Learning Resilient Radio Resource Management Policies With Graph Neural Networks

Navid NaderiAlizadeh (navid.naderi@duke.edu)

Department of Biostatistics and Bioinformatics
Duke University

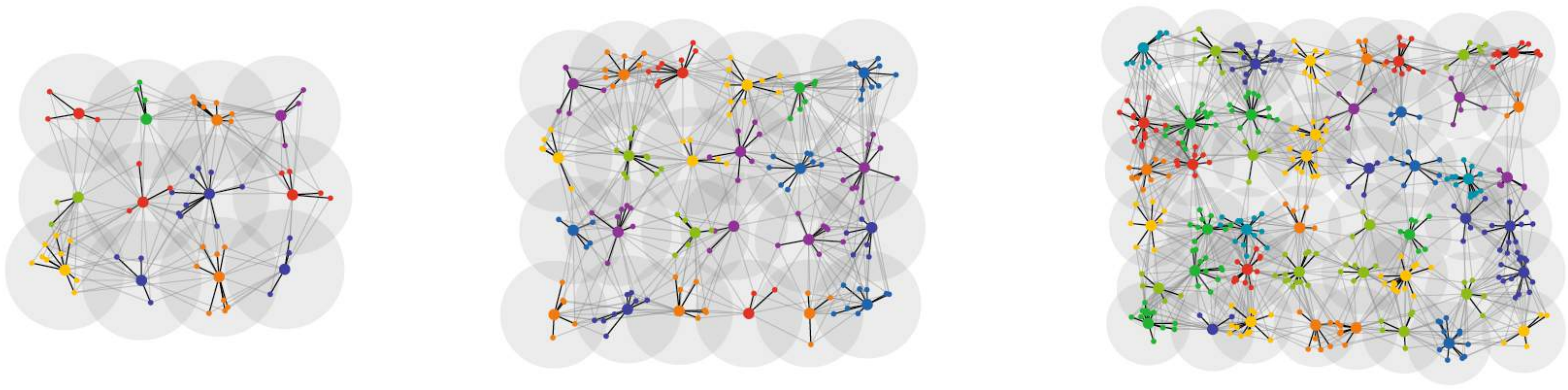
February 12, 2024

Joint work with Mark Eisen (JHU-APL) and Alejandro Ribeiro (UPenn)

Duke UNIVERSITY

Autonomous Wireless Networks

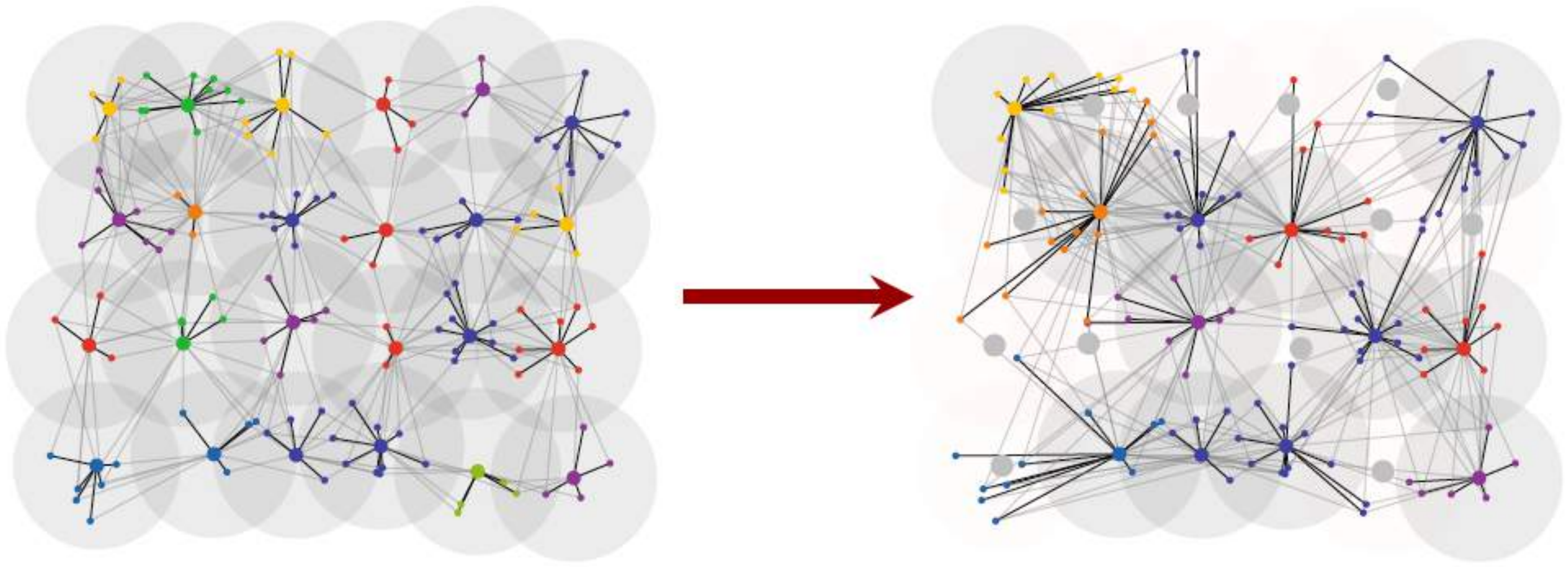
- Making operational decisions in wireless networks entails solving large-scale constrained optimization problems.
- Solving these problems is very challenging, leading to the design and use of heuristic methods.



- We can leverage data to learn better autonomous network management policies using machine learning

Learning to Adapt Problem Specifications

- Network realizations might exist in which enforcing performance constraints leads to catastrophic performance.
- Autonomous wireless networks must remain operational over a wide range of network realizations.

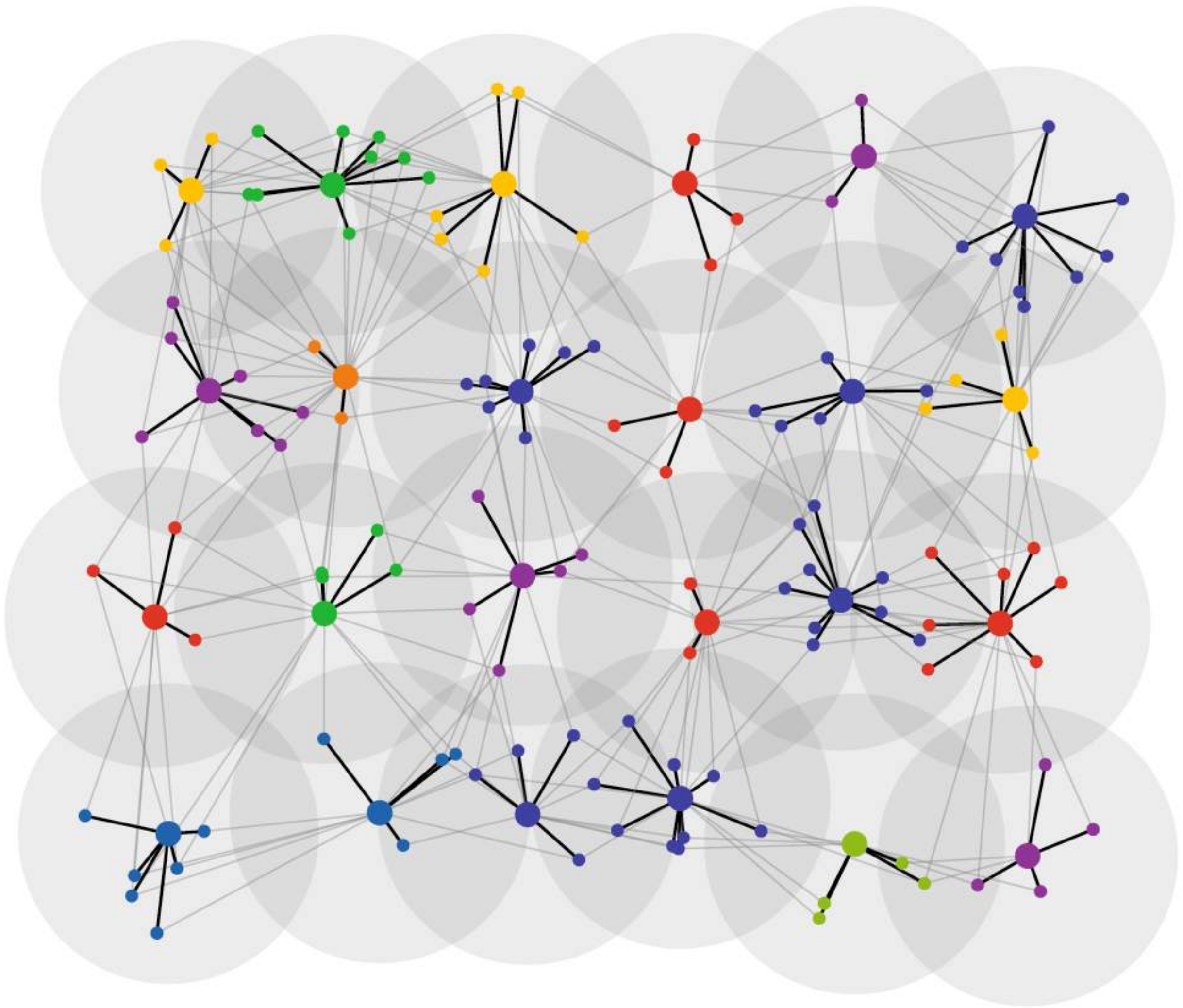


- We have developed a notion of resilience, where performance constraint levels are adapted autonomously.
  - Resilience is accomplished by elastic relaxation of constraints in proportion to marginal performance gains.

Wireless Resource Allocation

- Consider a wireless network with m access points $\{\text{AP}_1, \ldots, \text{AP}_m\}$ and n users $\{\text{UE}_1, \ldots, \text{UE}_n\}$.
- Let $\mathscr{R}_i$ denote the subset of users associated to AP_i, $i = 1,\ldots, m$.
  - We assume each user is only served by a single AP.
  - For every user UE_j, we use $[j]$ to denote its serving AP.
- Two types of resource allocation decisions we are interested in:
  - User selection $\gamma \in \{0,1\}^n$: Whether or not each user is served at each time slot.
    - No more than one user can be served by a given AP at a given time.
  - Power control $\mathbf{p} \in [0,P_{\max}]^m$: What transmit power to use at each AP.
- Shannon capacity of the link between $\text{AP}_{[j]}$ and UE_j: $f_j(\mathbf{H}, \mathbf{p}, \gamma) = \log_2(1 + \text{SINR}_j(\mathbf{H}, \mathbf{p}, \gamma))$.
- Signal-to-interference-plus-noise ratio (SINR) at UE_j:

$$\text{SINR}_j(\mathbf{H}, \mathbf{p}, \gamma) = \frac{\gamma_j \left| h_{[j]j} \right|^2 p_{[j]}}{N + \sum_{i=1,\ i\neq[j]}^{m} |h_{ij}|^2 p_i} .$$



Wireless Resource Allocation under Constraints

- The wireless network state is stochastic → Performance should be optimized over long-term windows.
- The ergodic average rate of each user is bounded by the ergodic Shannon limit $\mathbb{E}_{\mathbf{H}}[f_j(\mathbf{H}, \mathbf{p}, \gamma)]$.
- Certain applications may impose requirements on the long-term average performance of each user.

$$
\begin{aligned}
\max_{\mathbf{p},\gamma,\mathbf{x}} \quad & \mathcal{U}(\mathbf{x}), \\
\text{s.t.} \quad & \mathbf{x} \leq \mathbb{E}_{\mathbf{H}} \left[\mathbf{f}(\mathbf{H}, \mathbf{p}(\mathbf{H}), \gamma(\mathbf{H}))\right], \\
& \mathbf{x} \geq \mathbf{f}_{\min}, \\
& \mathbf{p}(\mathbf{H}) \in [0, P_{\max}]^m, \\
& \gamma(\mathbf{H}) \in \{0,1\}^n, \sum_{j \in \mathcal{R}_i} \gamma_j(\mathbf{H}) = 1, \forall i \in \{1, \ldots, m\}.
\end{aligned}
$$

Resilient Operation of Wireless Networks

- In practice, system requirements may be infeasible in some extreme scenarios.
- They could be relaxed just enough to find a feasible solution, leading to resilient resource allocation policies.

$$
\begin{aligned}
P^\star(\mathbf{z}) = \max_{\mathbf{p},\gamma,\mathbf{x}} \quad & \mathscr{U}(\mathbf{x}), \\
\text{s.t.} \quad & \mathbf{x} \leq \mathbb{E}_{\mathbf{H}}\left[\mathbf{f}(\mathbf{H}, \mathbf{p}(\mathbf{H}), \gamma(\mathbf{H}))\right], \\
& \mathbf{x} \geq \mathbf{f}_{\min} - \mathbf{z}, \\
& \mathbf{p}(\mathbf{H}) \in [0, P_{\max}]^m, \\
& \gamma(\mathbf{H}) \in \{0,1\}^n, \sum_{j \in \mathscr{R}_i} \gamma_j(\mathbf{H}) = 1, \forall i \in \{1, \ldots, m\}.
\end{aligned}
$$

- $\mathbf{z} \geq \mathbf{0}$ denote the non-negative slack variables that adapt the requirements for all users in the networks.

Resilience by Compromise

$$
\begin{aligned}
P^\star(\mathbf{z}) = \max_{\mathbf{p},\gamma,\mathbf{x}} \quad & \mathscr{U}(\mathbf{x}), \\
\text{s.t.} \quad & \mathbf{x} \leq \mathbb{E}_{\mathbf{H}}\left[\mathbf{f}(\mathbf{H}, \mathbf{p}(\mathbf{H}), \gamma(\mathbf{H}))\right], \\
& \mathbf{x} \geq \mathbf{f}_{\min} - \mathbf{z}, \\
& \mathbf{p}(\mathbf{H}) \in [0, P_{\max}]^m, \\
& \gamma(\mathbf{H}) \in \{0,1\}^n, \sum_{j \in \mathscr{R}_i} \gamma_j(\mathbf{H}) = 1, \forall i \in \{1,\ldots,m\}.
\end{aligned}
$$

- We desire the slack $\mathbf{z}$ to be as small as possible, so we associate a cost of $h(\mathbf{z}) = \frac{\alpha}{2}\|\mathbf{z}\|_2^2$ to it.

Definition: $\mathbf{z}^\star \geq \mathbf{0}$ is the optimal value of the slack if and only if $\nabla_{\mathbf{z}} P^\star(\mathbf{z})\,|_{\mathbf{z}=\mathbf{z}^\star} = \alpha \mathbf{z}^\star$.

- Resilient policies compromise to adapt: The more constraints are relaxed, the more the objective yields.
  - Elastic relaxation of constraints in proportion to marginal performance gains leads to resilience.

Finding the Optimal Slack Levels

- We can include the slack cost as a regularization term, and add the slacks as optimization variables:

$$
\begin{aligned}
P^\star = \max_{\mathbf{p},\gamma,\mathbf{x},\mathbf{z}} \quad & \mathscr{U}(\mathbf{x}) - \frac{\alpha}{2}\|\mathbf{z}\|_2^2, \\
\text{s.t.} \quad & \mathbf{x} \leq \mathbb{E}_{\mathbf{H}}\left[\mathbf{f}(\mathbf{H}, \mathbf{p}(\mathbf{H}), \gamma(\mathbf{H}))\right], \\
& \mathbf{x} \geq \mathbf{f}_{\min} - \mathbf{z}, \\
& \mathbf{p}(\mathbf{H}) \in [0, P_{\max}]^m, \\
& \gamma(\mathbf{H}) \in \{0,1\}^n, \sum_{j \in \mathscr{R}_i} \gamma_j(\mathbf{H}) = 1, \forall i \in \{1, \ldots, m\}, \\
& \mathbf{z} \geq \mathbf{0}.
\end{aligned}
$$

Policy Parameterization

- In this classical formulation, resource allocation decisions must be recalculated for any given network state $\mathbf{H}$.
  - This makes learning and deploying such a policy infeasible in practice.
- We parameterize the power control and user selection policies: $\mathbf{p}(\mathbf{H}) \rightarrow \mathbf{p}(\mathbf{H}; \theta^{\mathbf{p}}), \gamma(\mathbf{H}) \rightarrow \gamma(\mathbf{H}; \theta^{\gamma})$.
- The advantage of parameterization is that we do not need to solve the problem online to find the decisions.

Unparameterized formulation

$$\max_{\mathbf{p},\gamma,\mathbf{x},\mathbf{z}} \quad \mathscr{U}(\mathbf{x}) - \frac{\alpha}{2}\|\mathbf{z}\|_2^2,$$
$$\text{s.t.} \quad \mathbf{x} \leq \mathbb{E}_{\mathbf{H}}\left[\mathbf{f}(\mathbf{H}, \mathbf{p}(\mathbf{H}), \gamma(\mathbf{H}))\right],$$
$$\mathbf{x} \geq \mathbf{f}_{\min} - \mathbf{z},$$
$$\mathbf{p}(\mathbf{H}) \in [0, P_{\max}]^m, \mathbf{z} \geq \mathbf{0},$$
$$\gamma(\mathbf{H}) \in \{0,1\}^n, \sum_{j \in \mathscr{R}_i} \gamma_j(\mathbf{H}) = 1, \forall i.$$

Parameterized formulation

$$\max_{\theta^{\mathbf{p}},\theta^{\gamma},\mathbf{x},\mathbf{z}} \quad \mathscr{U}(\mathbf{x}) - \frac{\alpha}{2}\|\mathbf{z}\|_2^2,$$
$$\text{s.t.} \quad \mathbf{x} \leq \mathbb{E}_{\mathbf{H}}\left[\mathbf{f}(\mathbf{H}, \mathbf{p}(\mathbf{H}; \theta^{\mathbf{p}}), \gamma(\mathbf{H}; \theta^{\gamma}))\right],$$
$$\mathbf{x} \geq \mathbf{f}_{\min} - \mathbf{z},$$
$$\mathbf{p}(\mathbf{H}; \theta^{\mathbf{p}}) \in [0, P_{\max}]^m, \mathbf{z} \geq \mathbf{0}$$
$$\gamma(\mathbf{H}; \theta^{\gamma}) \in \{0,1\}^n, \sum_{j \in \mathscr{R}_i} \gamma_j(\mathbf{H}; \theta^{\gamma}) = 1, \forall i.$$

Learning in the Dual Domain

- We move to the Lagrangian dual domain and associate a set of dual variables λ, μ to the constraints.
  - The remaining constraints on $\mathbf{p}, \gamma, \mathbf{z}$ are assumed implicit (i.e., automatically satisfied by the parameterization)

- The Lagrangian function can then be written as

$$\mathscr{L}(\theta^{\mathbf{p}}, \theta^{\gamma}, \mathbf{x}, \mathbf{z}, \lambda, \mu) \; = \mathscr{U}(\mathbf{x}) - \frac{\alpha}{2}\|\mathbf{z}\|_2^2 - \lambda^T \left[\mathbf{x} - \mathbb{E}_{\mathbf{H}}\left[\mathbf{f}(\mathbf{H}, \mathbf{p}(\mathbf{H}; \theta^{\mathbf{p}}), \gamma(\mathbf{H}; \theta^{\gamma}))\right]\right] - \mu^T \left[\mathbf{f}_{\min} - \mathbf{z} - \mathbf{x}\right].$$

- We then seek to maximize the Lagrangian over the primal variables while minimizing it over λ, μ:

$$D^\star = \min_{\lambda,\mu} \max_{\theta^{\mathbf{p}}, \theta^{\gamma}, \mathbf{x}, \mathbf{z}} \mathscr{L}(\theta^{\mathbf{p}}, \theta^{\gamma}, \mathbf{x}, \mathbf{z}, \lambda, \mu)$$

The Duality Gap is Bounded

- The duality gap of the unparameterized problem is null, but that is not the case with paramaterization.
- Nevertheless, the duality gap with parameterization is bounded for near-universal parameterizations.

Theorem: Under certain assumptions, for near-universal parameterizations with degree ϵ and an L-Lipschitz performance function f, the dual value $D^\star$ is bounded as

$$P^\star - \epsilon L \|[\lambda^\star; \mu^\star]\|_1 \leq D^\star \leq P^\star,$$

- The closeness of the two problems allows us to use stochastic primal-dual methods to find the optimal policies.

N. NaderiAlizadeh, M. Eisen, and A. Ribeiro, "Learning resilient radio resource management policies with graph neural networks," IEEE Transactions on Signal Processing, Mar. 2023.

Iterative Unsupervised Primal-Dual Updates

$$\mathscr{L}(\theta^{\mathbf{p}}, \theta^{\gamma}, \mathbf{x}, \mathbf{z}, \lambda, \mu) = \mathscr{U}(\mathbf{x}) - \frac{\alpha}{2}\|\mathbf{z}\|_2^2 - \lambda^T \left[\mathbf{x} - \mathbb{E}_{\mathbf{H}}\left[\mathbf{f}(\mathbf{H}, \mathbf{p}(\mathbf{H}; \theta^{\mathbf{p}}), \gamma(\mathbf{H}; \theta^{\gamma}))\right]\right] - \mu^T \left[\mathbf{f}_{\min} - \mathbf{z} - \mathbf{x}\right].$$

Update policy parameters

$$\theta^{\mathbf{p}}_{k+1} = \theta^{\mathbf{p}}_{k} + \eta_{\mathbf{p}} \nabla_{\theta^{\mathbf{p}}} \left\{ \lambda^T \mathbb{E}_{\mathbf{H}} \left[\mathbf{f}(\mathbf{H}, \mathbf{p}(\mathbf{H}), \gamma(\mathbf{H}))\right] \right\}$$

$$\theta^{\gamma}_{k+1} = \theta^{\gamma}_{k} + \eta_{\gamma} \nabla_{\theta^{\gamma}} \left\{ \lambda^T \mathbb{E}_{\mathbf{H}} \left[\mathbf{f}(\mathbf{H}, \mathbf{p}(\mathbf{H}), \gamma(\mathbf{H}))\right] \right\}$$

Update rate and slack variables

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \eta_{\mathbf{x}} \left(\nabla_{\mathbf{x}_k} \left\{ \mathscr{U}(\mathbf{x}_k) \right\} + \mu_k - \lambda_k \right)$$

$$\mathbf{z}_{k+1} = \left[\mathbf{z}_k + \eta_{\mathbf{z}} \left(\mu_k - \alpha \mathbf{z}_k \right) \right]_+$$

Update dual variables

$$\lambda_{k+1} = \left[\lambda_k - \eta_{\lambda} \left(\mathbf{x}_k - \mathbb{E}_{\mathbf{H}} \left[\mathbf{f}(\mathbf{H}, \mathbf{p}(\mathbf{H}), \gamma(\mathbf{H}))\right] \right) \right]_+$$

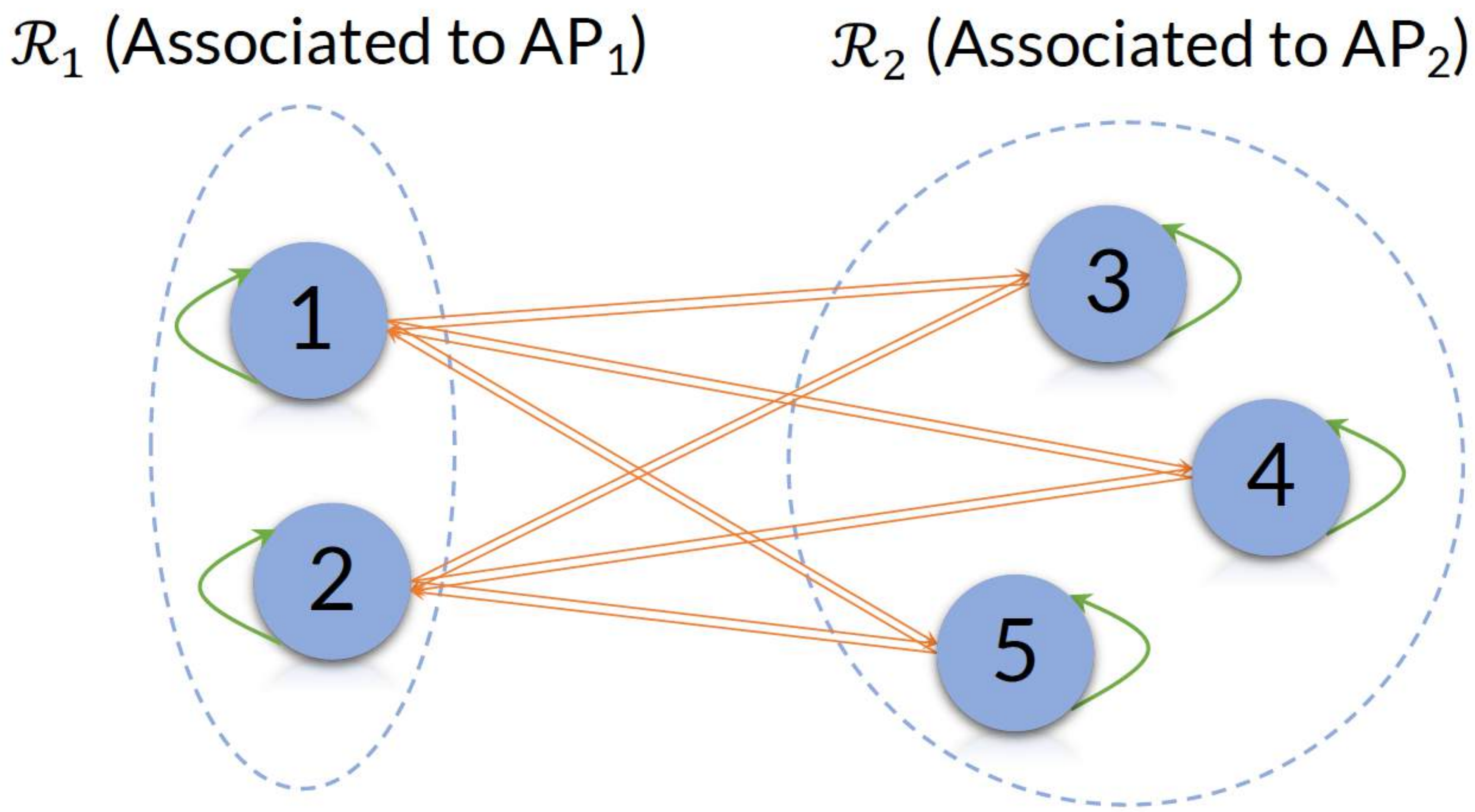
$$\mu_{k+1} = \left[\mu_k - \eta_{\mu} \left(\mathbf{f}_{\min} - \mathbf{z}_k - \mathbf{x}_k \right) \right]_+$$

$k \leftarrow k + 1$

Given a dataset with a finite number of samples, expectations are replaced with empirical means.

Modeling the Network Data Structure as a Graph

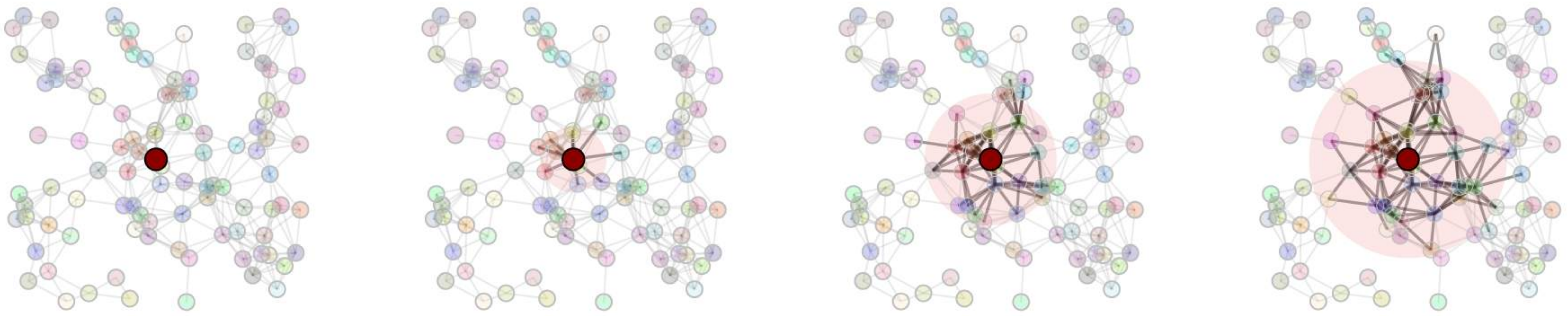
- We consider the data structure in the form of a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, w)$.
  - $\mathcal{V} = \{1, \ldots, n\}$ denotes the set of nodes, each representing a user.
  - $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of directed edges, with self-loop signal edges, and cross-AP interference edges.
  - $w : \mathcal{E} \to \mathbb{R}$ denotes the edge weight function, which we define as $w(h_{ij}) \propto \log\left(P_{\max} |h_{[i]j}|^2 / N\right)$.



Graph Neural Network (GNN) Parameterizations

- We leverage graph neural network (GNN) architectures to parameterize the resource allocation policies.
- Each node $v \in \mathscr{V}$ is endowed with a vector of initial features $\mathbf{y}_v^0 \in \mathbb{R}^{F_0}$ (e.g., proportional-fairness ratio, SNR, etc.)
- Node features are updated through a sequence of L message-passing GNN layers as

$$\mathbf{y}_v^l = \Psi^l \left(\mathbf{y}_v^{l-1}, w(v, v), \left\{ \mathbf{y}_u^{l-1}, w(u, v) \right\}_{u \in \mathscr{V} \setminus \{v\}:(u,v) \in \mathscr{E}}; \theta^l \right), \forall l \in \{1,2,\ldots,L\} .$$



GNN Outputs Drive Resource Allocation Decisions

- Let $\mathbf{s}_v = \mathbf{y}_v^L \in \mathbb{R}^{F_L}, \forall v \in \mathcal{V}$ denote the final node features, or node embeddings, at the output of the GNN.
- The node embeddings are converted to resource allocation decisions using linear projections $\mathbf{b}_\mathbf{p}, \mathbf{b}_\gamma \in \mathbb{R}^{F_L}$.
  - Power control: For each AP$_i$, $i \in \{1,\ldots,m\}$, we derive its transmit power level as

$$p_i = P_{\max} \cdot \sigma\left(\frac{1}{|\mathcal{R}_i|}\mathbf{b}_\mathbf{p}{}^T \sum_{j\in\mathcal{R}_i} \mathbf{s}_j\right).$$

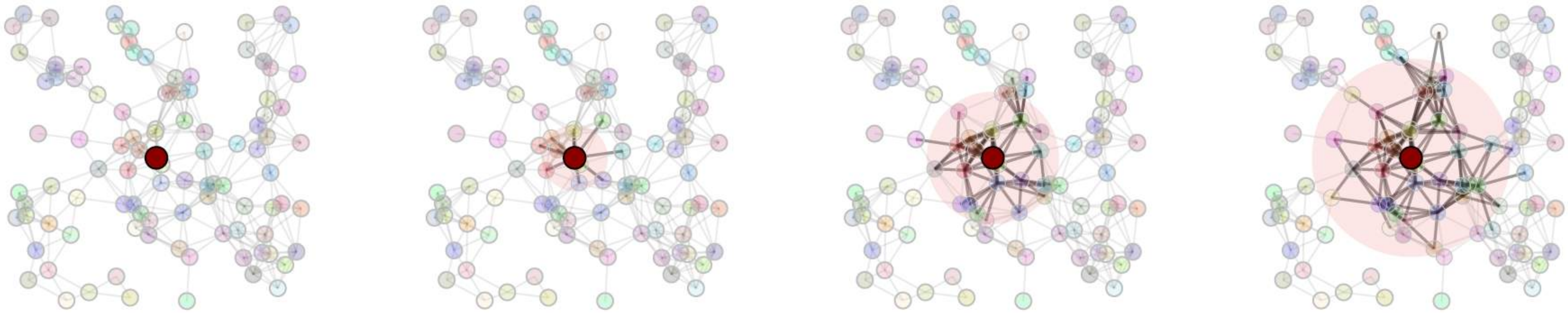
  - User selection: For each AP$_i$, $i \in \{1,\ldots,m\}$, each of its associated users UE$_j$, $j \in \mathcal{R}_i$ is served with probability

$$\gamma_j = \text{Softmax}_{\mathcal{R}_i}\left(\mathbf{b}_\gamma^T \mathbf{s}_j\right) = \frac{\exp\left(\mathbf{b}_\gamma{}^T \mathbf{s}_j\right)}{\sum_{k\in\mathcal{R}_i} \exp\left(\mathbf{b}_\gamma{}^T \mathbf{s}_k\right)}.$$

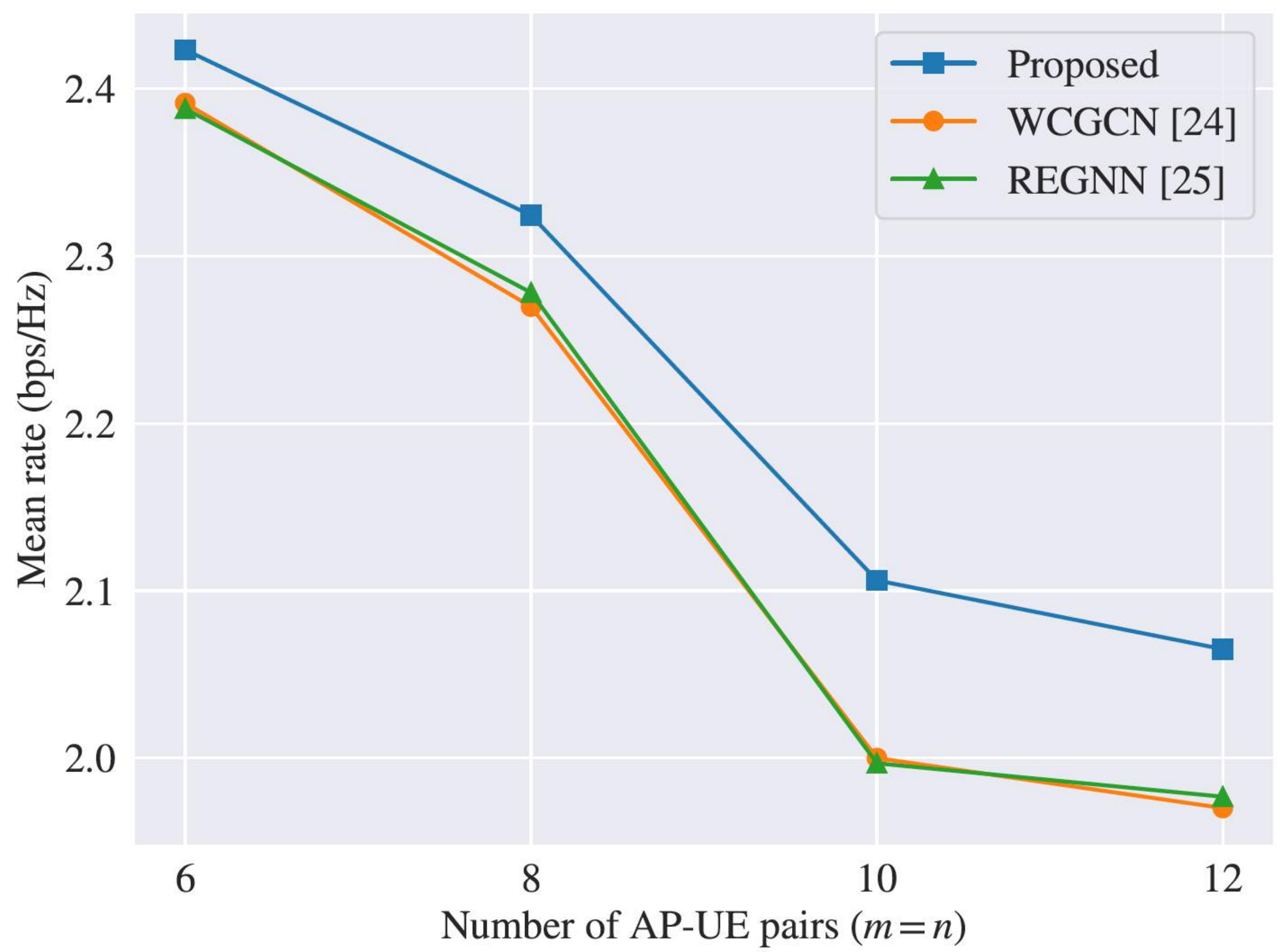
GNNs Are Scalable And Transferable

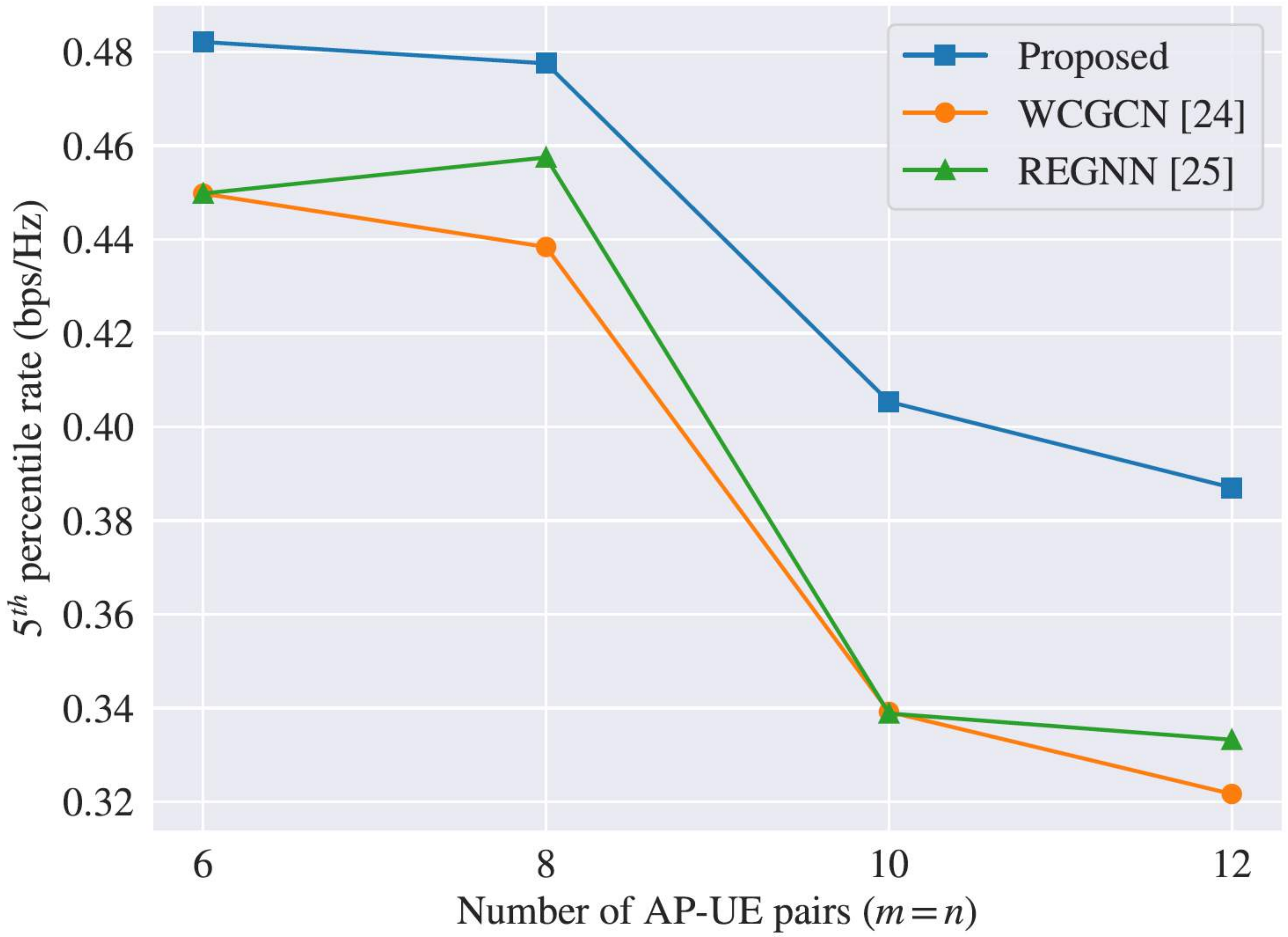
- GNNs exploit the regularities of the graph, therefore leading to the scalability of GNN-based policies.
- GNNs transfer across different scales: GNNs trained on small graphs can be executed on larger graphs.

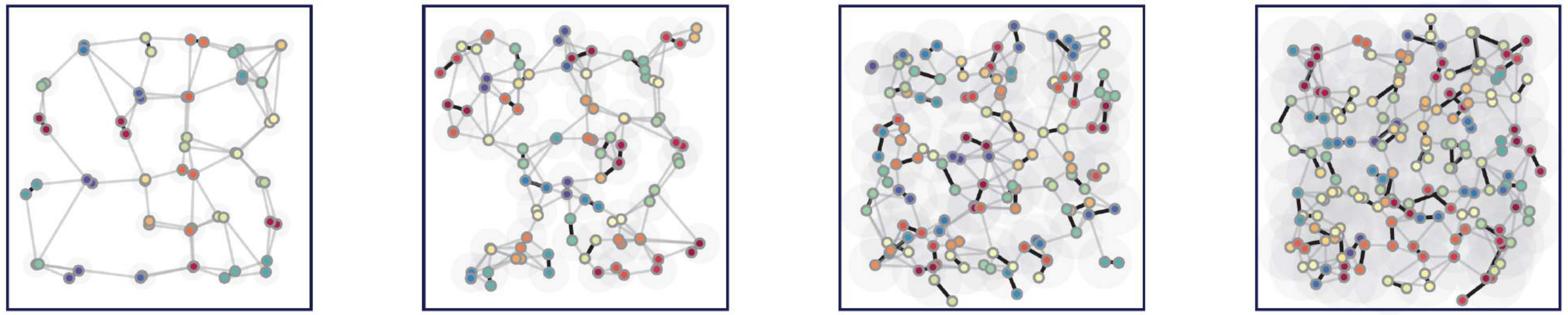
$$\mathbf{y}_v^l = \Psi^l\left(\mathbf{y}_v^{l-1}, w(v,v), \{\mathbf{y}_u^{l-1}, w(u,v)\}_{u\in\mathcal{V}\setminus\{v\}:(u,v)\in\mathcal{E}}; \theta^l\right), \forall l \in \{1,2,\ldots,L\}.$$



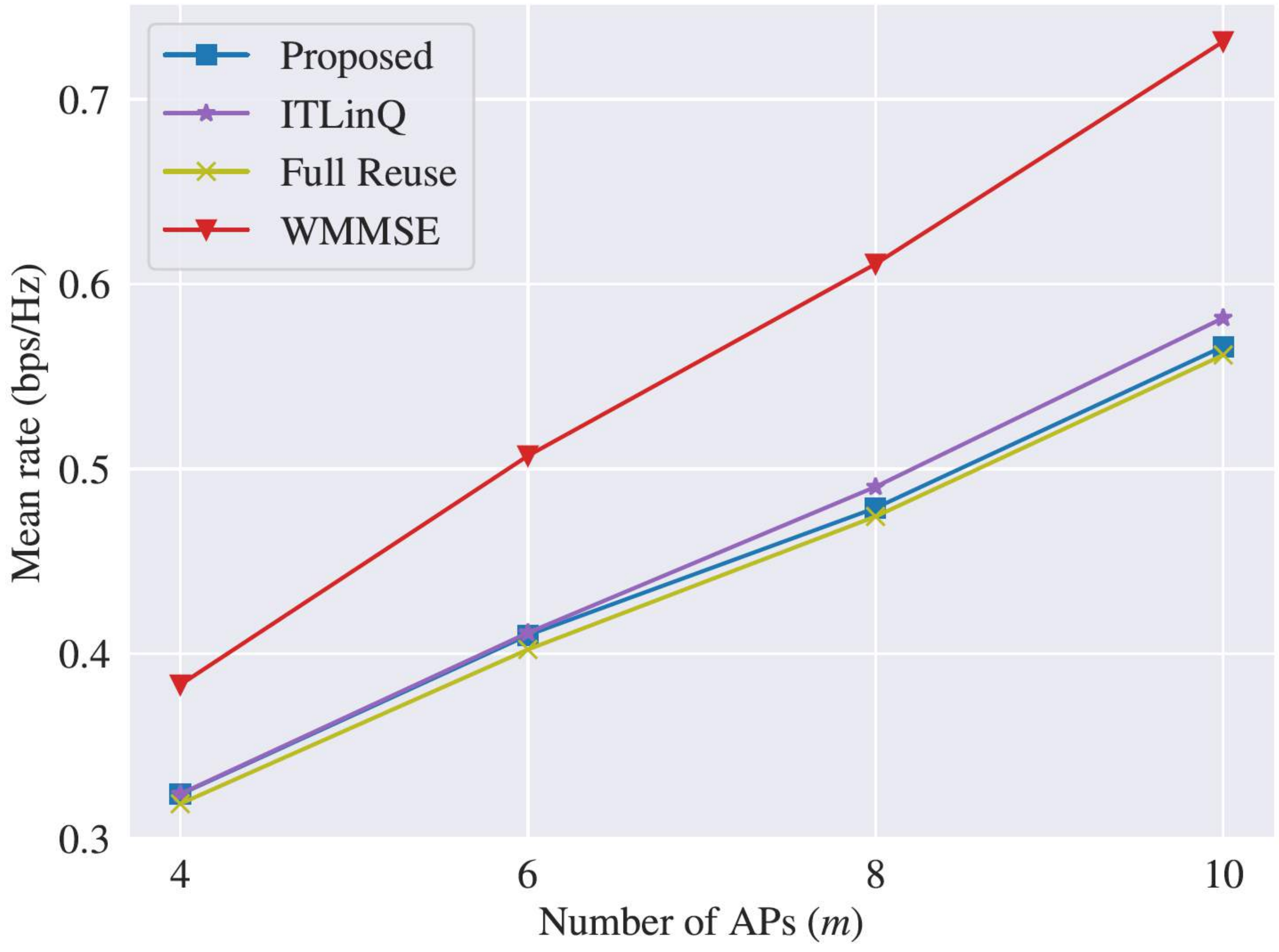
Scalability of Power Control Policies ($m = n$)

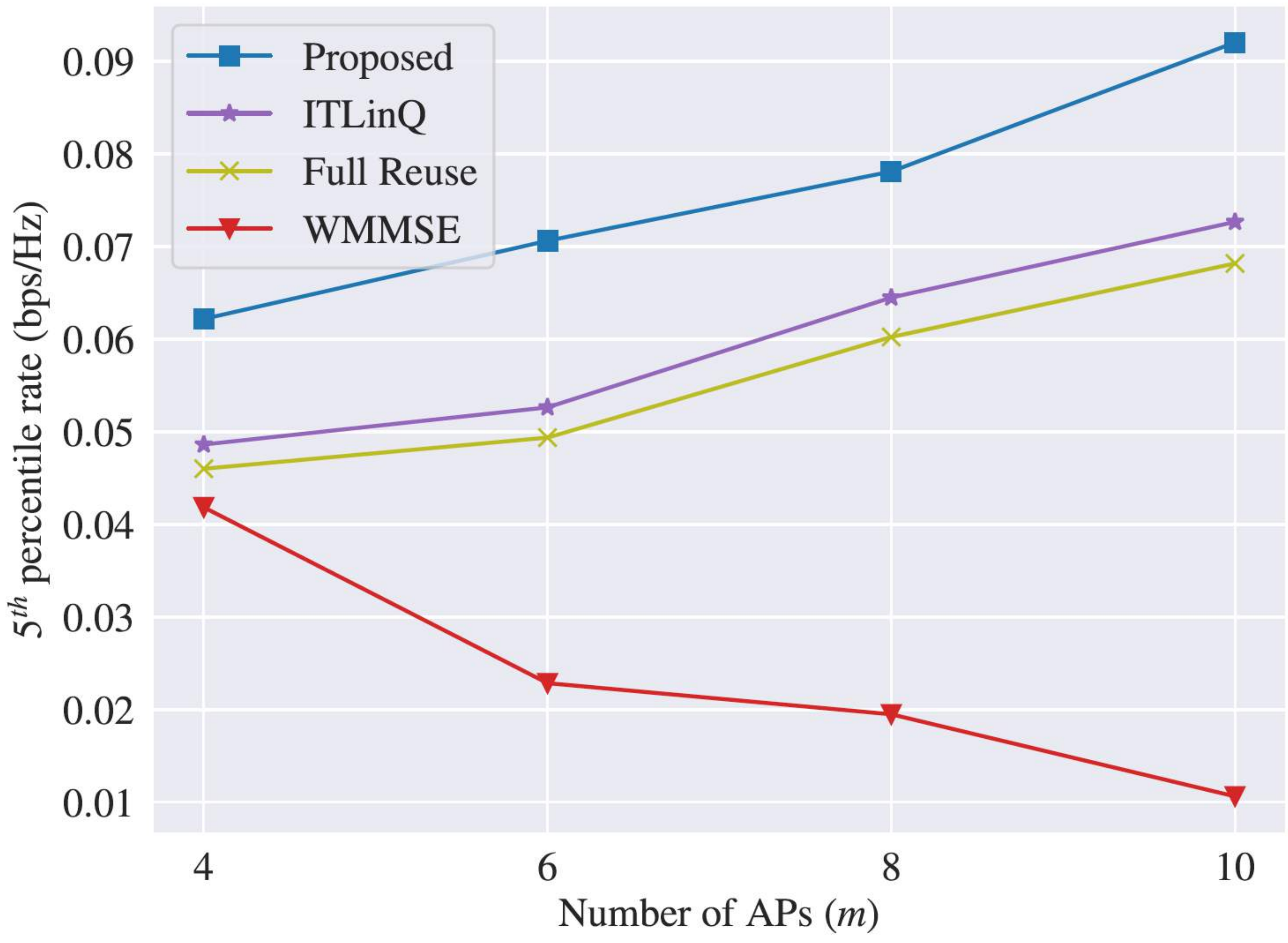


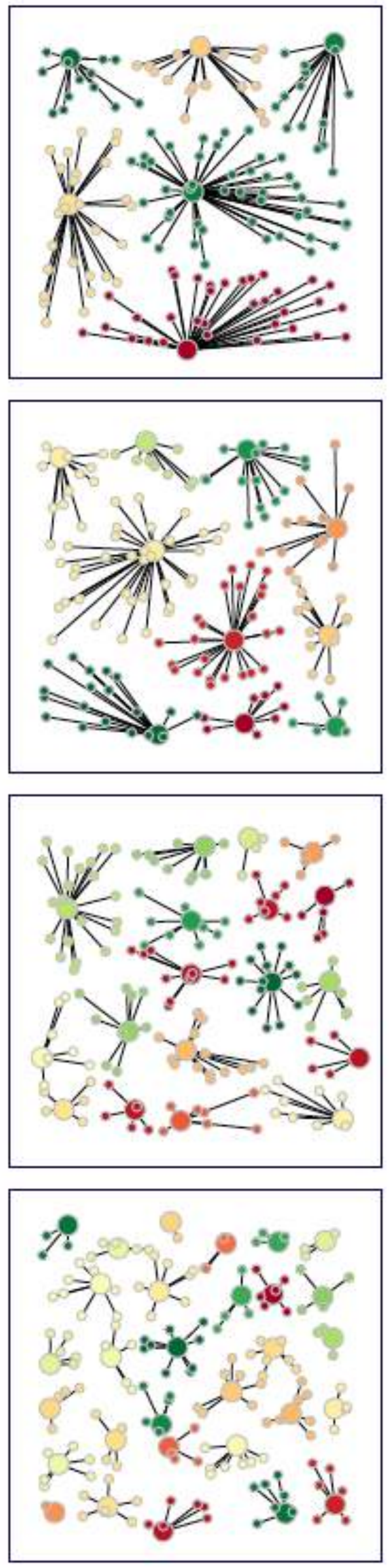




Scalability with the Number of APs ($n = 40$)

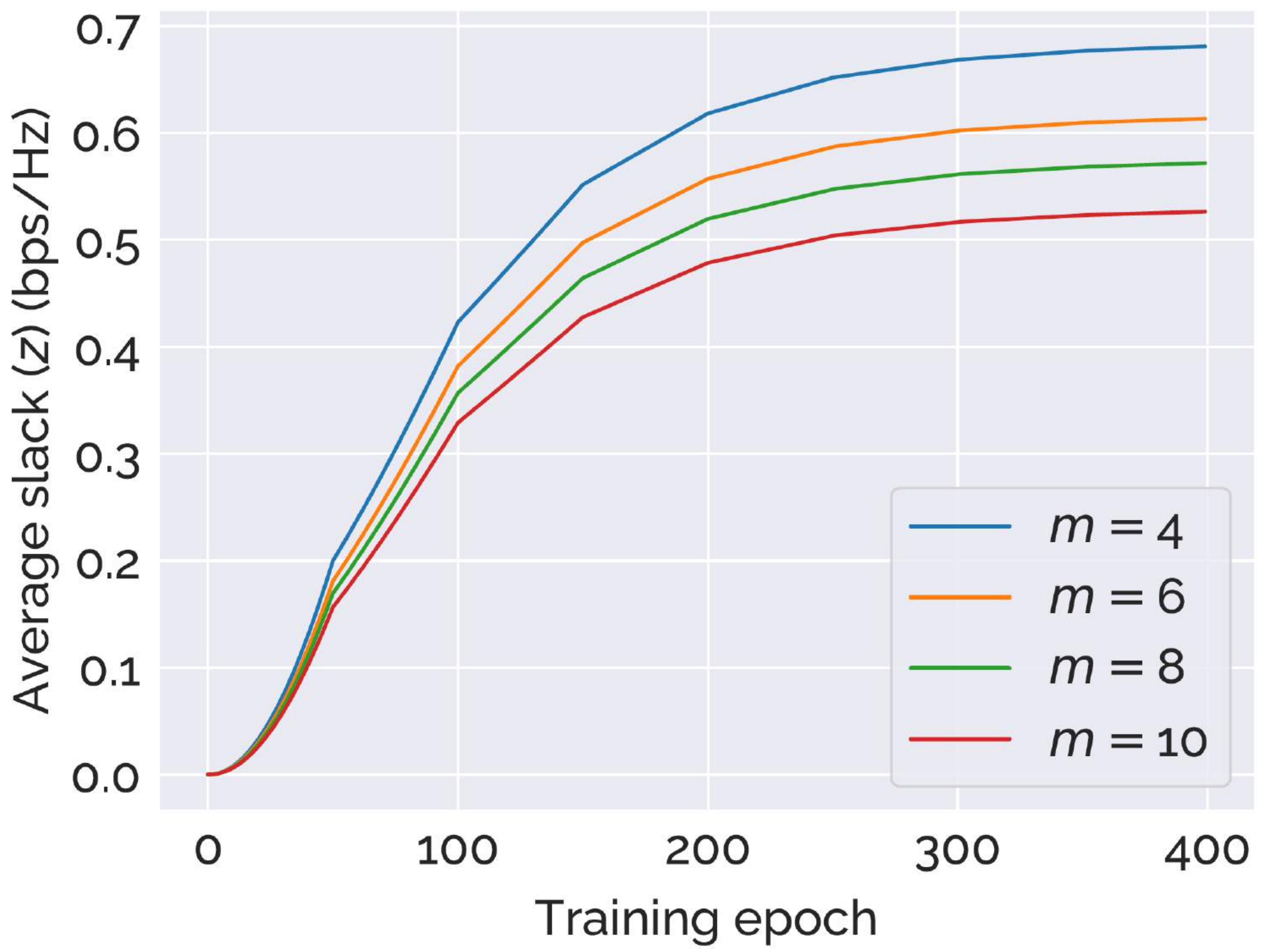


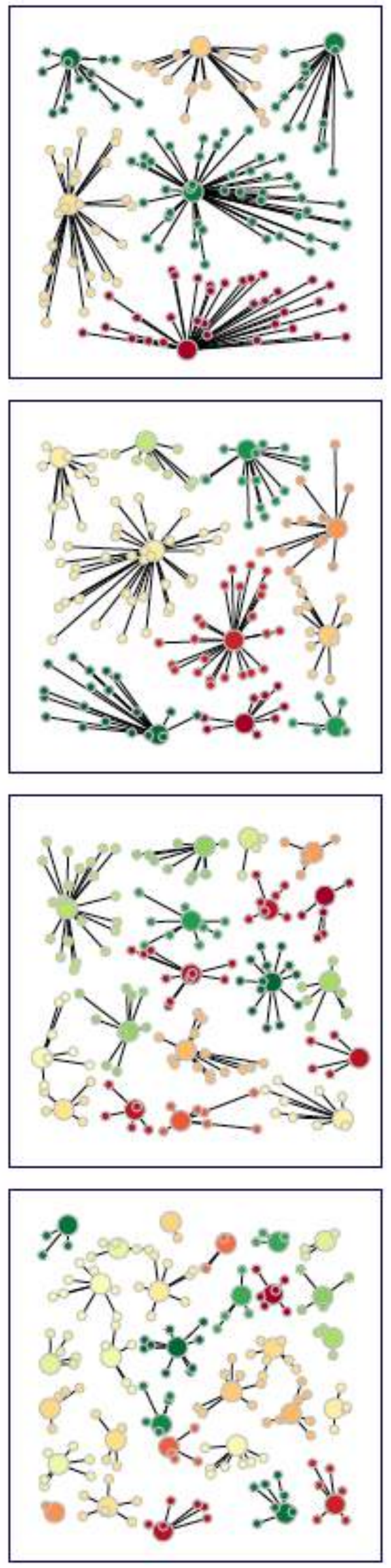




Optimal Slack Values Reflect Network Load

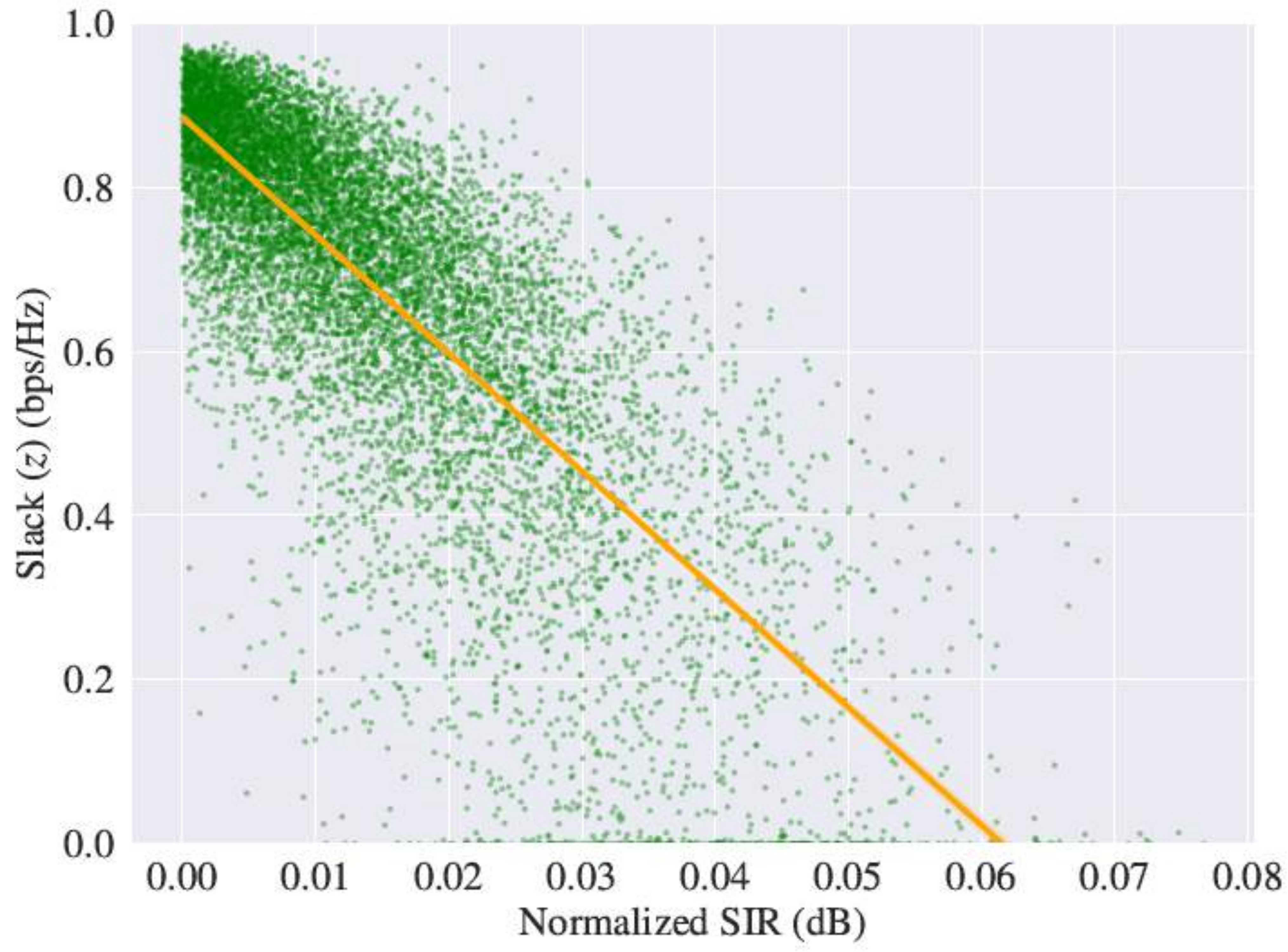
Constraints are relaxed further as the transmission resources become more limited.

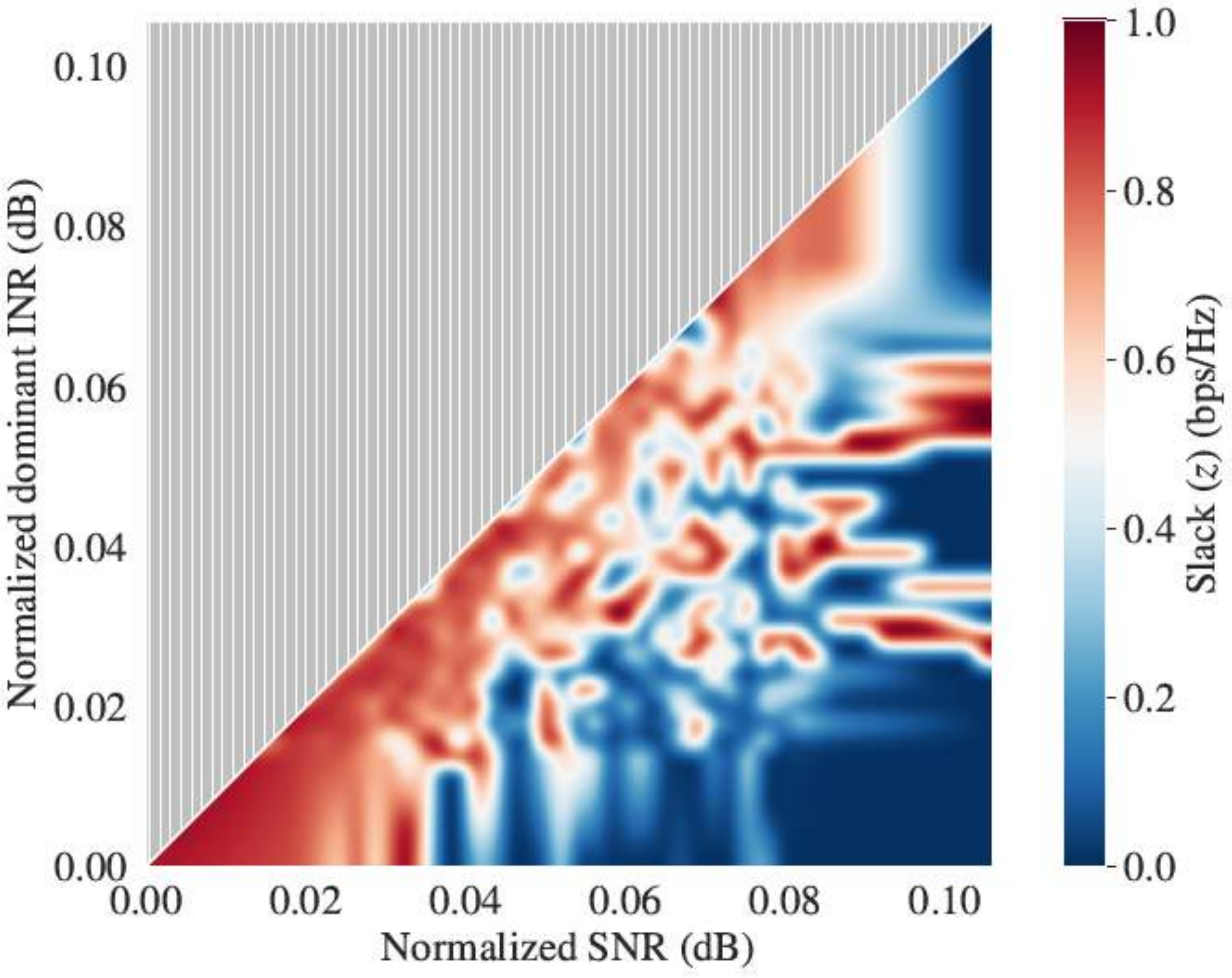




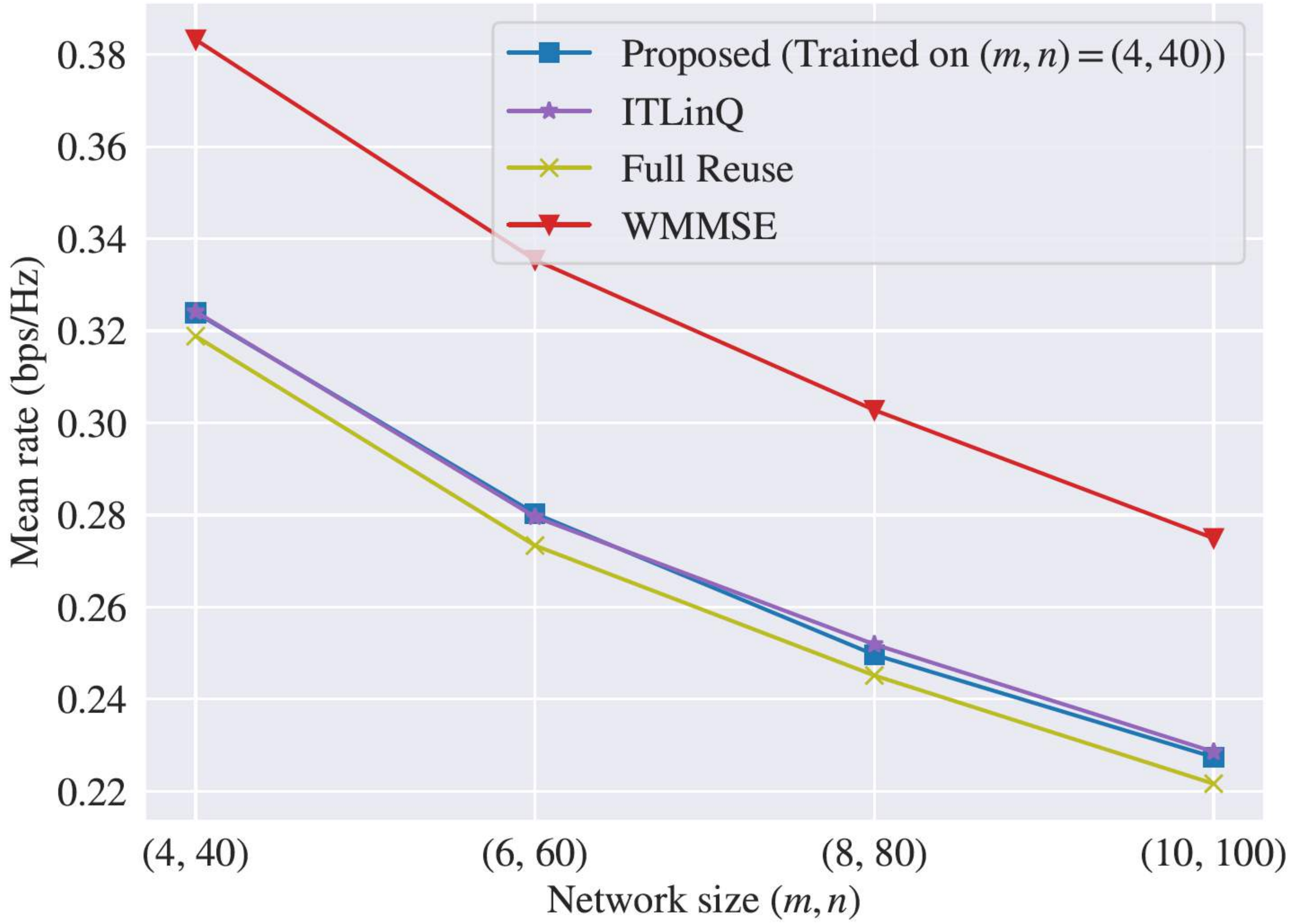
Slack Grows for Users in Unfavorable Conditions

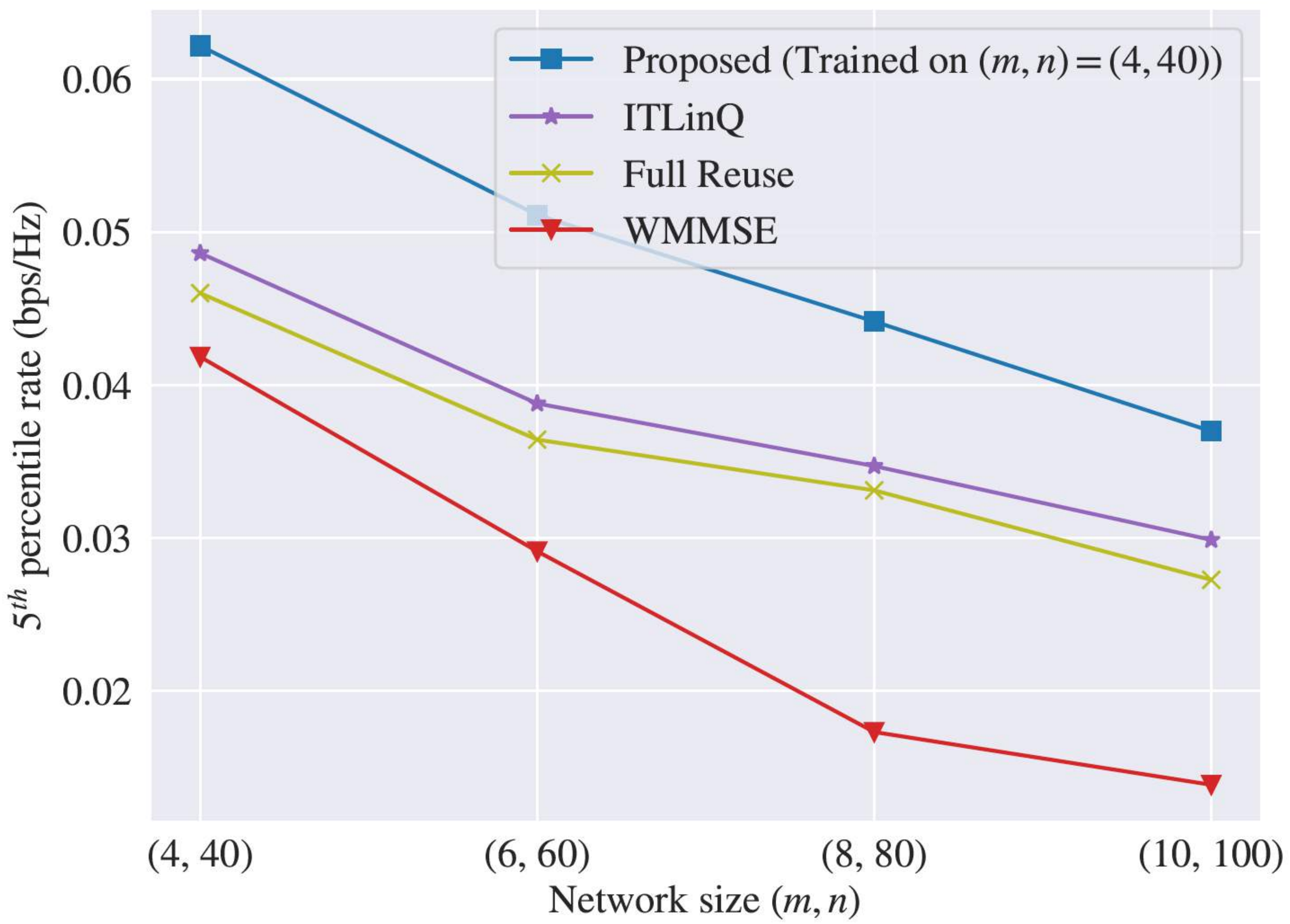
Users with low signal-to-noise ratio (SNR) and/or high interference-to-noise ratio (INR) levels have higher slacks.

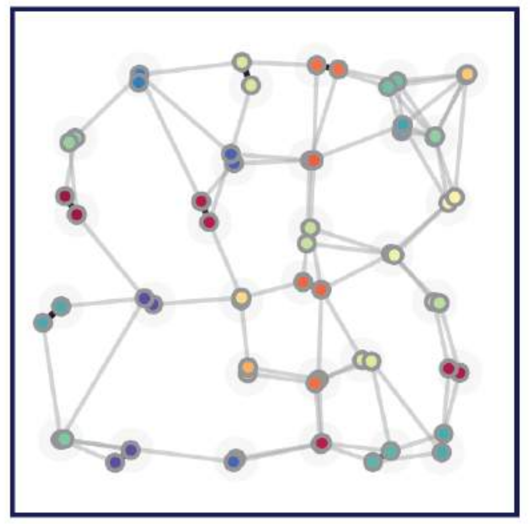


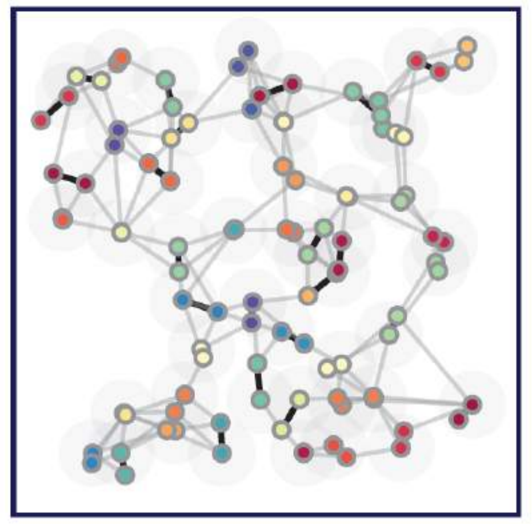


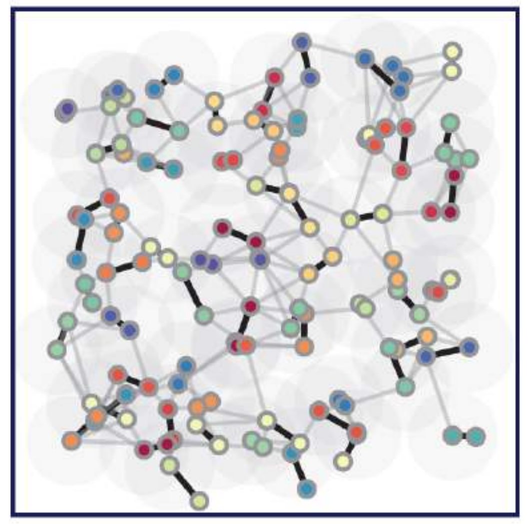
Transferability of GNN-Based Policies

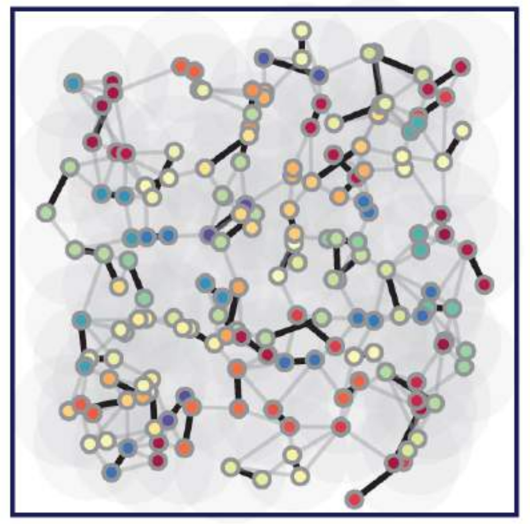






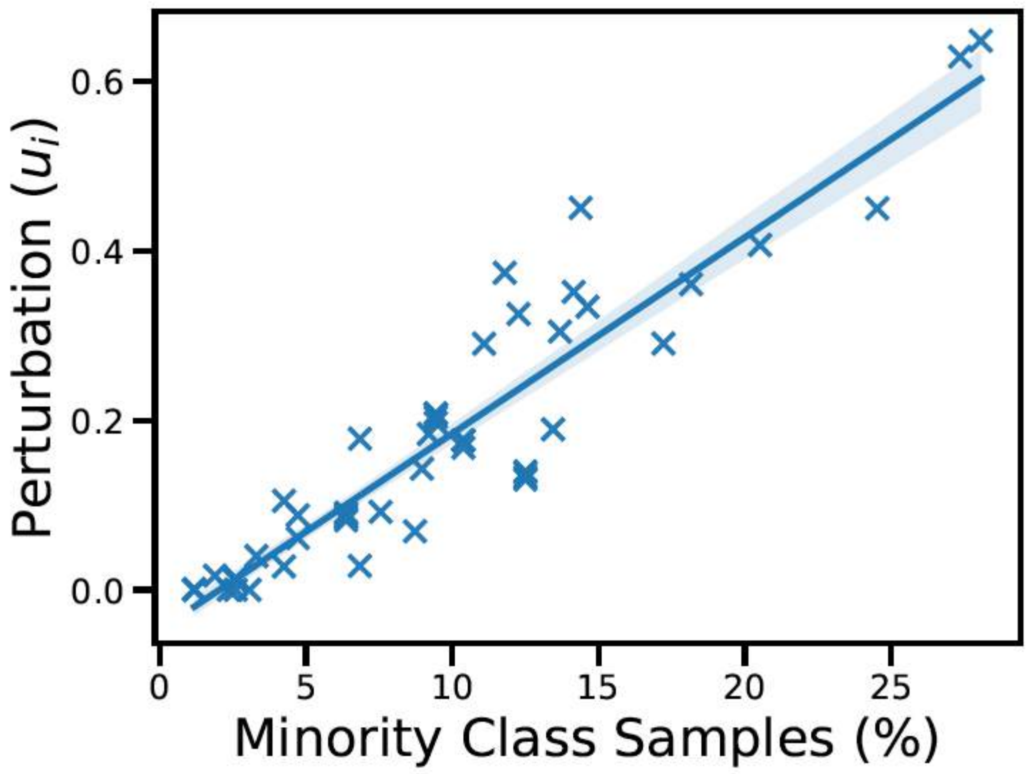


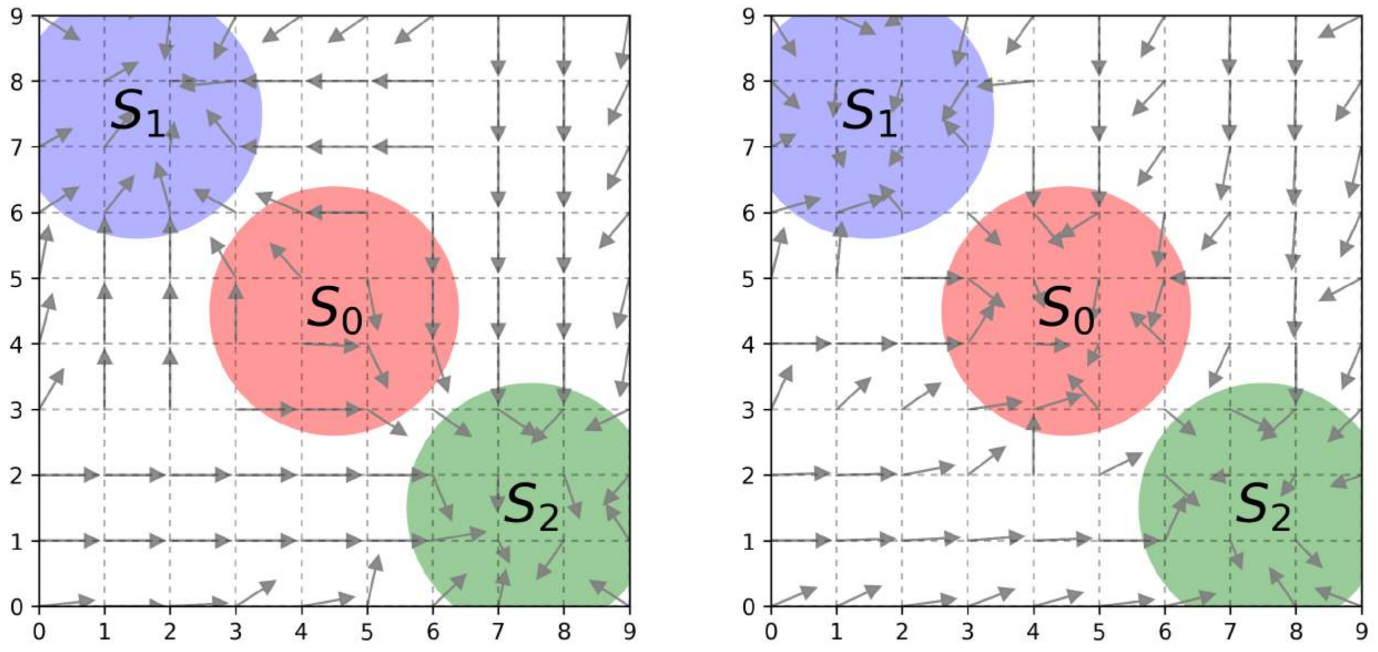




Concluding Remarks

- We considered the problem of allocating limited resources in wireless networks under requirements/constraints.
- Under adverse, extreme conditions, enforcing such requirements may lead to catastrophic failure.
- We introduced the notion of resilience, which adapts the requirements just enough to make them feasible.
- GNN-based resilient user selection and power control policies outperform baselines in multi-user wireless networks.
- The notion of resilience is also applicable to other areas, such as federated learning and reinforcement learning.





(a) non-resilient policy (b) resilient policy

Ignacio Hounie, Alejandro Ribeiro, and Luiz F. O. Chamon, "Resilient constrained learning," NeurIPS 2023.
Dongsheng Ding, Zhengyan Huan, and Alejandro Ribeiro, "Resilient constrained reinforcement learning," arXiv preprint arXiv:2312.17194, Dec. 2023.

Learning Resilient Radio Resource Management Policies With Graph Neural Networks

Navid NaderiAlizadeh (navid.naderi@duke.edu)

Department of Biostatistics and Bioinformatics
Duke University