Convex Optimization Project Report - MMSE Based MIMO Channel Estimator Via Primal-Dual Optimization Mehthod with Neural Network

312510199 Jing-Hao, Li and 312510197 Zhao-Jie, Luo
Advisor: Professor Carrson C. Fung
Jun. 12, 2024

I. Introduction

Channel estimation plays a crucial role in communication systems. Its main purpose is to determine the paths through which signals propagate from the transmitter to the receiver and to understand the signal attenuation and distortion along these paths. The accuracy of channel estimation directly impacts the quality of signal decoding at the receiver, making it an essential step in many communication systems.

In many cases, communication signals are affected by factors such as noise, multipath effects, and interference during transmission, all of which can lead to signal attenuation and distortion. By performing channel estimation, we can better understand these effects and take appropriate compensation measures to ensure the reliable transmission of signals. Channel estimation can also be used to optimize signal transmission and reception strategies, thereby improving system performance and efficiency.

In this work, a neural network based minimun mean square estimator, trained using deep reinforcement learning(DRL) and primal-dual optimization technique is proposed for MIMO channel estimation. It's posited that the proped estimation will outperform the least-square(LS) and linear MMSE(LMMSE) estimator.

II. Problem setup

Assume that the communication signals are affected by additive white Gaussian noise, fading effects, and phase shift during transmission, consequently, the system model can be represented as:

$$\mathbf{Y} = \mathbf{H}\mathbf{X} + \mathbf{W}, \tag{1}$$

where $\mathbf{X} \in \mathbb{C}^{n_T \times T}$ represents the transmitted pilot signal, $\mathbf{W} \in \mathbb{C}^{n_R \times T}$ represents the additive white Gaussian noise with zero mean and unit variance, $\mathbf{Y} \in \mathbb{C}^{n_R \times T}$ represents the received signal, and $\mathbf{H} \in \mathbb{C}^{n_R \times n_T}$ is the channel matrix containing channel coefficients which represents $n_R \times n_T$ flat fading channel. that we aim to estimate. Here, n_T and n_R denote the number of transmit and recieve antennas, respectively. T denotes the multiple of n_T, and one unit of n_T is commonly used. Since the column of $\mathbf{X}$ are usually orthonormal, without loss of generality, they are assumed to equal to columns of $\mathbf{I}_{n_T}$.

Vectorizing (1), it becomes

$$\begin{aligned}\mathbf{y} &= (\mathbf{X} \otimes \mathbf{I}_{n_R})\mathbf{h} + \mathbf{w} \\ &= \tilde{\mathbf{X}}\mathbf{h} + \mathbf{w},\end{aligned} \tag{2}$$

where $\tilde{\mathbf{X}} \triangleq (\mathbf{X} \otimes \mathbf{I}_{n_R}) \in \mathbb{C}^{n_T n_R \times T n_R}$.

III. Problem Formulation

The MMSE channel estimator aims to minimize the Bayesian MSE of the channel estimate $\hat{\mathbf{h}}$

$$\hat{\mathbf{h}} = \arg\min_{\hat{\mathbf{h}}} \mathrm{BMSE}(\hat{\mathbf{h}}), \tag{3}$$

where

$$\mathrm{BMSE}(\hat{\mathbf{h}}) = \mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\left\|\mathbf{h} - \hat{\mathbf{h}}\right\|_2^2\right]. \tag{4}$$

The closed-form solution of this MMSE problem is $\hat{\mathbf{h}} = \mathbb{E}_{\mathbf{h}|\mathbf{y}}[\mathbf{h}|\mathbf{y}]$. However, it requires knowledge about the conditional probability $p(\mathbf{h}|\mathbf{y})$, which may be unknown and/or difficult to obtain. Then (3) can be written as

$$\min_{\hat{\mathbf{h}}} \mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\left\|\mathbf{h} - \hat{\mathbf{h}}\right\|_2^2\right]. \tag{5}$$

IV. Proposed Approach

A. Primal-Dual Method

(5) can be reformulated to its epigraph form as

$$\begin{aligned}\min_{t,\hat{\mathbf{h}}} \quad & t \\ \text{s.t.} \quad & \mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\left\|\mathbf{h} - \hat{\mathbf{h}}\right\|_2^2\right] \le t.\end{aligned} \tag{6}$$

Then its Lagrangian function can be written as

$$\mathcal{L}\left(\hat{\mathbf{h}}, t, \lambda\right) = t + \lambda\left(\mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\left\|\mathbf{h} - \hat{\mathbf{h}}\right\|_2^2\right] - t\right) \tag{7}$$

In the MMSE estimator, $\hat{\mathbf{h}}$ is a nonlinear function of y, but its exact form is unknown. Also, since $\mathbf{h} \in \mathcal{H}$, and in turn $\hat{\mathbf{h}} \in \mathcal{H}$, where $\mathcal{H}$ can be interpreted as a set that contains samples of $\mathbf{h}$ (and $\hat{\mathbf{h}}$) from certain unknown distribution or certain (unknown/complicated) channel models. Herein, a neural network (or graph convolution neural network) may be used to parameterize $\hat{\mathbf{h}}$ so that $\hat{\mathbf{h}} = \phi(\mathbf{h}; \boldsymbol{\theta})$, with $\boldsymbol{\theta}$ denoting the parameters of the neural network. Then the Lagrangian function can also be written as

$$\mathcal{L}\left(\hat{\mathbf{h}}, t, \lambda\right) = t + \lambda\left(\mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\|\mathbf{h} - \phi(\mathbf{y};\boldsymbol{\theta})\|_2^2\right] - t\right). \tag{8}$$

The dual function can then be written as

$$g(\lambda) = \min_{t,\boldsymbol{\theta}}\ t + \lambda\left(\mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\|\mathbf{h} - \phi(\mathbf{y};\boldsymbol{\theta})\|_2^2\right] - t\right). \tag{9}$$

It is uncertain whether or not the duality gap equals zero. However, the stationary point of $\mathcal{L}\left(\hat{\mathbf{h}}, t, \lambda\right)$ can be found via the KKT conditions by solving for the primal and dual variables alternately using gradient descent and ascent, respectively.

Then we can update the primal and dual variable using gradient descent or ascent iteratively

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \alpha_{\boldsymbol{\theta},k}\lambda_k \nabla_{\boldsymbol{\theta}_k}\mathbb{E}\left[\|\mathbf{h} - \phi(\mathbf{y};\boldsymbol{\theta}_k)\|_2^2\right] \tag{10}$$

$$t_{k+1} = t_k - \alpha_{t,k}(1 - \lambda_k) \tag{11}$$

$$\lambda_{k+1} = \left[\lambda_k + \alpha_{\lambda,k}\left(\mathbb{E}\left[\|\mathbf{h} - \phi(\mathbf{y};\boldsymbol{\theta}_{k+1})\|_2^2\right] - t_{k+1}\right)\right]_+ \tag{12}$$

where $\alpha_{\boldsymbol{\theta},k},\ \alpha_{t,k},\ \lambda_k$ are the step sizes for their respective update equations. According to [3], $\nabla_{\boldsymbol{\theta}}\mathbb{E}\left[\|\mathbf{h} - \phi(\mathbf{y};\boldsymbol{\theta}_k)\|_2^2\right]$ can be computed using finite-difference gradients or policy gradient. It was claimed in [3] that the finite-difference method can be computational expensive especially when $\boldsymbol{\theta}$ is large, hence, it was suggested that the policy gradient method should be used. Also, [4] suggested sampling the distribution in $\mathbb{E}_{\mathbf{y},\mathbf{h}}$ so that the expectation does not have to be computed in (10).

Then we can iteratively update the primal and dual variables with policy gradient to obtain the solution.

B. Policy Gradient

The policy gradient theorem is given by

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_\tau[G(\tau)] = \mathbb{E}_\tau\left[\sum_{t=0}^{T-1} G(\tau)\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}(A_t|S_t)\right],$$

where $G(\tau)$ is the overall return of the trajectory, and $\pi_{\boldsymbol{\theta}(A_t|S_t)}$ is the distribution of action that the parameterized policy makes under the observation.

At each time step, $t = 1, ..., T-1$:

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}_\tau[G(\tau)] = \mathbb{E}_\tau\left[G(\tau)\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}(A_t|S_t)\right].$$

And we can estimate the policy gradient with sample mean:

$$\widehat{\nabla_{\boldsymbol{\theta}}}\mathbb{E}_\tau[G(\tau)] = \frac{1}{|\mathcal{D}|}\sum_{\mathcal{D}} G(\tau)\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}(A_t|S_t).$$

Our goal is to minimize the mean square error, by substituting $\mathbb{E}_\tau[G(\tau)]$, $\pi_{\boldsymbol{\theta}}(A_t|S_t)$ with $\mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\|\mathbf{h} - \phi(\mathbf{y};\boldsymbol{\theta})\|_2^2\right]$, and $\pi_{\boldsymbol{\theta}}(\hat{\mathbf{h}}|\mathbf{y})$.

Thus, the estimated policy gradient for our problem is

$$\begin{aligned}\widehat{\nabla}_{\boldsymbol{\theta}}\mathbb{E}_{\mathbf{y},\mathbf{h}}\left[\|\mathbf{h} - \phi(\mathbf{y};\boldsymbol{\theta})\|_2^2\right] = \\ \frac{1}{|\mathcal{D}|}\sum_{\mathcal{D}}\left\|\mathbf{h} - \widehat{\phi}(\mathbf{y};\boldsymbol{\theta})\right\|_2^2 \nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}\left(\hat{\mathbf{h}}|\mathbf{y}\right),\end{aligned} \tag{13}$$

where $\hat{\phi}(\mathbf{y};\boldsymbol{\theta}) = \hat{\mathbf{h}}$ is the sampled output of the policy.

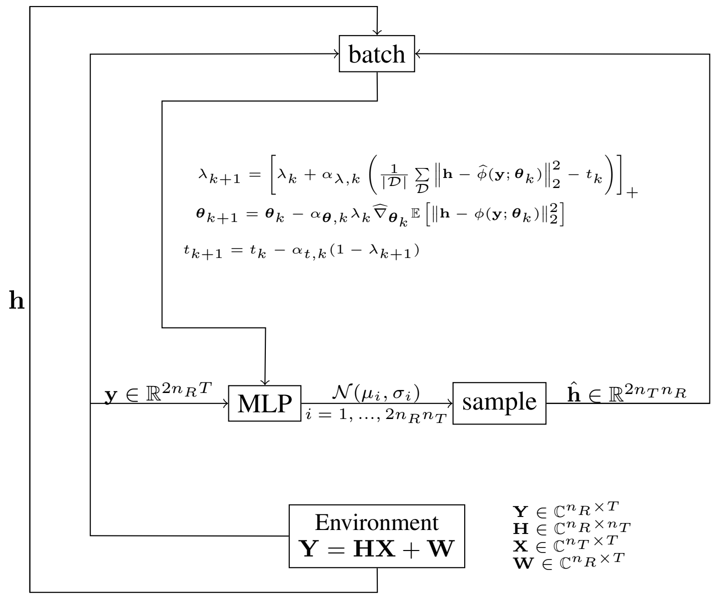


Fig. 1: Flow chart of training the parameterized policy using primal-dual method and policy gradient

C. GNN

From the content mentioned in [4], using GNN is shown to have better performance than MLP in aspects such as the number of parameters, time complexity, scalability, permutation invariance, and transference. Therefore, we replace the MLP block with a GNN and run the entire process again. According to the method in [4], the GNN architecture requires specifying the number of layers L, the number of features at each layer F F_ℓ, and the length of the filters used at each layer K K_ℓ.

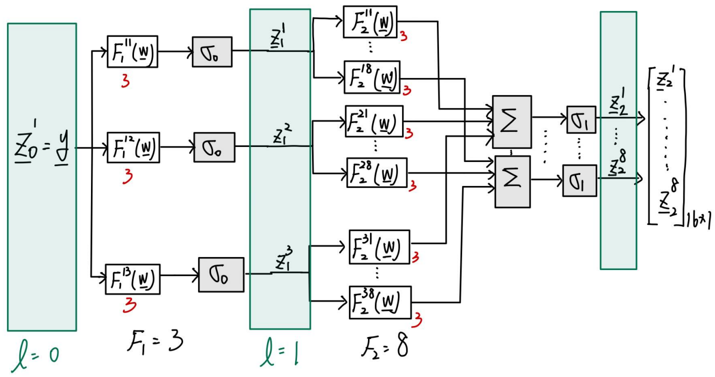
$$\begin{aligned}F_\ell^{I/O}(\mathbf{W}) &= \sum_{k=0}^{K_\ell - 1} a_{\ell/k}^{I/O}\mathbf{W}^k \\ &= a_{\ell/0}^{I/O}\mathbf{I} + a_{\ell/1}^{I/O}\mathbf{W} + a_{\ell/2}^{I/O}\mathbf{W}^2 + \ldots + a_{\ell/K_{\ell-1}}^{I/O}\mathbf{W}^{K_\ell - 1},\end{aligned} \tag{14}$$

where ℓ:layer, I:input index, O:output index

For example:

$$\begin{aligned}F_0^{1/3}(\mathbf{W}) &= \sum_{k=0}^{2} a_{0/k}^{1/3}\mathbf{W}^k \\ &= a_{0/0}^{1/3}\mathbf{I} + a_{0/1}^{1/3}\mathbf{W} + a_{0/2}^{1/3}\mathbf{W}^2\end{aligned} \tag{15}$$

Below is a schematic diagram of the GNN, where $K_\ell = 3, L = 2, F_1 = 3, F_2 = 8$, n_T = 2, n_R = 2

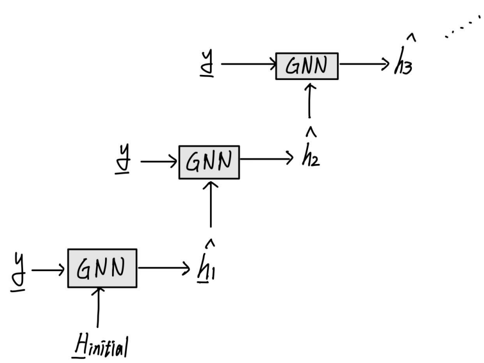


Total parameter : 3x3+3x3x8 = 81

However, there is an issue: Because in [4], the Channel matrix $\mathbf{H}$ is known, the adjacency matrix is also known, which means the Graph Shift Operator can be used for convolution. Therefore, it is not possible to use the adjacency matrix (Channel matrix, $\mathbf{H}$) as the Graph Shift Operator because the Channel matrix is the output of the REGNN, and we do not know $\mathbf{H}$ initially. Therefore, it might be impossible to use GNN methods to solve our problem.

Since we don't have $\mathbf{H}$, i guess we can use Least-Square estimator to find $\mathbf{H}_{initial}$. With this $\mathbf{H}_{initial}$, we can determine the adjacency matrix, which gives us the GSO. Then, we can use the GNN output to obtain $\hat{\mathbf{H}}$. If the performance of this $\hat{\mathbf{H}}$ is not satisfactory, we can use $\hat{\mathbf{H}}$ to determine a new adjacency matrix as the GSO and use the GNN again to obtain another $\hat{\mathbf{H}}$. We can repeat this process until the performance approaches that of the MMSE estimator.

As illustrated in the diagram below.



I guess that using this method, the estimated $\hat{\mathbf{H}}$ will get increasingly better.

V. Simulation Result

Let the loss be Normalized Mean Squared Error (NMSE):

$$NMSE = \frac{1}{N}\sum_{n=1}^{N}\frac{\|\mathbf{h}_n - \phi(\mathbf{y}_n;\boldsymbol{\theta}_k)\|_2^2}{\|\mathbf{h}_n\|_2^2}, \tag{16}$$

where N represents the size of the training or validation dataset, implying that the Normalized Mean Squared Error (NMSE) is calculated for each epoch.

Figure (2) shows the training results of different sizes of MLPs trained with SNR values of {10}, {-10}, {-10, -5, 5 10}. It displays the values of the primal and dual variables during the training process.

parameters	values
$[n_R, n_T, T]$	[4,8,8]
μ_H, σ_H	0, 1
Hidden layer size	1024 or 4×1024
Length of each trajectory	1
Batch size = $\|\mathcal{D}\|$	10,000
Number of batches	100
Trainning dataset	1,000,000
Validation dataset	2,000
Epoch	200 ∼ 1,000
Learning rate	1e-5 ∼ 1e-6

TABLE I: simulation configuration

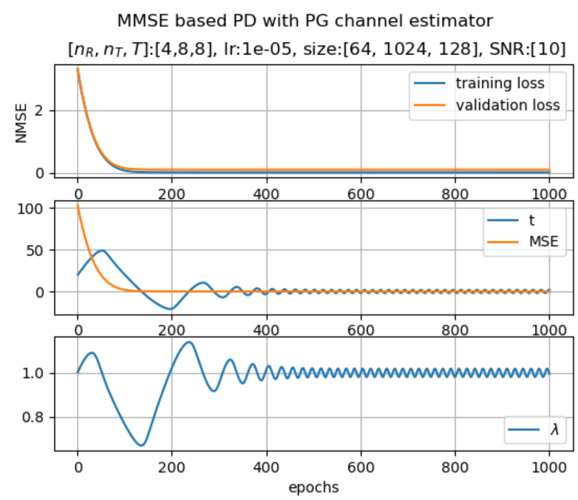


Figure (3) shows the testing results of MLPs from Figure (2) under different SNRs, compared with LS, sampled LMMSE, and ideal LMMSE estimators, where the ideal LMMSE is calculated using the ideal covariance matrix of $\mathbf{H}$: $\mathbf{C_{hh}}(\mathbf{C_{hh}}+\mathbf{C_{ww}})^{-1}\mathbf{y} = \sigma_{\mathbf{H}}^2\mathbf{I}(\sigma_{\mathbf{H}}^2\mathbf{I}+\sigma_{\mathbf{W}}^2\mathbf{I})^{-1}\mathbf{y} = \frac{\sigma_{\mathbf{H}}^2}{\sigma_{\mathbf{H}}^2+\sigma_{\mathbf{W}}^2}\mathbf{y}$ and the sample LMMSE is calculated using $\mathbf{H}$ from testing dataset: $\mathbf{C_{hh}} = \frac{1}{N}\sum_{n=1}^{N}\mathbf{h}_n\mathbf{h}_n^T$, where N is the size of testing dataset.

For the MLP trained with SNR = 10, its NMSE is close to that of the LMMSE when the SNR is high, but it performs poorly when the SNR is low. Conversely, the MLP trained with SNR = -10 performs well at low SNRs but poorly at high SNRs. However, the MLP trained with 4 different SNRs performs well at low SNRs but poorly at high SNRs, similar to the second case.

VI. Conclusion

In this work, we proposed a novel approach for MIMO channel estimation using a neural network-based minimum mean square error (MMSE) estimator, trained via deep reinforcement learning (DRL) and a primal-dual optimization method. Our method leverages the power of neural networks to model the complex relationship between the received signal and the channel state.

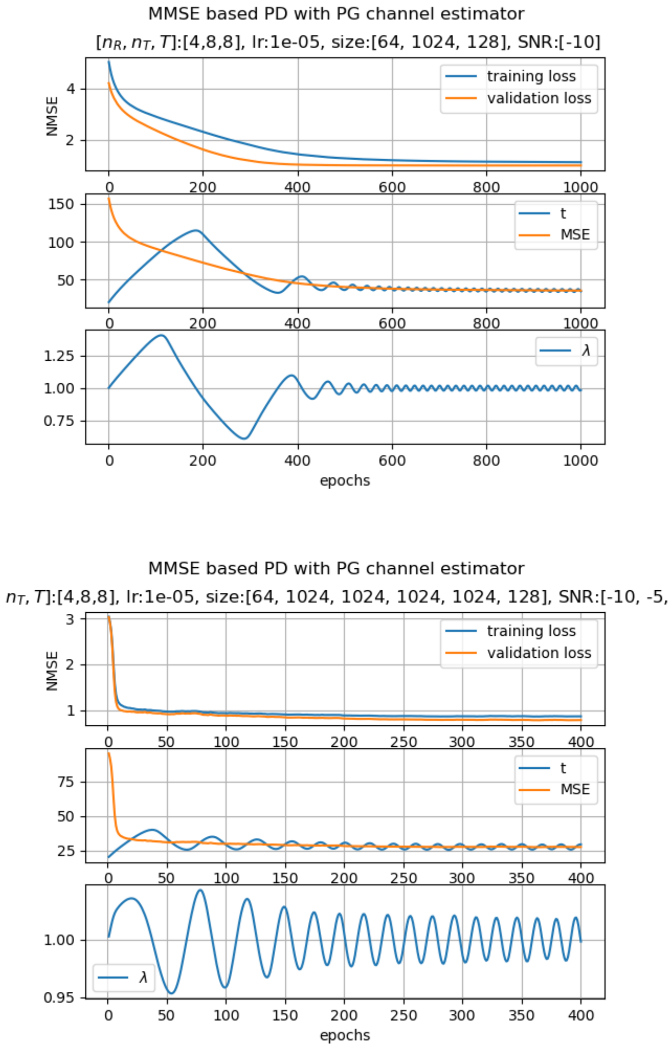


Fig. 2: (a) 1 hidden layer MLP trained with SNR = 10 (b) 1 hidden layer MLP trained with SNR = -10 (c) 4 hidden layer MLP trained with SNR = -10, -5, 5, 10

The primal-dual optimization framework facilitates efficient training by iteratively updating the primal and dual variables using gradient descent and ascent. The policy gradient method is employed to calculate the gradients, enabling the use of large neural network models without excessive computational costs.

Our simulation results demonstrate that the proposed neural network-based MMSE estimator can outperform the least squares (LS) and approach the performance of linear MMSE (LMMSE) estimators under specific signal-to-noise ratio (SNR) conditions, especially under challenging conditions with high noise levels. However, it requires a larger model size to achieve robustness in its performance.

Overall, the integration of DRL and primal-dual optimization with neural networks provides a powerful framework for tackling the MIMO channel estimation problem. Future work may explore further enhancements, such as incorporating more sophisticated neural network architectures or extending the approach to other types of communication channels and systems.

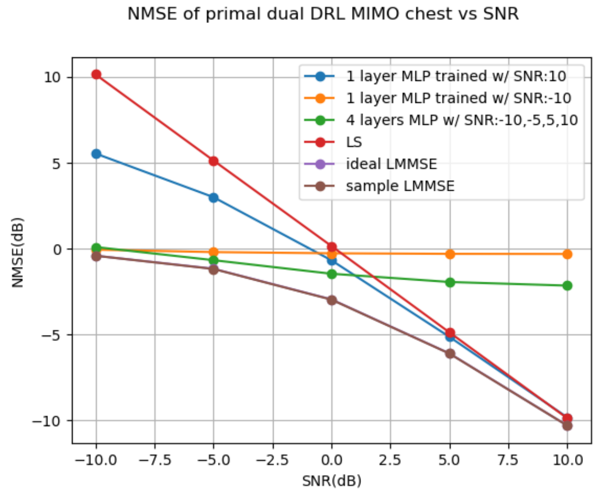


Fig. 3: performance of channel estimators under different SNRs

References

[1] C. C. Fung, and D. Ivakhnenkov, "Model-Driven Neural Network Based MIMO Channel Estimator".

[2] M. Eisen and A. Ribeiro, "Large scale wireless power allocation with graph neural networks," *Proc. of the 2019 IEEE 20th Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, pp. 1-5, 2019.

[3] M. Eisen, C. Zhang, L.F.O. Chamon, D.D. Lee and A. Ribeiro, "Learning optimal power allocations in wireless systems," *IEEE Trans. on Signal Processing*, vol. 67(10), pp. 2775-2790, May 2019.

[4] M. Eisen and A. Ribeiro, "Optimal wireless resource allocation with random edge graph neural networks," *IEEE Trans. on Signal Processing*, vol. 68, pp. 2977-2991, 2020.

[5] N. NaderiAlizadeh, M. Eisen and A. Ribeiro, "State-Augmented learnable algorithms for resource management in wireless networks," *IEEE Trans. on Signal Processing*, vol. 70, pp. 5898-5912, Dec. 2022.

[6] N. NaderiAlizadeh, M. Eisen and A. Ribeiro, "Learning resilient radio resource management policies with graph neural networks," *IEEE Trans. on Signal Processing*, volo. 71, pp. 995-1009, Mar. 2023.