sensors

MDPI

Article

Deep Reinforcement Learning-Based Power Allocation for Minimizing Age of Information and Energy Consumption in Multi-Input Multi-Output and Non-Orthogonal Multiple Access Internet of Things Systems

Qiong Wu [1,2,*,†], Zheng Zhang [1,2,†], Hongbiao Zhu [1,2,†], Pingyi Fan [3,*], Qiang Fan [4], Huiling Zhu [5] and Jiangzhou Wang [5]

[1] School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, China; zhengzhang@stu.jiangnan.edu.cn (Z.Z.); hongbiaozhu@stu.jiangnan.edu.cn (H.Z.)
[2] State Key Laboratory of Integrated Services Networks, Xidian University, Xi’an 710071, China
[3] Department of Electronic Engineering, Beijing National Research Center for Information Science and Technology, Tsinghua University, Beijing 100084, China
[4] Qualcomm, San Jose, CA 95110, USA; qf9898@gmail.com
[5] School of Engineering, University of Kent, Canterbury CT2 7NT, UK; h.zhu@kent.ac.uk (H.Z.); j.z.wang@kent.ac.uk (J.W.)
* Correspondence: qiongwu@jiangnan.edu.cn (Q.W.); fpy@tsinghua.edu.cn (P.F.); Tel.: +86-0510-8591-0633 (Q.W.); +86-010-6279-6973 (P.F.)
† These authors contributed equally to this work.

Abstract: Multi-input multi-output and non-orthogonal multiple access (MIMO-NOMA) Internet-of-Things (IoT) systems can improve channel capacity and spectrum efficiency distinctly to support real-time applications. Age of information (AoI) plays a crucial role in real-time applications as it determines the timeliness of the extracted information. In MIMO-NOMA IoT systems, the base station (BS) determines the sample collection commands and allocates the transmit power for each IoT device. Each device determines whether to sample data according to the sample collection commands and adopts the allocated power to transmit the sampled data to the BS over the MIMO-NOMA channel. Afterwards, the BS employs the successive interference cancellation (SIC) technique to decode the signal of the data transmitted by each device. The sample collection commands and power allocation may affect the AoI and energy consumption of the system. Optimizing the sample collection commands and power allocation is essential for minimizing both AoI and energy consumption in MIMO-NOMA IoT systems. In this paper, we propose the optimal power allocation to achieve it based on deep reinforcement learning (DRL). Simulations have demonstrated that the optimal power allocation effectively achieves lower AoI and energy consumption compared to other algorithms. Overall, the reward is reduced by 6.44% and 11.78% compared the to GA algorithm and random algorithm, respectively.

Keywords: deep reinforcement learning; age of information; MIMO-NOMA; Internet of Things

check for updates

Citation: Wu, Q.; Zhang, Z.; Zhu, H.; Fan, P.; Fan, Q.; Zhu, H.; Wang, J. Deep Reinforcement Learning-Based Power Allocation for Minimizing Age of Information and Energy Consumption in Multi-Input Multi-Output and Non-Orthogonal Multiple Access Internet of Things Systems. *Sensors* **2023**, *23*, 9687. https://doi.org/10.3390/s23249687

Academic Editor: Stefano Savazzi

Received: 6 November 2023
Revised: 3 December 2023
Accepted: 5 December 2023
Published: 7 December 2023

1. Introduction

With the development of the Internet of Things (IoT), the base station (BS) can support the real-time applications such as disaster management, information recommendation, vehicle network, smart city, connected health and smart manufacturing by collecting the data sampled by IoT devices [1,2]. However, the amount of sampled data is enormous and the number of IoT devices is usually high; thus, the realization of these IoT applications requires a large bandwidth spectrum [3]. The multi-input multi-output and non-orthogonal multiple access (MIMO-NOMA) IoT can transmit data through the MIMO-NOMA channel to solve these problems, wherein multiple antennas are deployed at the BS to improve the channel capacity and multiple IoT devices access the common bandwidth simultaneously to improve the spectrum efficiency.

The BS collects data during discrete slots in the MIMO-NOMA IoT system. In each slot, a BS first determines the sample collection commands and allocates the transmit power for each IoT device and then sends the corresponding sample collection commands and transmission power to each IoT device. Afterwards, each IoT device determines whether to sample data from the physical world according to their sample collection commands. Then, each IoT device adopts its allocated power to transmit the sampled data to the BS over the MIMO-NOMA channel. In the transmission process, multiple IoT devices transmit the signals of the data by using the same spectrum, and therefore interference exists between different IoT devices. To eliminate the interference, the BS adopts the successive interference cancellation (SIC) technique to decode the signals from each device [4]. Specifically, the BS sorts the power of all received signals in descending order and decodes the signal with the highest received power by considering other signals as interferences. Then, the BS removes the decoded signal from the received signals and resorts the received signals to decode the next signal. The process is repeated until all signals are decoded.

The age of information (AoI) is a metric to measure the freshness of the data, which is defined as the time from the data sampling to the time when the sampled data are received [5]. In the MIMO-NOMA IoT system, the BS needs to receive data, i.e., decode the signals of the data, in a timely manner after they are sampled to provide the real-time applications; thus, a low AoI is critical in MIMO-NOMA IoT systems [6]. Furthermore, the IoT devices are energy-limited. Thus, the MIMO-NOMA IoT system should also keep its energy consumption low to prolong the working time of the IoT devices [7]. Hence, the AoI and energy consumption are two important performance metrics of the MIMO-NOMA IoT system [8]. The sample collection commands and power allocation may affect the AoI and energy consumption of the system. Specifically, for the sample collection commands, if the BS selects more IoT devices to sample, the system will consume more energy because more IoT devices consume energy to sample data. However, if the BS selects less IoT devices to sample, the data transmitted from the unselected IoT devices become obsolete, which may increase the AoI of the system. Hence, the sample collection commands affect both the AoI and energy consumption of the MIMO-NOMA IoT system. For the power allocation, if an IoT device transmits with high power, the signal transmitted by the IoT device will be decoded wherein a significant amount of signals with lower power act upon the interferences in the SIC process, which may lead to a low signal-to-interference-plus-noise ratio (SINR). Otherwise, if an IoT device transmits data with low power, the SINR may also be deteriorated due to the low transmission power. The low SINR causes a low transmission rate, which may cause a long transmission delay and a high AoI of the MIMO-NOMA IoT system. Hence, the power allocation affects the AoI of the MIMO-NOMA IoT system. Moreover, the power allocation affects the energy consumption directly. Thus, the transmission power affects both the AoI and energy consumption of the MIMO-NOMA IoT system. As mentioned above, it is critical to determine the optimal policy including sample collection commands and power allocation to minimize the AoI and energy consumption of the MIMO-NOMA IoT system. To the best of our knowledge, there is no work to minimize the AoI in the MIMO-NOMA IoT system, which motivates us to conduct this work. In the MIMO-NOMA IoT system, the allocation of transmission powers has a direct impact on the transmission rate during the SIC process. Additionally, the MIMO-NOMA channel is inherently affected by stochastic noise. Model-based algorithms struggle to construct an accurate model to describe this process, which causes the traditional model-based algorithms unsuitable to solve the problem. Deep reinforcement learning (DRL) is a type of model-free-based method that enables an agent to learn how to make sequential decisions in a complex environment to achieve a specific goal. DRL can learn the near-optimal policy by learning from the interaction between action and the environment (i.e., dynamic stochastic MIMO-NOMA IoT system) [9]. There are some existing studies on DRL-based optimization frameworks in similar systems. In [10], Zhao et al. formalized the joint optimization problem of video frame resolution selection, computation offloading and resource allocation strategy, and proposed a hierarchical reward function based on the DRL algorithm that minimizes energy consumption, maximizes quality of experience (QoE) delay and analyzes the accuracy in the IoT system. In [11], Chen et al. considered a marginalized IoT system and studied the joint caching and computing service deployment (JCCSP) problem for IoT applications driven by perceptual data. An improved method based on twin-delayed (TD) deep deterministic policy gradient (DDPG) was proposed, which achieved significant convergence performance compared to benchmarks. In general, the DRL algorithm is used to solve problems with either continuous or discrete action spaces separately. However, we focus on simplifying the joint optimization problem when the space of the sample collection commands is discrete while the space of the transmission power is continuous, to make it applicable to DDPG. We achieve this goal by establishing the relationship between sample collection commands and transmission power, and then propose a DRL-based power allocation to minimize the AoI and energy consumption of the MIMO-NOMA IoT system (The source code has been released at: https://github.com/qiongwu86/MIMO-NOMA_AoI_GA.git (7 March 2023)). The main contributions are summarized as follows:

(1) We formulated the joint optimization problem to minimize the AoI and energy consumption of the MIMO-NOMA IoT system by determining the sample selection and power allocation. Specifically, we constructed an MIMO-NOMA channel model and an AoI model to find the relationship between transmission rate and AoI of each device under the SIC mode. Additionally, we constructed an energy consumption model. Then, the joint optimization problem was formulated based on the constructed models.

(2) Then, we simplified the formulated optimization problem to make it suitable for DRL algorithms. In the formulated optimization problem, the sample selection is discrete and power allocation is continuous, which cannot be solved by the traditional DRL method and results in a challenge for optimization. We substituted the energy model and AoI model by the formulated optimization problem, merged the homogeneous terms containing sample selection and simplified the formulated problem to make it suitable to be solved by the traditional continuous-control DRL algorithm.

(3) To solve the formulated optimization problem, we first designed a DRL framework which included the state, action and reward function, and then adopted the DDPG algorithm to obtain the optimal power allocation to minimize the AoI and energy consumption of the MIMO-NOMA IoT system.

(4) Extensive simulations were carried out to demonstrate that the DDPG algorithm successfully optimizes both the AoI and energy consumption compared with other baseline algorithms.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 introduces the system model and formulates the optimization problem. Section 4 simplifies the formulated optimization problem and presents the near-optimal solution by DRL. We carry out some simulation to demonstrate the effectiveness of our proposed DRL method in Section 5, and conclude this paper in Section 6.

2. Related Work

In this section, we first review the studies about the AoI in the IoT system, and then survey the state of the arts on the MIMO-NOMA IoT system.

2.1. AoI in IoT

In [12], Grybosi et al. proposed the SIC-aided age-independent random access (AIRA-SIC) scheme (i.e., a slotted ALOHA fashion) for the IoT system, wherein the receiver operates SIC to reconstruct the collisions of various devices. In [13], Wang et al. focused on the problem that minimizes the weighted sum of AoI cost and energy consumption in the IoT systems by adjusting the sample policy, and proposed a distributed DRL algorithm based on the local observation of each device. In [14], Elmagid et al. aimed to minimize the AoI at the BS and the energy consumption of the generate status for the IoT devices, formulated an optimization problem based on the Markov decision process (MDP) and then proved the monotonicity property of the value function associated with the MDP. In [15], Li et al. designed a resource block (RB) allocation, modulation-selecting and coding-selecting scheme for each IoT device based on its channel condition to minimize the long-term AoI of the IoT system. In [16], Hatami et al. employed the reinforcement learning to minimize the average AoI for users in an IoT system consisting of users, energy harvesting sensors and a cache-enabled edge node. In [17], Sun et al. aimed to minimize the weighted sum of the expected average AoI of all IoT devices, propulsion energy of unmanned aerial vehicle (UAV) and transmission energy of IoT devices by determining the UAV flight speed, UAV placement and channel resource allocation in the UAV-assisted IoT system. In [18], Hu et al. considered an IoT system wherein the UAVs take off from a data center to deliver energy and collect data from sensor nodes, and then fly back to the data center. They minimized the AoI of the collected data by dynamic programming (DP) and ant colony (AC) heuristic algorithms. In [19], Emara et al. developed a spatio-temporal framework to evaluate the peak AoI (PAoI) of the IoT system, and compared the PAoI under the time-triggered traffic with event-triggered traffic. In [20], Lyu et al. considered a marine IoT scenario, wherein the AoI is utilized to represent the impact of the packet loss and transmission delay. They investigated the relationship between AoI and state estimation error, and minimized the state estimation error by the decomposition method. In [21], Wang et al. investigated the impact of AoI on the system cost which consists of control cost and communication energy consumption of the industrial-Internet-of-Things (IIoT) system. They proved that the upper bound of cost is affected by the AoI. In [22], Hao et al. maximized the sum of the energy efficiency of the IoT devices under the constraints of AoI by optimizing the transmission power and channel allocation in a cognitive radio-based IoT system. However, none of these works have taken the MIMO-NOMA channel into account.

2.2. MIMO-NOMA IoT System

In [23], Yilmaz et al. proposed a user selection algorithm for the MIMO-NOMA IoT system to improve the sum data rate, and adopted the physical layer network coding (PNC) to improve the spectral efficiency. In [24], Shi et al. considered the downlink of the MIMO-NOMA IoT networks and studied the outage probability and goodput of the system with the Kronecker model. In [25], Wang et al. proposed that the resource allocation problem consists of the optimal beamforming strategy and power allocation in the MIMO-NOMA IoT system, wherein the beamforming optimization is solved by the zero-forcing method, and after that the power allocation is solved by the convex functions. In [26], Han et al. proposed a novel millimeter wave (mmWave) positioning MIMO-NOMA IoT system and proposed the position error bound (PEB) as a novel performance evaluation metric. In [27], Zhang et al. considered the massive MIMO and NOMA to study the performance of the IoT system, and calculated the closed-form function for spectral and energy efficiencies. In [28], Chinnadurai et al. considered the heterogeneous cellular network and formulated a problem to maximize the energy efficiency of the MIMO-NOMA IoT system, wherein the non-convex problem was solved based on the branch and reduced-bound (BRB) approach. In [29], Gao et al. considered the mmWave massive MIMO and NOMA IoT system to maximize the weighted sum transmission rate by optimizing the power allocation, and then solved the problem by the convex method. In [30], Feng et al. considered an UAV-aided MIMO-NOMA IoT system and regarded an UAV as the BS. They formulated the problem to maximize the sum transmission rate of the downlink by optimizing the placement of UAVs, beam pattern and transmission power, and then solved the problem by convex methods. In [31], Ding et al. designed a novel MIMO-NOMA system consisting of two different users, wherein user one should be served with strict quality-of-service (QoS) requirement, and user two accesses the channel by the non-orthogonal way opportunistically; thus, the requirement that small packets of user one in the IoT system should be transmitted in time can be met. In [32], Bulut et al. proposed the water cycle algorithm (WCA) based on the energy allocation method for MIMO-NOMA IoT systems. Their simulation results demonstrated that the proposed method performs better than empirical search algorithm (ESA) and genetic algorithm (GA). In [33], Ullah et al. proposed a power allocation algorithm based on DDPG to maximize energy efficiency in MIMO-NOMA next-generation Internet-of-Things (NG-IoT) networks. Their simulation results demonstrated that the proposed method achieved better performance compared with random algorithms and greedy algorithms. However, these works have not considered the AoI of the MIMO-NOMA IoT system.

As mentioned above, there is no work considering the joint optimization problem of age of information and energy in the MIMO-NOMA IoT system, which motivates us to conduct this work. The comparison of the related works is shown in Table 1.

Table 1. The comparison between related works.

Related Work	MIMO-NOMA	AoI Minimization	Energy Optimization
[12,15,17–19]	×	✓	×
[13,14,16,22]	×	✓	✓
[25,28,29,32,33]	✓	×	✓
[23,24,26,27]	✓	×	×

3. System Model And Problem Formulation

3.1. Scenario Description

The network scenario is illustrated in Figure 1. We consider a MIMO-NOMA IoT system consisting of a BS with K antennas and a set $\mathcal{M} = \{1, \ldots, m, \ldots, M\}$ of the single-antenna IoT devices. Here, each IoT device is embedded with a sensor and a transmitter. The time duration is divided into T slots, each of which is τ. The set of slots is denoted as $\mathcal{T} = \{1, \ldots, t, \ldots, T\}$. At the beginning of each slot t, the BS determines the policy (including the sample collection commands of each device m, denoted as $s_{m,t}$, and the transmission power of each device m, denoted as $p_{m,t}$) and then sends $s_{m,t}$ and $p_{m,t}$ to each device m. If $s_{m,t} = 1$, device m will sample data in slot t, and transmit the data to the BS with transmission power $p_{m,t}$ over the MIMO-NOMA channel. This action reduces the AoI, but also incurs a cost in terms of energy consumption. Otherwise, it does not sample data in slot t; therefore, it does not consume energy for sampling and transmission, while increasing the AoI due to a lack of updates. The key notations are listed in Table 2. Next, we will construct the MIMO-NOMA channel model.

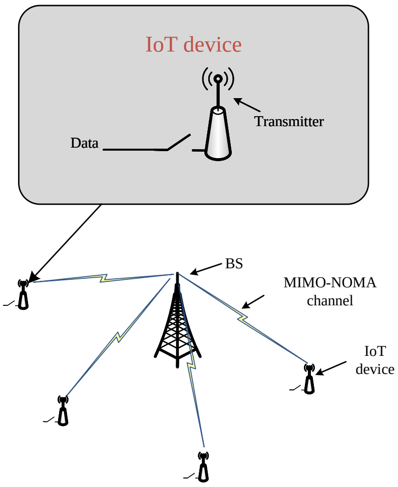

Figure 1. MIMO-NOMA IoT system.

Table 2. The summary of the notations.

Notation	Description	Notation	Description
B	Population size of genetic algorithm.	C_s	The energy consumption to sample fresh information and generate upload packet.
$c_{m,t}$	Complex data symbol with 1 as variance.	d_m	The communication distance between device m and BS.
E	Number of episodes.	F_c/F_m	Probability of offspring in genetic algorithm for crossover/mutation.
G_P/U_P	Complexity of the primary networks for computing gradients/updating parameters.	$\boldsymbol{h}_m(t)$	The channel vector between device m and BS in slot t.
i	Index of transition tuples in mini-batch.	I	The number of transition tuples in a mini-batch.
$\mathcal{I}_m$	The set of devices in which the received power is weaker than device m.	$J(\mu)$	The long-term discounted reward under policy μ.
K	The number of antennas equipped in BS.	$l_{m,t}$	The transmission delay of device m in slot t.
L	Loss function.	$\boldsymbol{n}(t)$	Additive white Gaussian noise.
N_{GA}	Evolution times of genetic algorithm	$m/M/\mathcal{M}$	Index/number/set of devices.
$\boldsymbol{o}_t/\boldsymbol{o}_{m,t}$	State in slot t of all devices/device m	$\boldsymbol{p}_t/p_{m,t}$	Transmission power of all devices/device m.
$P_{m,t}$	Maximum transmission power device m.	$Q(\boldsymbol{o}_t,\boldsymbol{p}_t)$	Action-value function under $\boldsymbol{o}_t$ and $\boldsymbol{p}_t$.
$Q(\boldsymbol{o}_t,\boldsymbol{p}_t)$	Action-value function under $\boldsymbol{o}_t$ and $\boldsymbol{p}_t$.	Q	Packet size.
r_t	Reward function.	$\boldsymbol{s}_t/s_{m,t}$	Indicator of sample or not for all devices/device m.
$\boldsymbol{s}_t/s_{m,t}$	Indicator of sample or not for all devices/device m.	S_d	Complexity of calculating sample decisions based on power allocation.
S_d	Complexity of calculating sample decisions based on power allocation.	$t/\mathcal{T}$	Index/set of slot.
$\mathcal{U}$	The set of undecoded received power of BS.	$\boldsymbol{u}_t/u_{m,t}$	Indicator of transmission success for all devices/device m.
W	Bandwidth of system.	α_a/α_c	Learning rate of actor network/critic network.
β	Discounting factor.	γ_a,γ_c	Weighted factors of reward function.
$\Gamma_{m,t}$	Received power of BS for device m in slot t.	Δ_t	Exploration noise.
$\varepsilon_{m,t}$	The energy consumed by device m in slot t.	$\bar{\varepsilon}$	The average sum energy consumption in slot t.
ζ/ζ'	Parameters of critic-network/target critic-network.	$\theta/\theta'/\theta^*$	Parameters of actor-network/target actor-network/optimal policy.
κ	The constant for the update of target networks.	μ_θ	Policy approximated by actor-network with θ.
$\pi_{m,t}$	Transmission rate of device m in slot t.	ρ_m	Normalized channel correlation coefficient.
σ_R^2	Variance of received signal's noise.	$\phi_{m,t}/\Phi_{m,t}$	AoI of device m in slot t on device/BS.
$\overline{\Phi}$	The average sum AoI.		

3.2. MIMO-NOMA Channel Model

Let $c_{m,t}$ be the data symbol of device m in slot t with 1 as variance; thus, the signal of the data transmitted by device m is $\sqrt{p_{m,t}}c_{m,t}$. Let $\boldsymbol{h}_m(t) \in \mathbb{C}^{K\times 1}$ be the channel power gain between the BS and device m in slot t; thus, the corresponding signal received by the BS is $\boldsymbol{h}_m(t)\sqrt{p_{m,t}}c_{m,t}$. Note that $c_{m,t}$ is unknown for the BS, so that it is difficult for the BS to calculate the received signal. Hence, the BS needs to adopt the SIC technology to decode the received signal transmitted by each device, which is expressed as

$$\begin{gathered} \boldsymbol{y}(t) = \sum_{m\in\mathcal{M}} \boldsymbol{h}_m(t)\sqrt{p_{m,t}}c_{m,t} + \boldsymbol{n}(t) \\ p_{m,t} \in [0, P_{m,max}], \forall m \in \mathcal{M}, \forall t \in \mathcal{T} \end{gathered} , \tag{1}$$

where $\boldsymbol{n}(t) \in \mathbb{C}^{K\times 1}$ is the complex additive white Gaussian noise (AWGN) with variance σ_R^2 and $P_{m,max}$ is the maximum transmission power of device m.

In [34,35], the authors adopted $\boldsymbol{h}_m(t)$ estimated by the deep neural network or minimum mean square error method in the SIC process and demonstrated its efficiency. In addition, the BS also knows $p_{m,t}$; thus, the BS can calculate the power of the received signal transmitted by device m as

$$\Gamma_{m,t} = p_{m,t}||\boldsymbol{h}_m(t)||^2. \tag{2}$$

Then, the BS decodes the received signal transmitted by each device sequentially. For one iteration, the BS decodes the signal with the highest received power from $\boldsymbol{y}(t)$ while considering the other signals as interference, and then removes the decoded signal from $\boldsymbol{y}(t)$ and starts the next iteration until all signals are decoded.

For instance, in an iteration, the received power of the signal transmitted by device m is the highest among the signals without being decoded. Denote $\mathcal{I}_m = \{k \in \mathcal{M} \mid \Gamma_{k,t} < \Gamma_{m,t}\}$ as the set of devices whose signals' received powers is less than device m. Thus, the signal transmitted by each device $k \in \mathcal{I}_m$ is deemed as the interference. In this case, $\boldsymbol{y}(t)$ is rewritten as

$$\boldsymbol{y}(t) = \boldsymbol{h}_m(t)\sqrt{p_{m,t}}c_{m,t} + \sum_{k\in\mathcal{I}_m} \boldsymbol{h}_k(t)\sqrt{p_{k,t}}c_{k,t} + \boldsymbol{n}(t), \tag{3}$$

where $\sum_{k\in\mathcal{I}_m} \boldsymbol{h}_k(t)\sqrt{p_{k,t}}c_{k,t}$ indicates the interference; thus, the signal-to-interference-plus-noise ratio (SINR) of device m is calculated as

$$\begin{aligned}\gamma_{m,t} &= \frac{p_{m,t}||\boldsymbol{h}_m(t)||^2}{\sum\limits_{k\in\mathcal{I}_m} p_{k,t}||\boldsymbol{h}_k(t)||^2 + \sigma_R^2} \\ &= \frac{\Gamma_{m,t}}{\sum\limits_{k\in\mathcal{I}_m} \Gamma_{k,t} + \sigma_R^2}\end{aligned}. \tag{4}$$

The transmission rate of device m in slot t can be derived according to Shannon capacity formula, i.e.,

$$\pi_{m,t} = W\log_2(1+\gamma_{m,t}), \tag{5}$$

where W is the bandwidth of the MIMO-NOMA channel.

3.3. AoI Model

Denote $\phi_{m,t}$ as the AoI at device m in slot t, which can be calculated as

$$\phi_{m,t} = \begin{cases} 0, & s_{m,t} = 1 \\ \phi_{m,t-1} + \tau, & otherwise \end{cases}. \tag{6}$$

According to Equation (6), at the beginning of slot t, if device m samples data, i.e., $s_{m,t} = 1$, $\phi_{m,t}$ will be reset to 0. Otherwise, $\phi_{m,t}$ will be increased by τ.

Device m will transmit data with transmission power $p_{m,t}$ after sampling data. If the data volume transmitted within a slot is larger than the packet size Q, i.e., $\pi_{m,t} \cdot \tau \geq Q$, device m will transmit the data successfully; otherwise, the transmission fails. Denoting $u_{m,t} = 1$ as a successful transmission by device m in slot t and $u_{m,t} = 0$ as an unsuccessful transmission, we have

$$u_{m,t} = \begin{cases} 1, & \pi_{m,t} \cdot \tau \geq Q \\ 0, & otherwise \end{cases}. \tag{7}$$

According to [36], if a transmission from device m is successful, the AoI at the BS equals the aggregation of AoI at device m and the transmission delay. Otherwise, the AoI at the BS is increased by a slot; therefore, we have

$$\Phi_{m,t} = \begin{cases} \phi_{m,t} + l_{m,t}, & u_{m,t} = 1 \\ \Phi_{m,t-1} + \tau, & otherwise \end{cases}, \tag{8}$$

where $l_{m,t}$ is the transmission delay of device m in slot t, which is calculated as

$$l_{m,t} = \frac{Q}{\pi_{m,t}}. \tag{9}$$

The AoI of the MIMO-NOMA IoT system is measured by averaging the AoI of all devices at the BS, i.e.,

$$\overline{\Phi} = \frac{1}{T} \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} \Phi_{m,t}. \tag{10}$$

3.4. Energy Consumption Model

Since each device consumes energy in data sampling and transmission, the energy consumption of device m in slot t can be calculated as

$$\varepsilon_{m,t} = s_{m,t} C_s + p_{m,t} l_{m,t}, \tag{11}$$

where C_s is the energy consumption for data sampling [13], and $p_{m,t} l_{m,t}$ is the energy consumption for transmission.

The BS has a stable power supply; hence, the energy consumption of the BS is sufficient and thus it is not taken into account in the system. Hence, the energy consumption of the MIMO-NOMA IoT system is measured by averaging the energy consumption of all devices, i.e.,

$$\bar{\varepsilon} = \frac{1}{T} \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} \varepsilon_{m,t}. \tag{12}$$

3.5. Problem Formulation

In this work, our target is to minimize the AoI and energy consumption of the MIMO-NOMA IoT system, which is impacted by $p_{m,t}$ and $s_{m,t}$. Therefore, the optimization problem is formulated as

$$\min_{\boldsymbol{s}_t, \boldsymbol{p}_t} \left[\gamma_a \overline{\Phi} + \gamma_e \bar{\varepsilon} \right] \tag{13}$$

$$s.t. \quad p_{m,t} \in [0, P_{m,max}], \forall m \in \mathcal{M}, \forall t \in \mathcal{T}, \tag{13a}$$

$$s_{m,t} \in \{0, 1\}, \forall m \in \mathcal{M}, \forall t \in \mathcal{T}, \tag{13b}$$

where $\boldsymbol{s}_t = \{s_{1,t}, \dots, s_{m,t}, \dots, s_{M,t}\}$ and $\boldsymbol{p}_t = \{p_{1,t}, \dots, p_{m,t}, \dots, p_{M,t}\}$, γ_a and γ_e are the non-negative weighted factors. Next, we will present a solution to the problem based on DRL.

4. DRL Method for Optimization of Power Allocation

In this section, we solve the optimization problem based on the DRL. First, we design the DRL framework including the state, action and reward function, wherein the relationship between the sample collection commands and transmission power is derived to facilitate the DRL algorithm in solving the problem. Then, we obtain a near-optimal power allocation based on the DRL algorithm.

4.1. DRL Framework

The DRL framework consists of three significant elements: state, action and reward function. For each slot, the agent observes the current state and takes the current action according to policy μ, where policy μ yields the action based on the state. Then, the agent calculates its corresponding reward under the current state and action according to the reward function, while the current state in the environment transits to the next state. Next, we will design agent, state action and reward function based on DRL [37], respectively.

- **Agent:** In each slot, the BS determines the transmission power and sample collection commands of each device based on its observation; thus, we consider the BS as the agent.
- **State:** In the system model, the state $\boldsymbol{o}_t$ observed by the BS in slot t is defined as

$$\boldsymbol{o}_t = [\boldsymbol{o}_{1,t}, \ldots, \boldsymbol{o}_{m,t}, \ldots, \boldsymbol{o}_{M,t}], \tag{14}$$

where $\boldsymbol{o}_{m,t}$ represents the observation of device m, which is designed as

$$\boldsymbol{o}_{m,t} = [u_{m,t-1}, \gamma_{m,t-1}, \Phi_{m,t-1}]. \tag{15}$$

Here, $u_{m,t-1}$, $\gamma_{m,t-1}$ and $\Phi_{m,t-1}$ can be calculated by the BS from the historical data in slot $t-1$.
- **Action:** According to the problem formulated in Equation (13), the action in slot t is set as

$$\boldsymbol{a}_t = [\boldsymbol{s}_t, \boldsymbol{p}_t]. \tag{16}$$

The two traditional DRL algorithms, namely DDPG and Deep Q-Learning (DQN), are suitable for continuous and discrete action space, respectively. However, $s_{m,t} \in \{0, 1\}$ and $p_{m,t} \in [0, P_{m,max}]$ in Equation (16); thus, the space of $\boldsymbol{s}_t$ is discrete while the space of $\boldsymbol{p}_t$ is continuous. Hence, the optimization problem can neither be solved by DQN nor DDPG. Next, we will investigate the relationship between $p_{m,t}$ and $s_{m,t}$ to handle this dilemma.

Substituting Equations (10) and (12) by Equation (13), the optimization objective is rewritten as Equation (17a). Then, substituting Equations (8) and (11) byEquation (17a), we can obtain Equation (17b), where $\phi_{m,t}$ is denoted as $\phi_{m,t}(s_{m,t})$ to indicate that it is the function of $s_{m,t}$. Then, by reorganizing Equation (17b), we have Equation (17c). The first term of Equation (17c) is related with $s_{m,t}$; next, we rewrite the first term of Equation (17c) as Equation (18) to investigate the relationship between $s_{m,t}$ and $p_{m,t}$. Substituting Equation (6) by Equation (18), we have Equation (18a). Then, by merging the homogeneous terms containing $s_{m,t}$ and γ_a in Equation (18a), respectively, we have Equation (18b). Let $C_{m,t,1} = \gamma_e C_s - \gamma_a u_{m,t}(\phi_{m,t-1} + \tau)$ and $C_{m,t,2} = \gamma_a[u_{m,t}(\phi_{m,t-1} + \tau) + (1 - u_{m,t})(\Phi_{m,t-1} + \tau) + u_{m,t} l_{m,t}] + \gamma_e p_{m,t} l_{m,t}$; thus, Equation (18b) is rewritten as Equation (18c), where $C_{m,t,1}$ is the coefficient for homogeneous terms containing $s_{m,t}$ in Equation (18b), and $C_{m,t,2}$ contains all terms without $s_{m,t}$ in Equation (18b).

$$\gamma_a \overline{\Phi} + \gamma_e \bar{\varepsilon} \tag{17}$$

$$= \frac{1}{T} \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} \left[\gamma_a \Phi_{m,t} + \gamma_e \varepsilon_{m,t} \right] \tag{17a}$$

$$= \frac{1}{T} \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} \left[\gamma_a \left[(1 - u_{m,t})(\Phi_{m,t-1} + \tau) + u_{m,t}(\phi_{m,t}(s_{m,t}) + l_{m,t}) \right] + \gamma_e (s_{m,t} C_s + p_{m,t} l_{m,t}) \right] \tag{17b}$$

$$= \frac{1}{T} \sum_{t \in \mathcal{T}} \sum_{m \in \mathcal{M}} \left[[\gamma_a u_{m,t} \phi_{m,t}(s_{m,t}) + \gamma_e s_{m,t} C_s] + \gamma_a [(1 - u_{m,t})(\Phi_{m,t-1} + \tau) + u_{m,t} l_{m,t}] + \gamma_e p_{m,t} l_{m,t} \right] \tag{17c}$$

$$\gamma_a \Phi_{m,t}(s_{m,t}, p_{m,t}) + \gamma_e \varepsilon_{m,t}(s_{m,t}, p_{m,t}) \tag{18}$$

$$= \gamma_a u_{m,t}(1 - s_{m,t})(\phi_{m,t-1} + \tau) + \gamma_e s_{m,t} C_s + \gamma_a [(1 - u_{m,t})(\Phi_{m,t-1} + \tau) + u_{m,t} l_{m,t}] + \gamma_e p_{m,t} l_{m,t} \tag{18a}$$

$$= s_{m,t}[\gamma_e C_s - \gamma_a u_{m,t}(\phi_{m,t-1} + \tau)] + \gamma_a [u_{m,t}(\phi_{m,t-1} + \tau) + (1 - u_{m,t})(\Phi_{m,t-1} + \tau) + u_{m,t} l_{m,t}] + \gamma_e p_{m,t} l_{m,t} \tag{18b}$$

$$= s_{m,t} C_{m,t,1} + C_{m,t,2} \tag{18c}$$

In $C_{m,t,1}$ and $C_{m,t,2}$, $\phi_{m,t-1}$ can be calculated by the BS based on the historical data in slot $t-1$ [13] and $\Phi_{m,t-1}$ is known for the BS. In addition, the BS can calculate $\gamma_{m,t}$ according to Equations (4) and (5); thus, $u_{m,t}$ and $l_{m,t}$ can be further calculated according to Equations (7) and (9) given $p_{m,t}$, which means that $C_{m,t,1}$ and $C_{m,t,2}$ depend on $p_{m,t}$ and are independent of $s_{m,t}$. Hence, the optimal sample collection commands to minimize $s_{m,t}C_{m,t,1} + C_{m,t,2}$, denoted as $s^*_{m,t}$, are achieved when the term $s_{m,t}C_{m,t,1}$ is at its minimum; thus, we have

$$s^*_{m,t} = \begin{cases} 1, & C_{m,t,1} < 0 \\ 0, & otherwise \end{cases}. \tag{19}$$

Hence, the optimal sample collection commands can be determined according to Equation (19) when $p_{m,t}$ is given and Equation (13) can be rewritten as

$$\min_{\boldsymbol{p}_t} \left[\gamma_a \overline{\Phi} + \gamma_e \bar{\varepsilon}\right] \tag{20}$$

$$s.t. \quad p_{m,t} \in [0, P_{m,max}], \forall m \in \mathcal{M}, \forall t \in \mathcal{T}, \tag{20a}$$

$$s^*_{m,t} = \begin{cases} 1, & C_{m,t,1} < 0 \\ 0, & otherwise \end{cases}. \tag{20b}$$

According to Equation (20), the action $\boldsymbol{a}_t$ is only reflected by $\boldsymbol{p}_t$. Therefore, DDPG, which is suitable for the continuous action space, can be employed as the desired algorithm to solve the optimization problem in Equation (20).

- **Reward function:** The BS aims to minimize the AoI and energy consumption of the MIMO-NOMA IoT system, and the target of the DDPG algorithm is to maximize the reward function. Therefore, the reward function in slot t can be defined as

$$r_t(\boldsymbol{o}_t, \boldsymbol{p}_t) = -\sum_{m \in \mathcal{M}} [\gamma_a \Phi_{m,t} + \gamma_e \varepsilon_{m,t}]. \tag{21}$$

Furthermore, the expected long-term discounted reward of the system can be defined as

$$J(\mu) = \mathbb{E}\left[\sum_{t=1}^{T} \beta^{t-1} r_t(\boldsymbol{o}_t, \boldsymbol{p}_t)|_{\boldsymbol{p}_t = \mu(\boldsymbol{o}_t)}\right], \tag{22}$$

where $\beta \in [0,1]$ is the discounting factor, $\boldsymbol{p}_t = \mu(\boldsymbol{o}_t)$ indicates the action under the state $\boldsymbol{o}_t$, which is derived through policy μ. Thus, our objective in this paper becomes finding the optimal policy to minimize $J(\mu)$.

4.2. Optimizing Power Allocation Based on DDPG

In this subsection, we will introduce the architecture of the DDPG algorithm including primary networks (an actor network and a critic network) and target networks (a target actor network and a target critic network) [38], wherein the actor network is adopted for policy approximation and improvement, the critic network is adopted for policy evaluation and the target networks are adopted to improve the stability of the algorithm. Both primary and target networks are neural networks (DNNs). The flow diagram is shown in Figure 2. Denote θ, ζ, θ' and ζ' as parameters of the actor network, critic network, target actor network and target critic network, respectively, μ_θ as the policy approximated by actor network and Δ_t as the noise added upon action for the exploration in slot t.Next, we will present the training stage of the DDPG algorithm in detail.

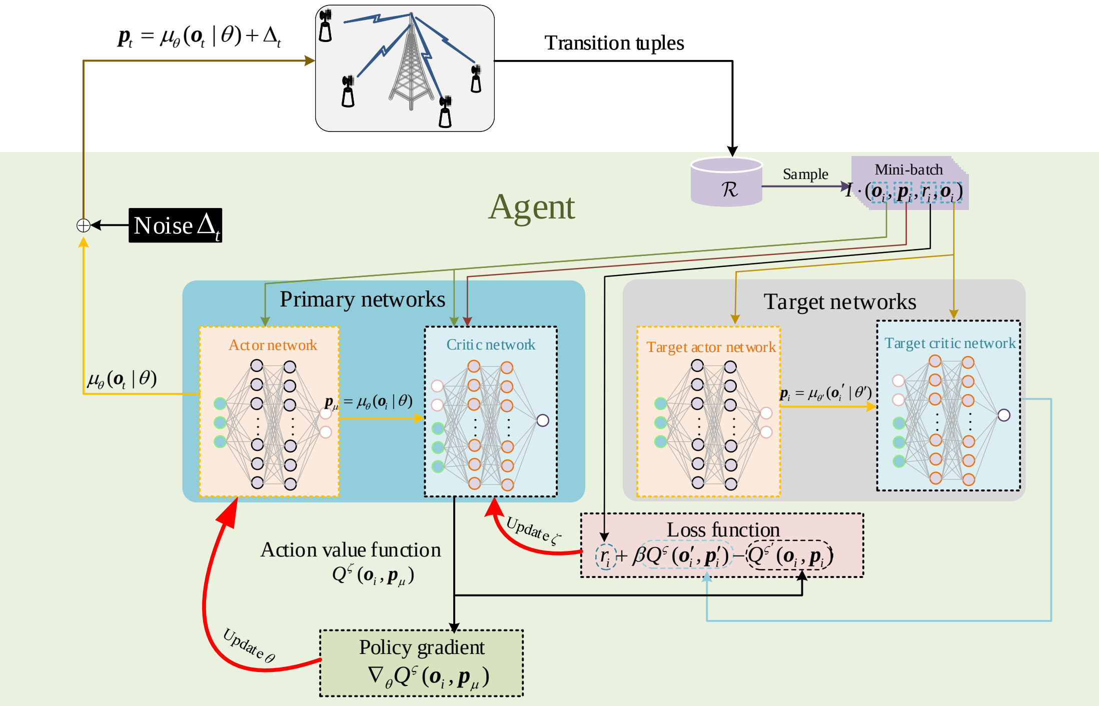
$p_t = \mu_\theta(o_t|\theta) + \Delta_t$
Transition tuples
$\mathcal{R}$
Sample
Mini-batch
$I \cdot (o_i, p_i, r_i, o_i')$
Agent
Noise Δ_t
Primary networks
Target networks
Actor network
Critic network
Target actor network
Target critic network
$\mu_\theta(o_t|\theta)$
$p_\mu = \mu_\theta(o_i|\theta)$
$p_i = \mu_{\theta'}(o_i'|\theta')$
Update ζ
Loss function
$r_i + \beta Q^{\varsigma'}(o_i', p_i') - Q^{\varsigma}(o_i, p_i)$
Action value function
$Q^{\varsigma}(o_i, p_\mu)$
Policy gradient
$\nabla_\theta Q^{\varsigma}(o_i, p_\mu)$
Update θ

Figure 2. Flow diagram of DDPG.

The parameters θ and ζ are first initialized randomly, θ' and ζ' are set as θ and ζ, respectively. In addition, a replay experience buffer $\mathcal{R}$ is built to cache the state transitions (lines 1–3).

Next, the algorithm loops for E episodes. At the beginning of each episode, the simulation parameters of the system model are reset as $u_{m,0} = 0$, $p_{m,0} = 1$ and $\Phi_{m,0} = 0$ for each device m, $\boldsymbol{h}_m(0)$ is initialized randomly. Given $p_{m,0}$ and $\boldsymbol{h}_m(0)$, the SINR $\gamma_{m,0}$ is calculated according to Equations (2)–(4); then, the state of each device m, i.e., $\boldsymbol{o}_{m,1} = [u_{m,0}, \gamma_{m,0}, \Phi_{m,0}]$ is observed by the agent (lines 4–6).

Afterwards, the algorithm iterates from slot 1 to T. For slot t, the actor network yields the output $\mu_\theta(\boldsymbol{o}_t|\theta)$ under the observed state $\boldsymbol{o}_t$ and policy μ with parameters θ. Then, a noise Δ_t is generated and the agent calculates the transmission powers of all devices according to $\boldsymbol{p}_t = \mu_\theta(\boldsymbol{o}_t|\theta) + \Delta_t$. After that, the agent calculates the $u_{m,t}$, $s_{m,t}$ and $\gamma_{m,t}$ of each device m according to Equations (4), (7) and (19), respectively. Afterwards, the agent calculates $\Phi_{m,t}$ and $\varepsilon_{m,t}$ according to Equations (8) and (11), respectively, and thus obtains the state of slot t, i.e., $\boldsymbol{o}_{t+1}$, and then calculates r_t according to Equation (21). The above tuple $[\boldsymbol{o}_t, \boldsymbol{p}_t, r_t, \boldsymbol{o}_{t+1}]$ in the replay buffer. Then, the agent inputs $\boldsymbol{o}_{t+1}$ into the actor network and starts the next iteration if the number of samples in the replay buffer is not larger than I (lines 7–10).

If the number of tuples in the replay buffer exceeds I, the parameters θ, θ', ζ and ζ' will be updated to maximize $J(\mu_\theta)$. Here, θ is updated toward the direction of the gradient $\nabla_\theta J(\mu_\theta)$. Specifically, the agent uniformly retrieves a mini-batch consisting of I tuples from the replay buffer. For each tuple i, i.e., $(\boldsymbol{o}_i, \boldsymbol{p}_i, r_i, \boldsymbol{o}_i')$ $(i \in \{1, 2, \ldots, I\})$, the agent inputs $\boldsymbol{o'}_i$ into the target actor network and outputs $\boldsymbol{p}_i' = \mu_{\theta'}(\boldsymbol{o'}_i|\theta')$, inputs $\boldsymbol{o}_i'$ and $\boldsymbol{p}_i'$ into the target critic network and outputs $Q^{\zeta'}(\boldsymbol{o}_i', \boldsymbol{p}_i')$ and then calculates the target value as

$$y_i = r_i + \beta Q^{\zeta'}(\boldsymbol{o}_i', \boldsymbol{p}_i')|_{\boldsymbol{p}_i' = \mu_{\theta'}(\boldsymbol{o'}_i|\theta')}. \tag{23}$$

While $\boldsymbol{o}_i$ and $\boldsymbol{p}_i$ are the input and $Q^{\zeta}(\boldsymbol{o}_i, \boldsymbol{p}_i)$ is the output of the critic network, the loss function can be expressed as

$$L(\zeta) = \frac{1}{I}\sum_{i=1}^{I}\left[y_i - Q^{\zeta}(\boldsymbol{o}_i, \boldsymbol{p}_i)\right]^2. \tag{24}$$

Then, the critic network is updated by the gradient descending method with the gradient of loss function $\nabla_{\zeta}L(\zeta)$ [39] (lines 11–13), i.e.,

$$\zeta \leftarrow \zeta - \alpha_c \nabla_{\zeta}L(\zeta), \tag{25}$$

where α_c is the learning rate of the critic network.

After that, the agent calculates the gradient $\nabla_{\theta}J(\mu_{\theta})$ as [40]

$$\begin{aligned} &\nabla_{\theta}J(\mu_{\theta}) \\ &\approx \frac{1}{I}\sum_{i=1}^{I}\nabla_{\theta}Q^{\zeta}(\boldsymbol{o}_i, \boldsymbol{p}_{\mu})|_{\boldsymbol{p}_{\mu}=\mu_{\theta}(\boldsymbol{o}_i|\theta)} \\ &= \frac{1}{I}\sum_{i=1}^{I}\nabla_{\theta}\mu_{\theta}(\boldsymbol{o}_i|\theta)\cdot\nabla_{\boldsymbol{p}_{\mu}}Q^{\zeta}(\boldsymbol{o}_i, \boldsymbol{p}_{\mu})|_{\boldsymbol{p}_{\mu}=\mu_{\theta}(\boldsymbol{o}_i|\theta)} \end{aligned}, \tag{26}$$

where the chain rule is applied to derive the gradient of $Q^{\zeta}(\boldsymbol{o}_i, \boldsymbol{p}_{\mu})$ with respect to θ [40]. Given $\nabla_{\theta}J(\mu_{\theta})$, the actor network can be updated by gradient ascending to maximize $J(\mu_{\theta})$, i.e.,

$$\theta \leftarrow \theta + \alpha_a \nabla_{\theta}J(\mu_{\theta}), \tag{27}$$

where α_a is the learning rate of the actor network.

After the parameters of the primary networks are updated, the parameters of the target networks are updated based on the parameters of primary networks, i.e.,

$$\begin{aligned} \zeta' &\leftarrow \kappa\zeta + (1-\kappa)\zeta' \\ \theta' &\leftarrow \kappa\theta + (1-\kappa)\theta' \end{aligned}, \tag{28}$$

where κ is a constant much smaller than 1, i.e., $\kappa \ll 1$ (line 15).

Up to now, the iteration for slot t is finished and the agent starts the next iteration until the number of slots reaches T. Then, the agent starts the next episode. When the number of episodes reaches E, the training stage is finished and outputs the near-optimal policy. The pseudo-code of the training stage is described in Algorithm 1.

Next, the testing stage is initialized to test the performance under the near-optimal policy. Compared with the training stage, the parameter-updating process is omitted in the testing process and actions in each slot are generated by the near-optimal policy. The corresponding pseudo-code is shown in Algorithm 2, where θ^* is the parameter to achieve the near-optimal policy in the training stage.

Algorithm 1: Training stage of the DDPG algorithm

```
Input: $\gamma, \tau, \theta, \zeta$
Output: optimized DNNs
Randomly initialize the $\theta, \zeta$;
Initialize target networks by $\zeta' \leftarrow \zeta, \theta' \leftarrow \theta$;
Initialize replay experience buffer $\mathcal{R}$;
for episode from 1 to $E$ do
    Reset simulation parameters for the system model;
    Receive initial observation state $\boldsymbol{o}_1$;
    for slot $t$ from 1 to $T$ do
        Generate the transmission power of all devices according to the current
          policy, state and exploration noise $\boldsymbol{p}_t = \mu_\theta(\boldsymbol{o}_t|\theta) + \Delta_t$ ;
        Execute action $\boldsymbol{p}_t$, observe reward $r_t$ and new state $\boldsymbol{o}_{t+1}$ from the system
          model;
        Store transition tuple $(\boldsymbol{o}_t, \boldsymbol{p}_t, r_t, \boldsymbol{o}_{t+1})$ in $\mathcal{R}$;
        if number of tuples in $\mathcal{R}$ is larger than $I$ then
            Randomly sample a mini-batch of $I$ transitions tuples from $\mathcal{R}$;
            Update the critic network by minimizing the loss function according to
              Equation (25);
            Update the actor network according to Equation (27);
            Update target networks according to Equations (28).
        end
    end
end
```

Algorithm 2: Testing stage of the DDPG algorithm

```
for episode from 1 to $E$ do
    Reset simulation parameters for the system model;
    Receive initial observation state $\boldsymbol{o}_1$;
    for slot $t$ from 1 to $T$ do
        Generate the transmission power of all devices according to the
          near-optimal policy and current state $\boldsymbol{p}_t = \mu_\theta(\boldsymbol{o}_t|\theta^{m*})$ ;
        Execute the action $\boldsymbol{p}_t$;
        Observe reward the $r_t$ and new state $\boldsymbol{o}_{t+1}$.
    end
end
```

4.3. Complexity Investigation

In this subsection, we investigate the complexity of the proposed algorithm. Denote G_P and U_p as the computational complexity for computing gradients and updating parameters of the primary networks, respectively. Since the architecture of the target networks is the same as that for the primary networks, the computational complexity for updating the parameters of the target networks is also the same as the one for the primary networks. The complexity of the proposed algorithm is related to the number of slots in the training process. To be specific, during each slot, the primary networks calculate the gradients and updating parameters, while the target networks update the parameters with the parameters of the primary networks according to Equation (28). Moreover, we denote the complexity of calculating sample decisions based on power allocation as S_d. Thus, the complexity of the proposed algorithm in a slot is $O(G_P + 2U_P + S_d)$. Note that the gradients calculation and parameters updating will be processed until the number of tuples cached in the replay buffer exceeds I. The proposed algorithm will loop for E episodes, each of which contains T slots. Thus, the complexity of the proposed algorithm can be expressed as $O((E \cdot T - I)(G_P + 2U_P + S_d))$.

5. Simulation Results and Analysis

In this section, we provide simulation results to verify the effectiveness of the proposed power allocation strategy. The scenario is described in the system model. The experiments were conducted during the training and testing phases. The simulation tool was Python 3.6. In the simulation, both the actor network and critic network are the four-layer fully connected DNN with two hidden layers which are equipped with 400 and 300 neurons, respectively. The Adam optimization method [41] is adopted to update the parameters of the critic network and actor network. The noise Δ_t (for exploration) follows the Ornstein–Uhlenbeck (OU) process with decay-rate 0.15 and variation 0.004, respectively [42]. The small-scale fading of each device is initialized by white Gaussian noise, and the Rayleigh block fading model is employed to simulate the stochastic small-scale fading [43]. The reference channel gain of each device is -30 dB when the communications distance is 1 m, the path-loss exponent is 2 and the communication distances is randomly set within a range of $[50, 100]$ meters. The parameters of the measurement setup and DDPG algorithm are set according to [13] and [39], respectively, which are shown in Table 3.

Table 3. Values of the parameters in the experiments.

Parameters of System Model [13]			
Parameter	**Value**	**Parameter**	**Value**
τ	0.1 s	K	4
W	18 kHz	C_s	0.5 J
$P_{m,max}$	2 W	T	500
Parameters of Agent [39]			
Parameter	**Value**	**Parameter**	**Value**
κ	0.001	I	64
E	800	β	0.99
$\lvert\mathcal{R}\rvert$	2.5×10^5	γ_e	0.5
γ_a	0.5	α_a	10^{-3}
α_c	10^{-4}	F_c/F_m	0.8/0.5
B	10	N_{GA}	50

5.1. Training Stage

Figure 3 shows the learning curves in the training stage, i.e., rewards in different episodes, for different numbers of IoT devices. It can be seen that the rewards of different curves rise and fluctuate from episode 0 to 150, which reflects that the agent is learning the policy to maximize the average reward. After that, the learning curves turn out to be stable, which indicates that the near-optimal policy has been learned by the agent. Note that there is a litter jitters after episode 150, which is due to the fact that the agent is adjusting slightly since the exploration noise prevents the agent from converging into the local optima. It can also be seen that the large number of devices incurs a low reward. This is attributed to the fact that each device will be affected by more interference as the number of devices in the system increases, which leads to the lower transmission rate. This will prolong the transmission delay and further increase the AoI of the system. Then, the BS may inform the devices to consume more energy to sample more frequently and transmit faster; thus, the lower AoI can be guaranteed.

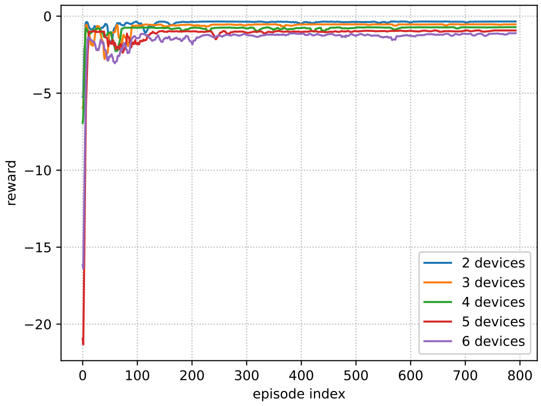

Figure 3. Learning curves under various number of devices.

5.2. Testing Stage

In the testing stage, we verify the performance of the near-optimal policy obtained in the training stage. Existing works have adopted GA [32] and random power allocation policy [33] as the baseline algorithm for power allocation; therefore, we selected these two algorithms for comparison. Here, random power allocation policy and GA are introduced as follows:

- Random policy: Randomly allocate the power of each device m within $[0, P_{m,max}]$ and the sample collection commands is obtained according to Equation (19).
- GA-based power allocation: In each time slot, the BS randomly generates a population vector according to $P_{m,max}$ and a population size B. Each individual element in the population vector stands for the power allocation for all devices. The BS selects the best individuals in the population vector as offspring according to their fitness, i.e., the reward function, of each individual. Then, after evolving for N_{GA} times, for each evolution, the probabilities of crossover and mutation for these offspring become F_c and F_m, respectively, where crossover means that two individuals in the offspring exchange the power allocation of a random device, and mutation means that the power allocation of any device in the offspring is selected within $[0, P_{m,max}]$ randomly. After that, selecting best individuals from the offspring that has experienced crossover and mutation as the input for the next evolution. After all the evolutions, the best individual from the last offspring, which is the near-optimal power allocation derived by GA, is elected. After that, the BS calculates the optimal sample collection based on the near-optimal power allocation derived by GA according to Equation (19), and then executes the near-optimal power allocation derived by GA and the optimal sample collection. In the end, the BS iterates into the next time slot.

Figure 4 presents the AoI of the near-optimal policy derived by DDPG, the random policy and the near-optimal power allocation derived by GA. It can be seen that the AoI of the three policies increases as the number of devices increases. This is because each device will suffer from the interference as the number of devices increases, and thus degrades its transmission delay according to Equation (9), which may further increase the AoI of the system. Meanwhile, the near-optimal policy derived by DDPG and the near-optimal power allocation derived by GA always outperform the random policy, because the near-optimal policy derived by DDPG can adjust the power allocation adaptively according to the observed state, and the near-optimal power allocation derived by GA will find the optimal power allocation according to the fitness in the evolutions to ensure a low AoI, while the random policy just generates power allocation randomly. It also can be seen that the near-optimal policy derived by DDPG outperforms the near-optimal power allocation derived by GA, because the DDPG will consider the influence of the power allocation in each time slot on the subsequent AoI, while GA cannot.

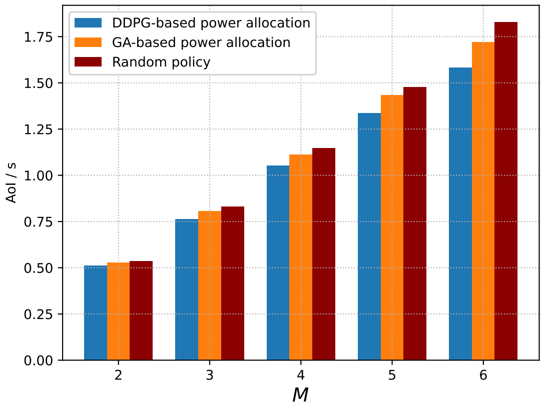

Figure 4. AoI of the system vs. number of devices.

Figure 5 compares the energy consumption of three policies. It can be seen that energy consumption increases as the number of devices increases. This is due to the fact that, according to Equation (4), the increasing number of vehicles increases the interference power, leading to a decrease in SINR. According to Equation (9), the AoI of the system increases as the SINR decreases. Hence, the devices may consume more energy for more frequent sampling and faster transmission to reduce AoI. Moreover, the increasing number of devices contributes to the increasing energy consumption according to Equation (12). Meanwhile, the near-optimal policy derived by DDPG and the near-optimal power allocation derived by GA always outperform the random policy, because DDPG and GA can allocate power adaptively to ensure a low-energy consumption. Moreover, it also can be seen that the near-optimal policy derived by DDPG always outperforms the near-optimal power allocation derived by GA, which is due to the fact that the GA cannot take into account the influence of the power allocation in each time slot on the subsequent energy consumption.

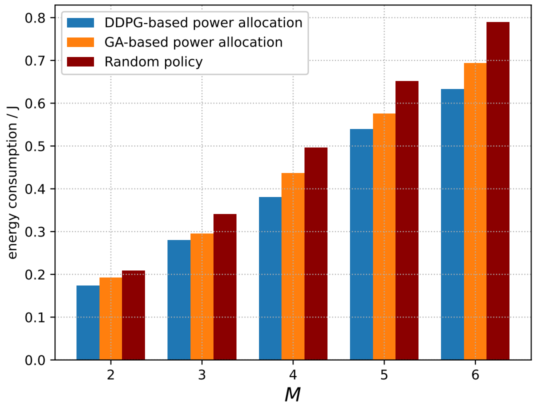

Figure 5. Energy consumption of the system vs. number of devices.

Figure 6 compares the average reward under the three policies, where the reward is obtained by averaging the test results over all slots. We can see that the average reward decreases as the number of devices increases. This is due to the fact that the reward function consists of the AoI and energy consumption of the system according to Equation (21), and both of them increase as the number of devices increases. Moreover, the average reward under the near-optimal policy derived by DDPG and the near-optimal power allocation derived by GA are higher than that of the random policy. This is attributed to the fact that the near-optimal policy allocates power according to the observed state to maximize the long-term discounted reward, and the GA obtains the near-optimal power allocation by maximizing the reward. It can also be seen that the near-optimal policy obtained by the DDPG-based method always outperforms the near-optimal power allocation derived by the GA. This is due to the fact that the GA aims to find the near-optimal power allocation based on fitness, i.e., the reward in each slot, while ignoring the long-term reward maximization.

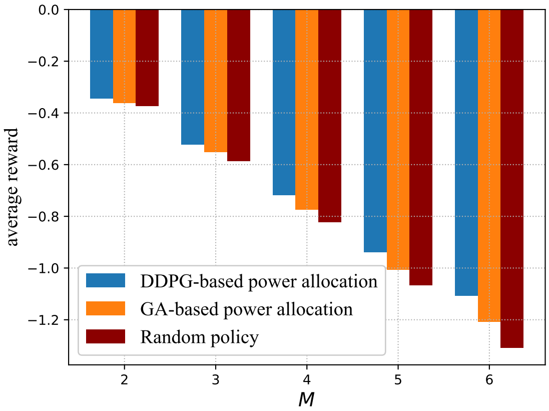

Figure 6. Average reward vs. number of devices.

Figure 7 shows the relationship between the AoI of the system and packet size, i.e., Q, under three policies. It can be seen that the AoI increases as the packet size increases under the three policies. This is due to the fact that, according to Equation (9), the packet size influences the transmission delay. That is, the transmission delay is long when the packet size is large. With regards to Equation (8), the AoI is affected by transmission delay, wherein a smaller transmission delay results in a smaller AoI. In addition, we can see that the AoI of the near-optimal policy obtained by DDPG and the near-optimal power allocation derived by the GA are lower than the AoI under the random policy. This is because the near-optimal policy derived by DDPG can adjust the power allocation based on the observed state, and the GA obtain near-optimal power allocation according to fitness, which can significantly reduce the AoI of the system. The gap between the near-optimal power allocation derived by DDPG and the near-optimal power allocation derived by the GA is caused by the advantage of long-term minimization for DDPG.

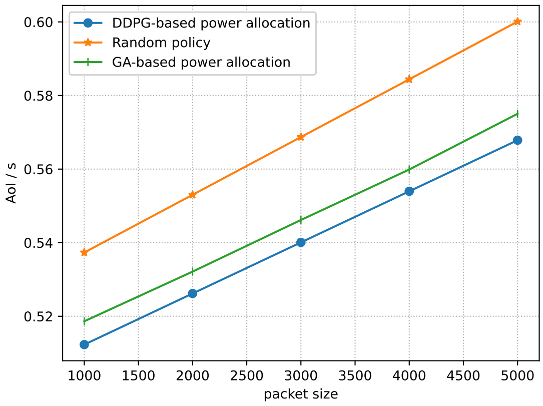

Figure 7. AoI of the system vs. packet size.

Figure 8 shows the relationship between the energy consumption of the system and packet size under three policies. It can be seen that the energy consumption of all three policies increases when the packet size increases. As shown in Figure 7, the transmission delay is long when the packet size is large, thus incurring the increase in energy consumption of the system. We can also see that the energy consumption of the near-optimal policy derived by DDPG and the near-optimal power allocation derived by the GA are lower than that of the random policy. This is due to the fact that the near-optimal policy derived by DDPG can adaptively allocate power and the GA can obtain the near-optimal power allocation according to fitness to ensure a lower energy consumption. However, the near-optimal policy derived by DDPG accounts for the influence of power allocation on the energy consumption of later time slots; thus, the near-optimal policy obtained by DDPG has a lower energy consumption than the near-optimal power allocation derived by the GA.

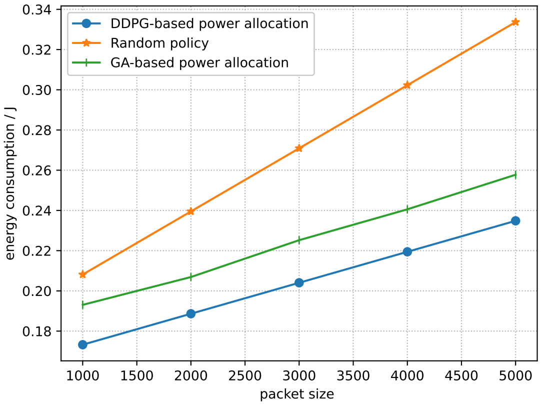

Figure 8. Energy consumption vs. packet size.

6. Conclusions

In this paper, we formulated a problem to minimize the AoI and energy consumption of the MIMO-NOMA IoT system. To solve it, we simplified the formulated problem and proposed the power allocation scheme based on DDPG to maximize the long-term discounted reward. Extensive simulations have demonstrated that the proposed scheme reduces the reward by 6.44% compared to the GA, and by 11.78% compared to the random policy, respectively. According to the theoretical analysis and simulation results, the key findings and contributions of this paper can be summarized as follows: (1) An increase in the number of devices and packet size will increase the AoI of the system. In this case, agents can inform the devices to consume more energy to sample more frequently and transmit faster, thereby reducing the AoI and increasing the energy consumption. (2) The near-optimal policy trained by DDPG outperforms the baseline policy under different numbers of users and packet sizes, which has a good capability to suit the system dynamic variation. We also noted some limitations and future directions for further research in this study: DDPG may face challenges when addressing high-dimensional state and action spaces. In future work, we will consider decomposing the problem into multiple subtasks for independent learning or improving the function approximators to enhance its robustness. In addition, as mentioned in [44], fairness is also a relatively important factor in the NOMA system. Therefore, our future research will focus on achieving a fair resource allocation in MIMO-NOMA systems and evaluating its impact on other performances.

Author Contributions: Conceptualization, Z.Z., H.Z. and Q.W.; Methodology, Z.Z., H.Z. and Q.W.; Software, Z.Z. and H.Z.; Writing—Original Draft Preparation, Z.Z.; Writing—Review and Editing, Q.F., P.F., H.Z. and J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant No. 61701197, in part by the open research fund of State Key Laboratory of Integrated Services Networks under Grant No. ISN23-11, in part by the National Key Research and Development Program of China under Grant No. 2021YFA1000500(4), in part by the 111 project under Grant No. B23008.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: Author Qiang Fan was employed by the company Qualcomm. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Wu, Q.; Wang, X.; Fan, Q.; Fan, P.; Zhang, C.; Li, Z. High stable and accurate vehicle selection scheme based on federated edge learning in vehicular networks. *China Commun.* **2023**, *20*, 1–17. [CrossRef]
2. Wu, Q.; Wang, S.; Ge, H.; Fan, P.; Fan, Q.; Letaief, K.B. Delay-Sensitive Task Offloading in Vehicular Fog Computing-Assisted Platoons. *IEEE Trans. Netw. Serv. Manag.* 2023, *early access*. [CrossRef]
3. Wu, Q.; Zhao, Y.; Fan, Q.; Fan, P.; Wang, J.; Zhang, C. Mobility-Aware Cooperative Caching in Vehicular Edge Computing Based on Asynchronous Federated and Deep Reinforcement Learning. *IEEE J. Sel. Top. Signal Process.* **2023**, *17*, 66–81. [CrossRef]
4. Gao, Y.; Xia, B.; Xiao, K.; Chen, Z.; Li, X.; Zhang, S. Theoretical Analysis of the Dynamic Decode Ordering SIC Receiver for Uplink NOMA Systems. *IEEE Commun. Lett.* **2017**, *21*, 2246–2249. [CrossRef]
5. Wu, Q.; Shi, S.; Wan, Z.; Fan, Q.; Fan, P.; Zhang, C. Towards V2I Age-aware Fairness Access: A DQN Based Intelligent Vehicular Node Training and Test Method. *Chin. J. Electron.* **2022**, *32*, 1–15. [CrossRef]
6. Bo, Z.; Saad, W. Joint Status Sampling and Updating for Minimizing Age of Information in the Internet of Things. *IEEE Trans. Commun.* **2019**, *67*, 7468–7482.
7. Zhu, H.; Wu, Q.; Wu, X.J.; Fan, Q.; Fan, P.; Wang, J. Decentralized Power Allocation for MIMO-NOMA Vehicular Edge Computing Based on Deep Reinforcement Learning. *IEEE Internet Things J.* **2022**, *9*, 12770–12782. [CrossRef]
8. Long, D.; Wu, Q.; Fan, Q.; Fan, P.; Li, Z.; Fan, J. A Power Allocation Scheme for MIMO-NOMA and D2D Vehicular Edge Computing Based on Decentralized DRL. *Sensors* **2023**, *23*, 3449. [CrossRef] [PubMed]
9. Volodymyr, M.; Koray, K.; David, S.; Rusu, A.A.; Joel, V.; Bellemare, M.G.; Alex, G.; Martin, R.; Fidjeland, A.K.; Ostrovski, G. Human-level control through deep reinforcement learning. *Nature* **2019**, *518*, 529–533.
10. Zhao, T.; He, L.; Huang, X.; Li, F. DRL-Based Secure Video Offloading in MEC-Enabled IoT Networks. *IEEE Internet Things J.* **2022**, *9*, 18710–18724. [CrossRef]
11. Chen, Y.; Sun, Y.; Yang, B.; Taleb, T. Joint Caching and Computing Service Placement for Edge-Enabled IoT Based on Deep Reinforcement Learning. *IEEE Internet Things J.* **2022**, *9*, 19501–19514. [CrossRef]
12. Grybosi, J.F.; Rebelatto, J.L.; Moritz, G.L. Age of Information of SIC-Aided Massive IoT Networks With Random Access. *IEEE Internet Things J.* **2022**, *9*, 662–670. [CrossRef]
13. Wang, S.; Chen, M.; Yang, Z.; Yin, C.; Saad, W.; Cui, S.; Poor, H.V. Distributed Reinforcement Learning for Age of Information Minimization in Real-Time IoT Systems. *IEEE J. Sel. Top. Signal Process.* **2022**, *16*, 501–515. [CrossRef]
14. Abd-Elmagid, M.A.; Dhillon, H.S.; Pappas, N. AoI-Optimal Joint Sampling and Updating for Wireless Powered Communication Systems. *IEEE Trans. Veh. Technol.* **2020**, *69*, 14110–14115. [CrossRef]
15. Li, C.; Huang, Y.; Li, S.; Chen, Y.; Jalaian, B.A.; Hou, Y.T.; Lou, W.; Reed, J.H.; Kompella, S. Minimizing AoI in a 5G-Based IoT Network Under Varying Channel Conditions. *IEEE Internet Things J.* **2021**, *8*, 14543–14558. [CrossRef]
16. Hatami, M.; Leinonen, M.; Codreanu, M. AoI Minimization in Status Update Control with Energy Harvesting Sensors. *IEEE Trans. Wireless Commun.* **2021**, *69* , 8335–8351. [CrossRef]
17. Sun, M.; Xu, X.; Qin, X.; Zhang, P. AoI-Energy-Aware UAV-assisted Data Collection for IoT Networks: A Deep Reinforcement Learning Method. *IEEE Internet Things J.* **2021**, *8*, 17275–17289. [CrossRef]
18. Hu, H.; Xiong, K.; Qu, G.; Ni, Q.; Fan, P.; Letaief, K.B. AoI-Minimal Trajectory Planning and Data Collection in UAV-Assisted Wireless Powered IoT Networks. *IEEE Internet Things J.* **2021**, *8*, 1211–1223. [CrossRef]
19. Emara, M.; Elsawy, H.; Bauch, G. A Spatiotemporal Model for Peak AoI in Uplink IoT Networks: Time Versus Event-Triggered Traffic. *IEEE Internet Things J.* **2020**, *7*, 6762–6777. [CrossRef]
20. Lyu, L.; Dai, Y.; Cheng, N.; Zhu, S.; Guan, X.; Lin, B.; Shen, X. AoI-Aware Co-Design of Cooperative Transmission and State Estimation for Marine IoT Systems. *IEEE Internet Things J.* **2021**, *8*, 7889–7901. [CrossRef]
21. Wang, X.; Chen, C.; He, J.; Zhu, S.; Guan, X. AoI-Aware Control and Communication Co-Design for Industrial IoT Systems. *IEEE Internet Things J.* **2021**, *8*, 8464–8473. [CrossRef]
22. Hao, X.; Yang, T.; Hu, Y.; Feng, H.; Hu, B. An Adaptive Matching Bridged Resource Allocation Over Correlated Energy Efficiency and AoI in CR-IoT System. *IEEE Trans. Green Commun. Netw.* **2022**, *6*, 583–599. [CrossRef]
23. Yilmaz, S.S.; Özbek, B.; İlgüy, M.; Okyere, B.; Musavian, L.; Gonzalez, J. User Selection for NOMA based MIMO with Physical Layer Network Coding in Internet of Things Applications. *IEEE Internet Things J.* **2021**, *9*, 14998–15006. [CrossRef]
24. Shi, Z.; Wang, H.; Fu, Y.; Yang, G.; Ma, S.; Hou, F.; Tsiftsis, T.A. Zero-Forcing-Based Downlink Virtual MIMO–NOMA Communications in IoT Networks. *IEEE Internet Things J.* **2020**, *7*, 2716–2737. [CrossRef]
25. Wang, Q.; Wu, Z. Beamforming Optimization and Power Allocation for User-Centric MIMO-NOMA IoT Networks. *IEEE Access* **2021**, *9*, 339–348. [CrossRef]
26. Han, L.; Liu, R.; Wang, Z.; Yue, X.; Thompson, J.S. Millimeter-Wave MIMO-NOMA-Based Positioning System for Internet-of-Things Applications. *IEEE Internet Things J.* **2020**, *7*, 11068–11077. [CrossRef]
27. Zhang, S.; Wang, L.; Luo, H.; Ma, X.; Zhou, S. AoI-Delay Tradeoff in Mobile Edge Caching With Freshness-Aware Content Refreshing. *IEEE Trans. Wireless Commun.* **2021**, *20*, 5329–5342. [CrossRef]

28. Chinnadurai, S.; Yoon, D. Energy Efficient MIMO-NOMA HCN with IoT for Wireless Communication Systems. In Proceedings of the 2018 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Republic of Korea, 17–19 October 2018; pp. 856–859. [CrossRef]
29. Gao, J.; Wang, X.; Shen, R.; Xu, Y. User Clustering and Power Allocation for mmWave MIMO-NOMA with IoT devices. In Proceedings of the 2021 IEEE Wireless Communications and Networking Conference (WCNC), Nanjing, China, 29 March–1 April 2021; pp. 1–6. [CrossRef]
30. Feng, W.; Zhao, N.; Ao, S.; Tang, J.; Zhang, X.; Fu, Y.; So, D.K.C.; Wong, K.K. Joint 3D Trajectory and Power Optimization for UAV-Aided mmWave MIMO-NOMA Networks. *IEEE Trans. Commun.* **2021**, *69*, 2346–2358. [CrossRef]
31. Ding, Z.; Dai, L.; Poor, H.V. MIMO-NOMA Design for Small Packet Transmission in the Internet of Things. *IEEE Access* **2016**, *4*, 1393–1405. [CrossRef]
32. Bulut, I.S.; Ilhan, H. Energy Harvesting Optimization of Uplink-NOMA System for IoT Networks Based on Channel Capacity Analysis Using the Water Cycle Algorithm. *IEEE Trans. Green Commun. Netw.* **2021**, *5*, 291–307. [CrossRef]
33. Ullah, S.A.; Zeb, S.; Mahmood, A.; Hassan, S.A.; Gidlund, M. Deep RL-assisted Energy Harvesting in CR-NOMA Communications for NextG IoT Networks. In Proceedings of the 2022 IEEE Globecom Workshops (GC Wkshps), Rio de Janeiro, Brazil, 4–8 December 2022; pp. 74–79. [CrossRef]
34. Kang, J.M.; Kim, I.M.; Chun, C.J. Deep Learning-Based MIMO-NOMA With Imperfect SIC Decoding. *IEEE Syst. J.* **2020**, *14*, 3414–3417. [CrossRef]
35. He, X.; Huang, Z.; Wang, H.; Song, R. Sum Rate Analysis for Massive MIMO-NOMA Uplink System with Group-Level Successive Interference Cancellation. *IEEE Wirel. Commun. Lett.* **2023**, *12*, 1194–1198. [CrossRef]
36. Wang, S.; Chen, M.; Saad, W.; Yin, C.; Cui, S.; Poor, H.V. Reinforcement Learning for Minimizing Age of Information under Realistic Physical Dynamics. In Proceedings of the GLOBECOM 2020—2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6. [CrossRef]
37. Arulkumaran, K.; Deisenroth, M.P.; Brundage, M.; Bharath, A.A. Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]
38. Qiao, G.; Leng, S.; Maharjan, S.; Zhang, Y.; Ansari, N. Deep Reinforcement Learning for Cooperative Content Caching in Vehicular Edge Computing and Networks. *IEEE Internet Things J.* **2020**, *7*, 247–257. [CrossRef]
39. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2015**, arXiv:1509.02971.
40. Silver, D.; Lever, G.; Heess, N.; Degris, T.; Wierstra, D.; Riedmiller, M. Deterministic Policy Gradient Algorithms. In Proceedings of the 2014 International Conference on Machine Learning(ICML), Beijing, China, 21–26 June 2014; pp. 387–395.
41. Kingma, D.P.; Ba, J. ADAM: A method for stochastic optimization. *arXiv* **2015**, arXiv:1412.6980.
42. Uhlenbeck, G.E.; Ornstein, L.S. On the Theory of the Brownian Motion. *Rev. Latinoam. Microbiol.* **1973**, *15*, 29–35. [CrossRef]
43. Ngo, H.Q.; Larsson, E.G.; Marzetta, T.L. Energy and Spectral Efficiency of Very Large Multiuser MIMO Systems. *IEEE Trans. Commun.* **2013**, *61*, 1436–1449. [CrossRef]
44. Darsena, D.; Gelli, G.; Iudice, I.; Verde, F. A Hybrid NOMA-OMA Scheme for Inter-plane Intersatellite Communications in Massive LEO Constellations. *arXiv* **2023**, arXiv:2307.08340.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.