ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10888576

Multi-Agent Decision Transformer for Power Control in Wireless Networks

Yiming Zhang*, Kun Yang†, Cong Shen†, and Dongning Guo*
*Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208, USA
†Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA 22904, USA

***Abstract*—This paper introduces a novel offline approach to power control in wireless networks using a multi-agent reinforcement learning (MARL) framework. We develop a multi-agent decision transformer method to optimize performance metrics including sum-rate or packet delay. In this distributed method, each agent controls an individual link and determines its power level based on its own measurements and information exchange with a few agents within a limited neighborhood.**

Numerical results demonstrate that the proposed method achieves quality of service performance comparable to centralized methods using global information, for both sum-rate maximization and traffic-driven packet delay minimization problems. As an offline learning solution, it can efficiently leverage knowledge from existing mature techniques and offers significant advantages in the safety, stability, and convergence rate over existing online methods. This work provides a promising alternative for learning-based resource management in wireless networks.

***Index Terms*—decision transformer; multi-agent reinforcement learning; offline reinforcement learning (RL); radio resource management; wireless networks.**

I. Introduction

The growing number of devices in cellular networks, driven by increasing consumer demand, has made interference management a key area of wireless networks research. Power control is a well-known interference mitigation tool used in wireless networks. Centralized optimization-based methods such as the weighted minimum mean-squared error (WMMSE) [1] and fractional programming (FP) [2] have demonstrated good quality of service (QoS) performance in power allocation. However, both algorithms require global and up-to-date channel state information (CSI), and their computational complexities scale quickly with the network size. Distributed optimization approaches like [3], [4] aim to avoid the extensive information collection required by centralized methods, but often exhibit inferior performance compared to centralized methods due to partial or imperfect CSI. We aim to develop distributed power allocation methods that achieve QoS performance comparable to centralized approaches while utilizing only locally-available information for decision-making.

In realistic wireless contexts with delayed or imperfect CSI, data-driven methods show promise over model-driven approaches like FP and WMMSE. Reinforcement learning (RL), a data-driven method with inherent sequential decision-making capabilities, has been extensively experimented with in recent years. Deep RL was applied to perform power control in [5]. More recently, RL has found applications in joint resource allocation [6]–[11].

While the aforementioned works utilize *online* RL, where policies improve through environmental interactions, *offline* RL has been introduced for radio resource management [12]. This shift addresses a major obstacle in deploying state-of-the-art RL algorithms in real-world wireless systems: the lack of performance guarantees during exploration. In the early stages, when environmental information is scarce, online RL tends to explore randomly, which could lead to poor QoS for users.

Offline RL offers two key advantages: 1) It trains policies without costly online interactions and leverages existing good policies; and 2) with datasets collected from mature policies, offline RL often enjoys a cleaner dataset and thus normally outperforms online RL in terms of training efficiency.

Previous research [12] explored offline algorithms such as batch-constrained Q-learning [13], conservative Q-learning [14], and implicit Q-learning [15]. However, all these deep RL methods build upon neural networks with multi-layer perceptron (MLP) layers, which are limited in capturing sequential information. Previous work [11] demonstrated that sequential information is crucial for power control in dense wireless networks. Therefore, we employ the decision transformer [16], leveraging the *attention* module, known for its effectiveness in extracting sequential features.

The main contributions in this paper are as follows:

- We formulate power allocation as a distributed learning problem and adopt offline RL to learn efficient control policies. This approach, which eliminates the need for real-time interactions while offering performance guarantees, represents a significant step towards practical RL implementation in wireless networks.
- We extend the decision transformer to a multi-agent setting, aligning with our distributed deployment requirements and ensuring that only locally-available information is needed post deployment.
- In addition to using sum-rate as a performance metric as in [5]–[10], we explore traffic-driven resource allocation, prioritizing average packet delay. This approach necessitates learning adaptable policies that map dynamic traffic and CSI to a broad spectrum of actions, rather than converging to a static sum-rate-maximizing solution.
- Our simulations validate the performance and scalability of the proposed solution. Results show that our method,

The work was supported by the National Science Foundation (NSF) under Grants 2003098, 2216970, 2002902, 2029978 and 2132700.

Corresponding author: yimingzhang2026@u.northwestern.edu

using only delayed and partial CSI, achieves performance (in terms of both sum-rate and packet delay) comparable to genie-aided centralized methods.

The remainder of this paper is organized as follows. We describe the system model and problem formulation in Section II. Section III introduces the MARL framework and offline training. Section IV presents the simulation setup and numerical results. Concluding remarks are given in Section V.

II. System Model and Problem Formulation

A. System model

We consider the problem of power control in a network comprising N transmitter-receiver pairs or links, modeling a mobile ad hoc network or a cellular network where each access point serves a single device. We assume that all transmitters and receivers are equipped with a single antenna. Let time be slotted with duration T and let $\mathcal{N} = \{1, 2, \ldots, N\}$ denote the set of link indices. For simplicity, we consider a single frequency band with flat fading. The channel gain from transmitter i to receiver j in time slot t is expressed as:

$$g_{i\to j}^{(t)} = \alpha_{i\to j} \left|\beta_{i\to j}^{(t)}\right|^2, \quad t = 1, 2, \ldots \tag{1}$$

where $\alpha_{i\to j} \geq 0$ accounts for the large-scale path loss, which remains constant over many time slots, and $\beta_{i\to j}$ represents a small-scale Rayleigh fading component. In simulations, we use a first-order complex Gauss-Markov process to model small-scale fading:

$$\beta_{i\to j}^{(t)} = \rho\, \beta_{i\to j}^{(t-1)} + \sqrt{1-\rho^2}\, e_{i\to j}^{(t)} \tag{2}$$

where $\left(\beta_{i\to j}^{(0)}, e_{i\to j}^{(1)}, e_{i\to j}^{(2)}, \ldots\right)$ are independent and identically distributed circularly symmetric complex Gaussian random variables with unit variance.

The power allocated to transmitter n in time slot t is denoted as $p_n^{(t)}$. Then the global power allocation of the network in time slot t is defined as $\boldsymbol{p}^{(t)} = \left(p_1^{(t)}, p_2^{(t)}, \ldots, p_N^{(t)}\right)$. We also assume additive white Gaussian noise (AWGN) with the same power spectral density σ^2 for all receivers. The spectral efficiency of link n in time slot t can be expressed as

$$\mathcal{C}_n^{(t)}(\boldsymbol{p}) = \log\left(1 + \frac{g_{n\to n}^{(t)}\ p_n^{(t)}}{\sum_{j\in\mathcal{N}, j\neq n} g_{j\to n}^{(t)}\ p_j^{(t)} + \sigma^2}\right). \tag{3}$$

B. Problem formulation

We consider two performance metrics: sum-rate and packet delay. If we use sum-rate as the metric, the dynamic power allocation problem in slot t is to maximize the weighted network sum-rate $\sum_{i\in\mathcal{N}} w_i^{(t)} \cdot C_i^{(t)}$. A central controller must solve this NP-hard problem [17] in each time slot.

Beyond the classic sum-rate maximization problem, we investigate a traffic-driven system where each link has a First-In-First-Out queue for packet transmission, aiming to minimize packet delays. At time slot t, we define $\zeta_n^{(t)}$ as the number of packet arrivals at the slot's beginning and let L denote the packet size in bits. With bandwidth W, the queue length at the slot's end is denoted as $q_n^{(t)}$ and the system's queueing dynamic is described as:

$$q_n^{(t)} = \max\left(0,\, q_n^{(t-1)} + \zeta_n^{(t)} L - \mathcal{C}_n^{(t)} WT\right). \tag{4}$$

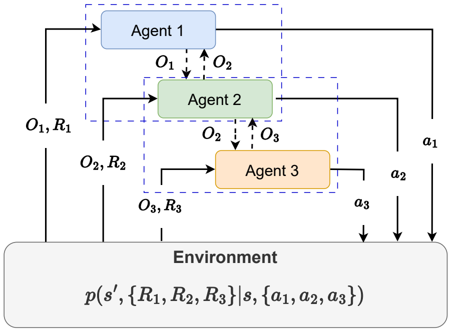

Fig. 1: Examples of MARL framework with three agents.

In this traffic-driven scenario, the environment dynamics encompass both channel state and queueing dynamics. As formulating a tractable delay minimization problem is challenging, we propose a model-free MARL method to address both sum-rate maximization and packet delay minimization problem.

III. Offline MARL

A. MARL framework

To leverage historical information and enable distributed execution for real-world deployment, we propose a distributed MARL approach. In this framework, each agent makes local power allocation decisions based on locally-available information and receives feedback, collectively forming a near-optimal global power allocation.

We define a neighborhood for each agent, allowing for observation sharing within the neighborhood. Fig. 1 illustrates an example of agent-environment interaction in our MARL framework, showing three agents with agents 1 and 2 forming one neighborhood, agents 2 and 3 forming another. Agents receive local observations from the environment and communicate with neighboring agents before making decisions. The global state evolves according to joint actions and exogenous randomness, with the environment generating rewards for each agent per state transition.

In this work, each link acts as an agent. An agent's neighborhood includes all agents that may cause significant interference to its link. Specifically, if the ratio of path loss gains $\alpha_{i\to i}/\alpha_{j\to i}$ is below a certain threshold, agent j is included in agent i's neighborhood. Let l_n denote the number of neighbors of agent n, and $\mathcal{D}(n) = \{n, \nu_{n,1}, \ldots, \nu_{n,l_n}\}$ denote agent n's neighborhood, which always includes the agent itself.

B. RL design

We assume each agent can measure its direct channel gain, total interference-plus-noise power, and compute its spectral efficiency, with all CSI delayed by one time slot. The transmitter also records the previous time slot's transmission power. The local observation $O_n^{(t)}$ for agent n at time slot t includes: link n's previous action decision $p^{(t-1)}n$, the direct gain $g_{n\to n}^{(t-1)}$, the interference-plus-noise power at receiver n $\sum_{j\in\mathcal{N}, j\neq n} g_{j\to n}^{(t-1)} p^{(t-1)}j + \sigma^2$, and the spectral efficiency $C_n^{(t-1)}\left(\boldsymbol{p}^{(t-1)}\right)$ computed from (3).

The local aggregate information of agent n at time slot t is $X_n^{(t)} = \left(O_n^{(t)}, O_{\nu_{n,1}}^{(t)}, \dots, O_{\nu_{n,l_n}}^{(t)}\right)$, incorporating neighborhood information exchange. Based on this, agent n makes a local power allocation decision $p_n^{(t)}$ from a quantized log-step power action space ranging from $P_{\min}$ to $P_{\max}$ in addition to zero power:

$$\mathcal{A} = \left\{0, P_{\min}, P_{\min}\left(\frac{P_{\max}}{P_{\min}}\right)^{\frac{1}{|\mathcal{P}|-2}}, \dots, P_{\max}\right\}. \quad (5)$$

To maximize weighted sum-rate, agent n's direct contribution is $w_n^{(t)} \cdot C_n^{(t)}\left(\boldsymbol{p}^{(t)}\right)$. We incorporate neighbors' indirect contributions to promote collaborative behavior and discourage aggressive power allocation that might lead to high interference. Consequently, the individual reward function of agent n is defined as:

$$R_n^{(t)} = \sum_{i \in \mathcal{D}(n)} w_i^{(t)} \cdot C_i^{(t)}. \quad (6)$$

In the traffic-driven system, packet arrivals $\zeta_n^{(t)}$ occur at the beginning of each time slot. We add queue length after packet arrival $q_n^{(t)} + \zeta_n^{(t)}$ to the local observation described in the sum-rate maximization problem. The action space remains as in (5). To minimize packet delay, we define the learning objective using queue lengths as surrogates, as longer queue lengths lead to longer packet delays. The utility function of agent n is $u_n^{(t)} = -q_n^{(t)}$. Similarly, neighbors' utilities are included as indirect contributions. The reward function remains as in (6), with $C_i^{(t)}$ replaced by the utility function $u_i^{(t)}$.

C. Decision transformer

The *policy* of agent n is denoted as π_n, which represents a conditional probability distribution of actions based on the agent n's historical local observations. Agent n samples its power allocation decision from this distribution. The learning goal for agent n is to find a good *policy* π_n to maximize its own future cumulative reward defined in (6). As mentioned in Section I, we adopt offline RL to learn efficient policies due to its advantages over online RL.

Recently, transformers [18] have emerged as a powerful tool for modeling multi-modal data distributions, including language and images. Building on this, the authors of [16] introduced the decision transformer, reformulating reinforcement learning as a sequence modeling task. Unlike traditional RL methods using value or policy iteration via temporal difference learning, decision transformers generate future actions by conditioning on sequences of past trajectories, including previous states, actions, and rewards. This approach leverages the transformer architecture's strength in capturing long-term dependencies and sequential patterns, enabling it to perform well in environments where temporal relationships are crucial.

We adopt a decision transformer as our policy network and extend it to multi-agent setting. For each agent n, given the episode length $\mathcal{T}$ and return $\hat{R}_n^{(t)} = \sum_{l=t}^{\mathcal{T}} R_n^{(t)}$, the transformer determines a policy that maps a historical sequences to action distribution $a_n^{(t)} \sim \pi_n(\cdot | X_n^{(t-K+1:t)}, a_n^{(t-K+1:t-1)}, \hat{R}_n^{(t-K+1:t)})$, where $X_n^{(t)}$ denotes the local aggregate information as defined earlier, superscript $(t-K+1:t)$ denotes past K time slots, and $a_n^{(t)}$ is the discrete power allocation choice from (5). This forms an auto-regressive model of order K. The decision transformer, which learns from pre-collected datasets derived from behavior policies, has been shown to converge to the performance level of these policies, as demonstrated by [19].

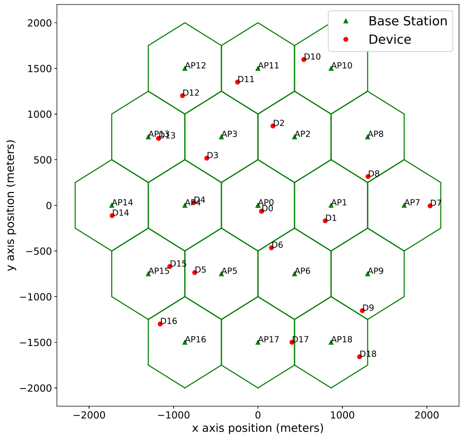

Fig. 2: Network configuration example

D. Offline training

The policies are trained using trajectories of states, actions, and return from offline datasets generated by methods with good QoS performance (e.g., WMMSE and FP). The training procedure comprises two phases: data collection and centralized training. In data collection, we execute the centralized method for multiple episodes while simulate agents in the network and collect their accessible local observations and rewards in each time slot. Joint action decisions are split for individual agents, and return is calculated for each episode to formulate the offline dataset. In centralized training, all agents share a common policy π_{DT} and a common policy network is trained using experience from all agents. Denote a sequence transition trajectory (of all agents) with starting time index t as $\tau^{(t)} = \{\hat{R}_n^{(t:t+K-1)}, X_n^{(t:t+K-1)}, a_n^{(t:t+K-1)}\}_{n \in \mathcal{N}}$. For each training step, we sample sequence transition mini-batch $\{\tau^{(l)}\}_{l \in B}$ from different episodes, update the policy by minimizing the cross-entropy loss:

$$\frac{1}{|B|N} \sum_{n=1}^{N} \sum_{l \in B} \quad (7)$$

$$-\log \pi_{DT}\left(a_n^{(l+K-1)} | X_n^{(l:l+K-1)}, \hat{R}_n^{(l:l+K-1)}, a_n^{(l:l+K-2)}\right).$$

Once the training is finished, the policy network operates using only local information for each agent during execution, ensuring decentralized deployment.

IV. Numerical Results

A. Simulation Setup

To evaluate the performance of proposed methods, we simulated different wireless network configurations with parameters

Cell radius:	500m
Path loss (LTE standard):	$128.1 + 37.6 \log_{10}(\text{distance})$ (dB)
AWGN power:	$\sigma^2 = -114$ dBm
Max transmitter power:	$P_{max} = 23$ dBm
Discretized power levels:	$\lvert\mathcal{P}\rvert = 7$
Time slot duration:	$T = 20$ ms
Bandwidth:	$W_h = 10$ MHz

TABLE I: System model parameters

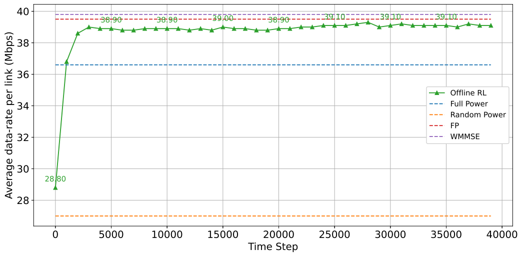

Fig. 3: Policy convergence in sum-rate maximization

listed in Table I. Due to space limitation, we present the results of a network comprising of 19 devices in 19 homogeneously deployed cells, as depicted in Fig. 2. Each cell's transmitter is located at the center, and the corresponding receiver is located randomly within the cell.

We compare the QoS performance of our multi-agent decision transformer scheduler against four benchmarks: 1) full power, 2) random uniform power, 3) FP [2] assuming real-time global CSI, and 4) WMMSE [1] assuming real-time global CSI. In the traffic-driven system, we model link traffic arrivals as discrete-time Poisson processes. The links' weights are proportional to queue lengths.

B. *Offline training and policy convergence*

We generate the offline RL dataset by operating FP for 20,000 time slots (chosen for its speed advantage over WMMSE). Offline training then proceeds based on this dataset. We use a GPT-2 Mini model (6 layers, 192 hidden size, 6 attention heads) as the foundation of our policy network. The network is trained using Adam optimizer with a learning rate of 0.005. We use a sample batch size of 64 and sequence length $K = 20$.

The model is evaluated every 1000 training time steps, with the resulting learning curves shown in Fig. 3 and Fig. 4. In both tasks, the agent achieves and maintains stable performance closely matching the benchmark used for dataset generation. Notably, offline RL in wireless network learns efficient policies within a few thousand time steps, converging much faster than online RL studied in [6]–[11]. One possible reason for the superior sample efficiency of offline RL stems from learning from experience generated by expert policies. In contrast, online RL agents interact with the environment to collect transitions and perform training based on these experiences. Many of these transitions, especially in early stages, provide limited useful information, hindering learning efficiency.

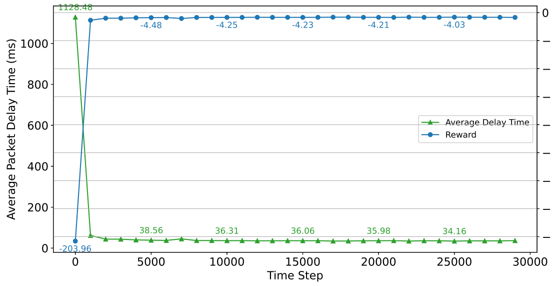

Fig. 4: Policy convergence in packet delay minimization

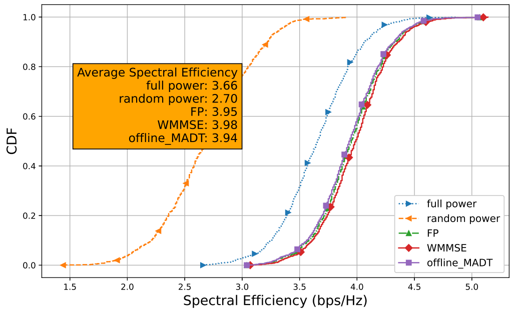

Fig. 5: CDF of spectral efficiency in 19-link network

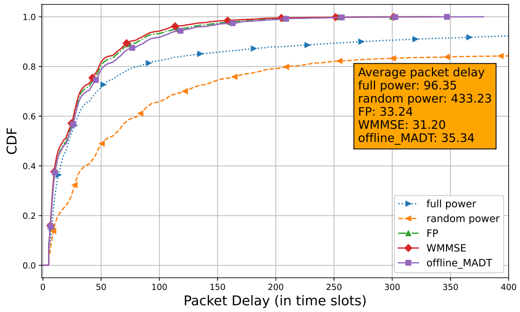

Fig. 6: CDF of packet delays in 19-link network

C. *Performance*

After centralized training, we test our learned policies using distributed execution. Provided with only local information, we plot the CDF of all links' spectral efficiency over a test episode in Fig. 5 and the CDF of packet delays for all transmitted packets in Fig. 6. Compared to the genie-aided centralized methods, our methods offer similar performance utilizing only partial and delayed information.

V. Conclusion

We have presented an offline multi-agent decision transformer approach as a solution to the power allocation problem in wireless networks. The proposed distributed method achieves sum-rate and packet delay comparable to those of two most advanced centralized optimization-based power allocation algorithms. By leveraging offline training from existing techniques, our method offers advantages in safety, stability, convergence rate over online RL methods.

References

[1] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, “An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel,” *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4331–4340, 2011.

[2] K. Shen and W. Yu, “Fractional programming for communication systems—Part I: Power control and beamforming,” *IEEE Transactions on Signal Processing*, vol. 66, no. 10, pp. 2616–2630, 2018.

[3] J. Huang, R. A. Berry, and M. L. Honig, “Distributed interference compensation for wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 5, pp. 1074–1084, 2006.

[4] S. G. Kiani, G. E. Oien, and D. Gesbert, “Maximizing multicell capacity using distributed power allocation and scheduling,” in *2007 IEEE Wireless Communications and Networking Conference*. IEEE, 2007, pp. 1690–1694.

[5] Y. S. Nasir and D. Guo, “Multi-agent deep reinforcement learning for dynamic power allocation in wireless networks,” *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2239–2250, 2019.

[6] J. Tan, Y.-C. Liang, L. Zhang, and G. Feng, “Deep reinforcement learning for joint channel selection and power control in D2D networks,” *IEEE Transactions on Wireless Communications*, vol. 20, no. 2, pp. 1363–1378, 2020.

[7] Y. S. Nasir and D. Guo, “Deep actor-critic learning for distributed power control in wireless mobile networks,” in *2020 54th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2020, pp. 398–402.

[8] A. A. Khan and R. S. Adve, “Centralized and distributed deep reinforcement learning methods for downlink sum-rate optimization,” *IEEE Transactions on Wireless Communications*, vol. 19, no. 12, pp. 8410–8426, 2020.

[9] M. K. Tefera, S. Zhang, and Z. Jin, “Deep reinforcement learning-assisted optimization for resource allocation in downlink OFDMA cooperative systems,” *Entropy*, vol. 25(3), no. 413, 2023.

[10] J. Ge, Y.-C. Liang, L. Zhang, R. Long, and S. Sun, “Deep reinforcement learning for distributed dynamic coordinated beamforming in massive MIMO cellular networks,” *IEEE Transactions on Wireless Communications*, 2023.

[11] Y. Zhang and D. Guo, “Traffic-driven spectrum and power allocation via scalable multi-agent reinforcement learning,” in *60th Annual Allerton Conference on Communication, Control, and Computing*, 2024.

[12] K. Yang, C. Shi, C. Shen, J. Yang, S.-p. Yeh, and J. J. Sydir, “Offline reinforcement learning for wireless network optimization with mixture datasets,” *IEEE Transactions on Wireless Communications*, 2024.

[13] S. Fujimoto, D. Meger, and D. Precup, “Off-policy deep reinforcement learning without exploration,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 2052–2062.

[14] A. Kumar, A. Zhou, G. Tucker, and S. Levine, “Conservative q-learning for offline reinforcement learning,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 1179–1191, 2020.

[15] I. Kostrikov, A. Nair, and S. Levine, “Offline reinforcement learning with implicit q-learning,” *arXiv preprint arXiv:2110.06169*, 2021.

[16] L. Chen, K. Lu, A. Rajeswaran, K. Lee, A. Grover, M. Laskin, P. Abbeel, A. Srinivas, and I. Mordatch, “Decision transformer: Reinforcement learning via sequence modeling,” *Advances in neural information processing systems*, vol. 34, pp. 15 084–15 097, 2021.

[17] Z.-Q. Luo and S. Zhang, “Dynamic spectrum management: Complexity and duality,” *IEEE journal of selected topics in signal processing*, vol. 2, no. 1, pp. 57–73, 2008.

[18] A. Vaswani, “Attention is all you need,” *Advances in Neural Information Processing Systems*, 2017.

[19] S. Hu, Z. Fan, C. Huang, L. Shen, Y. Zhang, Y. Wang, and D. Tao, “Q-value regularized transformer for offline reinforcement learning,” in *International Conference on Machine Learning*, 2024.