbb{F}_1)||^2, \ldots, ||\mathbf{g}_{i,j}(\mathbb{F}_{|\mathbb{F}|})||^2 \right]^{\mathrm{T}}$$
$$\mathbf{s} = \left[\mathbf{x}_{1,1}^{\mathrm{T}}, \ldots, \mathbf{x}_{1,K}^{\mathrm{T}}, \ldots, \mathbf{x}_{K,K}^{\mathrm{T}} \right]^{\mathrm{T}}$$

- The action set $\mathcal{A}$: all combinations of transmit power levels in $\mathbb{P}$ and beamforming vectors in $\mathbb{F}$
- The reward r : average system capacity

- Any actions in the current state cannot affect the reward in the future states



$$Q_\pi(\mathbf{s}_\mathrm{o}; \mathbf{w}_z) = 0$$

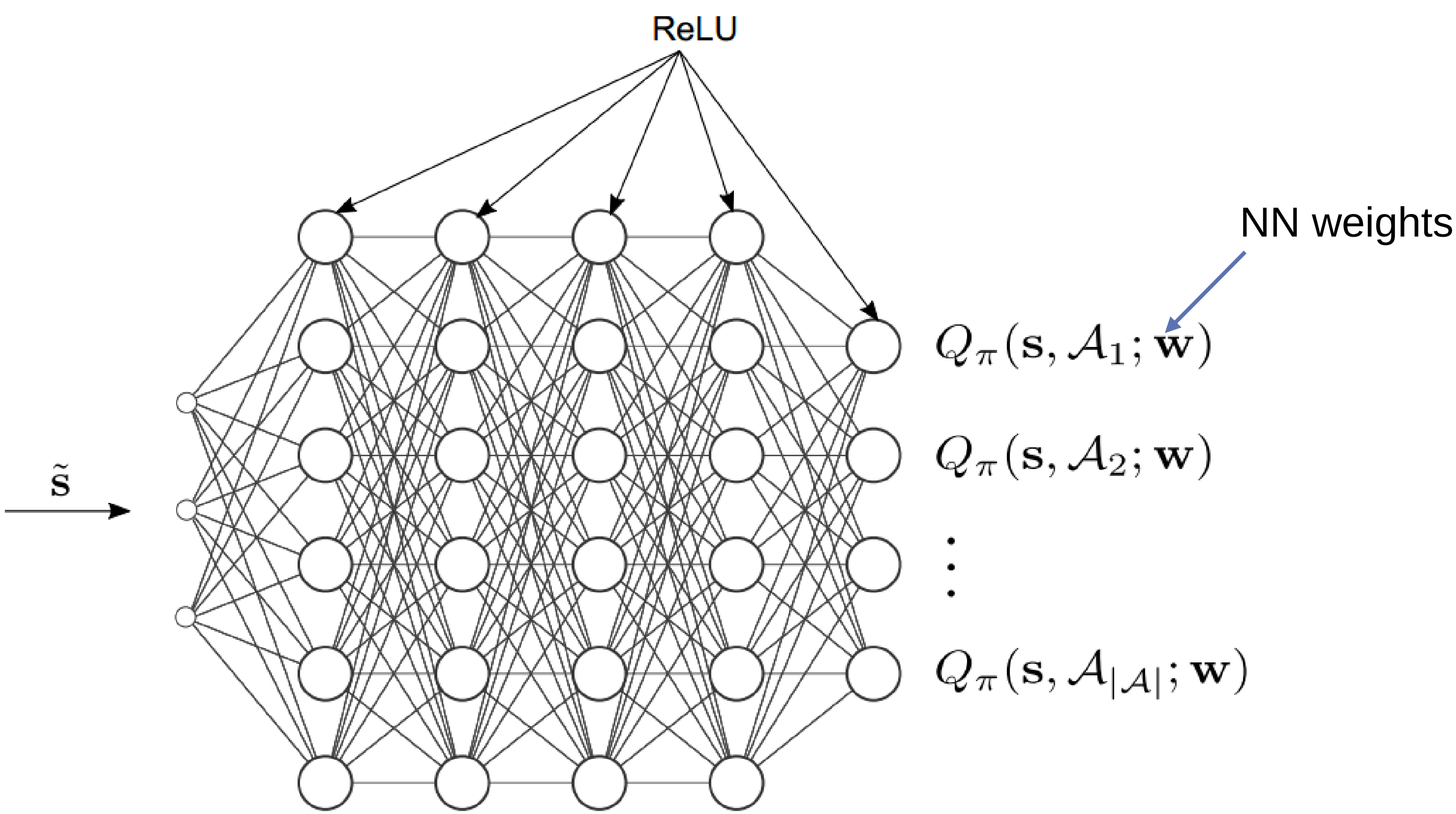
Q-network

- Input:

$$\tilde{\mathbf{s}} = \log_{10}\left(\frac{\mathbf{s}}{s_{\max}}\right)$$

The greatest element in **s**

- Output:
expected system capacities of all combinations of power & beam



Training Q-network in DQL

- Loss function of the Q-network is set to:

$$L = \mathbb{E}_{\mathbf{s},a}\left[\left(y - Q_{\pi}\left(\mathbf{s}, a; \mathbf{w}_z\right)\right)^2\right]$$

- y represents the target (expected reward)

$$y = \mathbb{E}\left[r|\mathbf{s}, a\right]$$

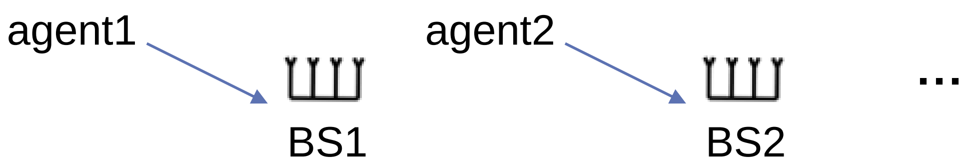
- With the steepest descent algorithm, one training step of the Q-network is given by

$$\mathbf{w}_{z+1} = \mathbf{w}_z - \underset{\substack{\uparrow \\ \text{learning rate}}}{\alpha_l} \frac{\partial\left[y - Q_{\pi}(\mathbf{s}, a; \mathbf{w}_z)\right]^2}{\partial \mathbf{w}_z}$$

Multi-Agent Setting and IDQL

- BSs in distributed manner benefits scalability

- Multi-agent setting
  - Each BS is controlled by a different agent
  - All agents operate independently and simultaneously



- Independent DQL (IDQL)
  - employ independent DQL agents with independent Q-network
  - each Q-network depends only on its own state and action

Policy, Exploration and Exploitation

- Exploration: agent tries different actions
- Exploitation: agent takes the best action (according to its Q-network)
- Policy: agent finds balance between exploration & exploitation

- The ε-greedy policy

exploration rate $\in$ (0, 1)

$$\pi(\mathbf{s}) = \begin{cases} \text{random action } a \in \mathcal{A}, & \text{with probability} \quad \varepsilon \\ \arg\max_a Q(\mathbf{s}, a), & \text{with probability} \quad 1 - \varepsilon \end{cases}$$

- Decreasing ε-greedy : start with a high ε and decrease ε along iterations

Training Process of IDQL-based ICIC

Stage 1:
Collect $\{\tilde{\mathbf{s}}, a_j, y\}$ using the current Q-network and policy

Stage 2:
Train Q-networks with $\{\tilde{\mathbf{s}}, a_j, y\}$

Stage 3:
Evaluate the Q-networks (ε = 0)

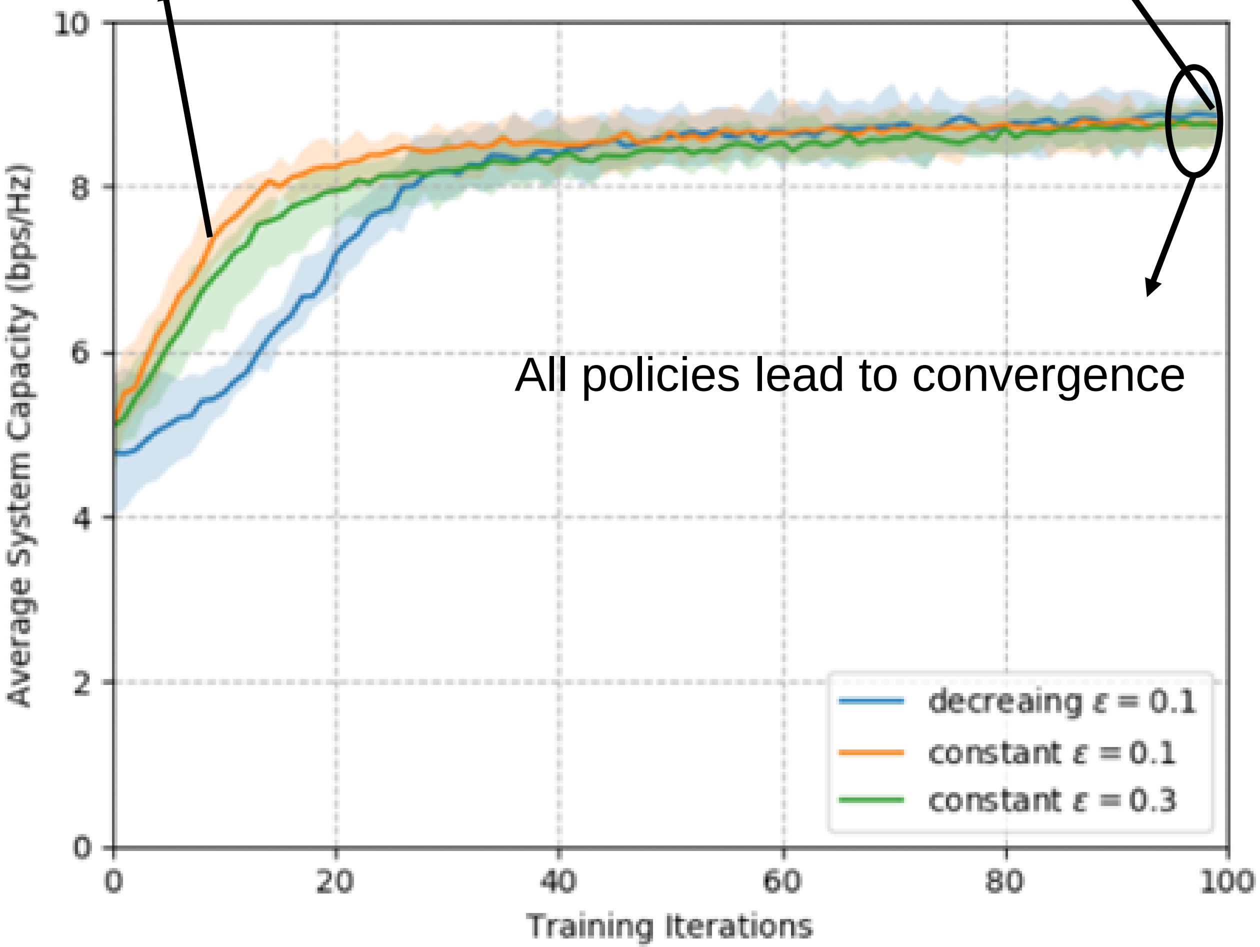
Algorithm IDQL-based ICIC

Input: Number of iterations I, number of episodes E, number of BSs K, and ε
Output: solution of (8) by DQL agents
1: **Initialize** weight vector $\mathbf{w}$ of Q-networks randomly
2: **for** (iteration = 1, iteration $\leq I$, iteration++){
    //Stage 1: collecting data
3:     Clear memory buffers of all agents
4:     **for** (episode = 1, episode $\leq E$, episode++){
5:         Generate channel instance $\{\mathbf{H}_{i,j}\}$
6:         Obtain state $\mathbf{s}$
7:         Normalize $\mathbf{s}$ into $\tilde{\mathbf{s}}$
8:         **for** (agent(j) = 1, agent $\leq K$, agent++){
9:             Select agent's action a_j with ε
10:             Obtain y with $\{\mathbf{H}_{i,j}\}$ and collected actions $\{a_1, \ldots, a_K\}$
11:             The j-th agent stores $\{\tilde{\mathbf{s}}, a_j, y\}$ into its memory
          }
12:         Decrease ε for exploitation (optional)
      }
    //Stage 2: updating the Q-network
13:     **for** (agent = 1, agent $\leq K$, agent++){
14:         Train the agent's Q-network with saved data
      }
    //Stage 3: evaluating performance
15:     Repeat lines from 5 to 11 with $\varepsilon = 0$, where y is collected only in line 11
16:     Compute the average of y that are collected in line 15 to evaluate performance of the agents.
  }

Simulation Results (IDQL) – Training Process

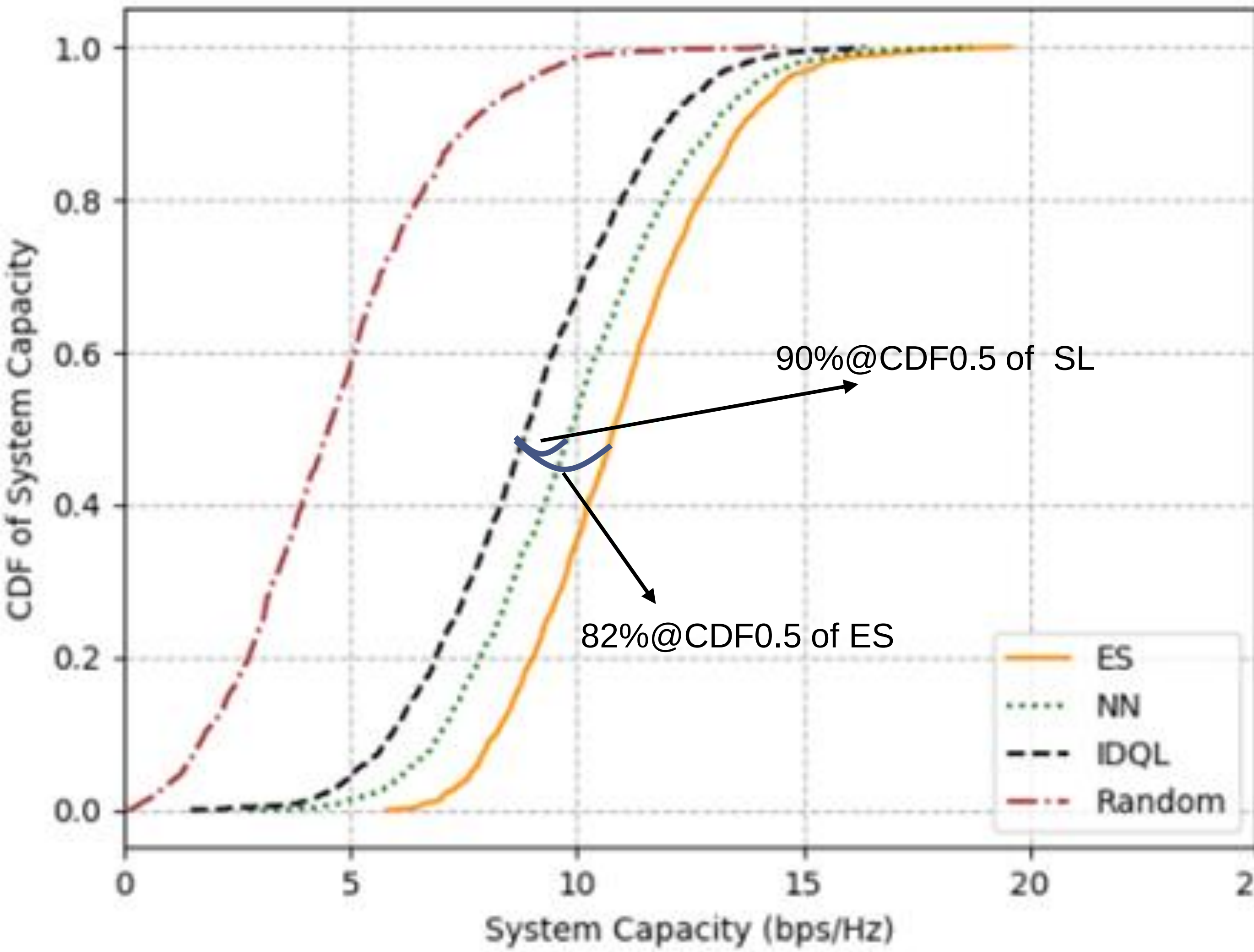
Decreasing-ε leads to best performance

Lower ε converges the fastest



All policies lead to convergence

Simulation Results (IDQL) – System Capacity



4. Future Research Direction

- The training phase of DQN is not completely independent. Full channel state information need to be shared in backhaul of all BSs in system.

- Pilot signal is also required when channel state information change (ex. Movement of UTs). A dynamic scheme for online network adjustment could be developed.

- Other base station parameters could be considered besides BS transmitting power and beamforming vector (ex. Coding scheme).

5. Reference

[1] Luong, N. C., Hoang, D. T., Gong, S., Niyato, D., Wang, P., Liang, Y. C., & Kim, D. I. (2019). Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys & Tutorials*, *21*(4), 3133-3174.

[2] Jiang, S., Chang, Y., & Fukawa, K. (2020, October). Distributed Inter-cell Interference Coordination for Small Cell Wireless Communications: A Multi-Agent Deep Q-Learning Approach. In *2020 International Conference on Computer, Information and Telecommunication Systems (CITS)* (pp. 1-5). IEEE.

[3] S. Jiang, Y. Chang, and K. Fukawa, "Joint transmit power and 3D beamforming control using neural networks for MIMO small cell systems," IEICE Technical Report, vol. RCS 2018-62, pp. 161–166, Jun. 2020.

[4] Wijaya, M. A., Fukawa, K., & Suzuki, H. (2016). Neural network based transmit power control and interference cancellation for MIMO small cell networks. IEICE Transactions on Communications, E99B(5), 1157–1169.

[5] J. Huang, R.A. Berry, and M.L. Honig, "Distributed interference compensation for wireless networks," IEEE J. Sel. Areas Commun., vol.24, no.5, pp.1074–1084, May 2006.

[6] J.G. Andrews, H. Claussen, M. Dohler, S. Rangan, and M.C. Reed, "Femtocells: Past, present, and future," IEEE J. Sel. Areas Commun., vol.30, no.3, pp.497–508, April 2012.

[7] Nasir, Y. S., & Guo, D. (2019). Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks. IEEE Journal on Selected Areas in Communications, 37(10), 2239-2250.

[8] Ge, J., Liang, Y. C., Joung, J., & Sun, S. (2020). Deep Reinforcement Learning for Distributed Dynamic MISO Downlink-Beamforming Coordination. *IEEE Transactions on Communications*, *68*(10), 6070-6085.

Thank you for your attention

liuyuchen@radio.ict.e.titech.ac.jp



Q & A