remote sensing

Article

Deep Reinforcement Learning-Based Adaptive Modulation for Underwater Acoustic Communication with Outdated Channel State Information

Yuzhi Zhang [1,*], Jingru Zhu [1], Haiyan Wang [2,3], Xiaohong Shen [2], Bin Wang [1] and Yuan Dong [4]

[1] School of Communication and Information Engineering, Xi'an University of Science and Technology, Xi'an 710054, China
[2] School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China
[3] School of Electronics and Information, Xi'an Polytechnic University, Xi'an 710054, China
[4] School of Electronics and Control Engineering, Chang'an University, Xi'an 710064, China
[*] Correspondence: yuzhizhang@xust.edu.cn

Abstract: Underwater acoustic (UWA) adaptive modulation (AM) requires feedback about channel state information (CSI) but the long propagation delays and time-varying features of UWA channels can cause the CSI feedback to be outdated. When the AM mode is selected by outdated CSI, the mismatch between the outdated CSI and the actual CSI during transmission degrades the performance and can even lead to communication failure. Reinforcement learning has the ability to learn the relationships between adaptive systems and the environment. This paper proposes a deep Q-network (DQN)-based AM method for UWA communication that uses a series of outdated CSI as the system input. Our study showed that it could extract channel information and select appropriate modulation modes in the expected channels more effectively than single Q-learning (QL) without needing a deep neural network structure. Furthermore, to mitigate any decision bias that was caused by partial observations of UWA channels, we improved the DQN-based AM by integrating a long short-term memory (LSTM) neural network, named LSTM-DQN-AM. The proposed scheme could enhance the DQN's ability to remember and process historical input channel information, thus strengthening its relationship mapping ability for state-action pairs and rewards. The pool and sea experimental results demonstrated that the proposed LSTM-DQN-AM outperformed DQN-, QL- and threshold-based AM methods.

Keywords: adaptive modulation; channel state information; deep reinforcement learning; underwater acoustic communication

check for updates

Citation: Zhang, Y.; Zhu, J.; Wang, H.; Shen, X.; Wang, B.; Dong, Y. Deep Reinforcement Learning-Based Adaptive Modulation for Underwater Acoustic Communication with Outdated Channel State Information. *Remote Sens.* **2022**, *14*, 3947. https://doi.org/10.3390/rs14163947

Academic Editors: Songzuo Liu, Nan Chi and Zhi Sun

Received: 24 May 2022
Accepted: 11 August 2022
Published: 14 August 2022

1. Introduction

Oceans cover nearly two thirds of the earth and are rich in natural resources. Although the majority of underwater regions remain unknown and unexplored, humans have gradually expanded ocean development and exploration for various applications, including renewable energy, underwater mining and offshore oil and gas facilities. In ocean exploration, underwater communication techniques play an important role in remote sensing and sensing information transmission. Underwater acoustic (UWA) communication is suitable for long-distance data transmission [1]. Despite decades of development, UWA communication and networking is still a very challenging research area due to the complex and harsh underwater environments [2]. UWA channels have the characteristics of severe path loss, noise, limited bandwidth and service energy, long propagation delays and severe multi-path and Doppler effects [3,4]. At the same time, channels are temporal and spatial and have varying frequencies [5,6]. In this situation, when transmitters only use a single communication mode, it is difficult to maintain a robust performance in the long-term deployment of UWA systems [7]. When the modes of transmitters are designed according to the worst channel state, the reliability of the communication can be guaranteed; however, when the channel states later improve, the spectrum efficiency becomes lower. In this case, transmitters cannot work efficiently or robustly in UWA time-varying channels when using single modulation schemes. Adaptive modulation (AM) technology allows transmitters to adjust their modulation mode according to the different channel conditions and can increase system throughput while maintaining communication reliability. Considering the characteristics of UWA channels, the classical threshold-based adaptive switching method has been investigated for UWA AM. In UWA systems, one of the main challenges in utilizing channel state information (CSI) is feedback delay. Additionally, low sound propagation speeds result in long propagation delays. Non-negligible delays are required to estimate CSI at receivers and feed it back to transmitters. In time-varying UWA channels, feedback-estimated CSI is not the exact CSI for the adaptive data transmission and is too outdated for AM decisions. There are usually two remedies to address the delay and time variation effects of UWA AM: channel prediction for more accurate CSI and the design of stable metrics that are less sensitive to channel variations.

Some works have focused on channel prediction-based UWA AM. For UWA adaptive orthogonal frequency division multiplexing (OFDM) systems, modulation levels and the power of subcarriers are adjusted according to the predicted CSI [8]. Channels are predicted one time step ahead, based on the feedback-estimated channel impulsive response. Experimental results have demonstrated that performance gains can be achieved by using predicted values instead of outdated CSI. The classical channel prediction methods assume that the channel variations behave in linear or smooth nonlinear fashions [9,10]. However, this assumption does not always hold for practical UWA channels. The deep learning-based methods can track channel variations even when they do not form linear patterns. A CSI prediction model that is based on online deep learning has been proposed [11] for UWA adaptive orthogonal frequency division multiple access (OFDMA). Considering the channel correlations in both the time and frequency domains, the authors designed a neural network that integrated a one-dimensional convolutional neural network (CNN) and a long short-term memory (LSTM) network. Their experimental data analysis showed that its performance was better than the widely used recursive least predictor.

For the design of mode switching metrics, effective signal to noise ratio (ESNR) has been proposed as a performance metric for adaptive modulation and coding mode switching [12]. It is more consistent for evaluating system performance than the input signal to noise ratio (SNR) or pilot SNR. The ESNR metric has been further discussed and analyzed in [13]. In order to design a consistent metric for AM in UWA time-varying channels, an adaptive system has been proposed that exploits the long-term stability of the second-order statistics of CSI [14]. An adaptive coding and bit-power loading algorithm has also been proposed to find the optimal bit rate for OFDM communication with a fixed error rate. In order to explore the channel correlation characteristics of UWA adaptive OFDMA systems, the subcarrier correlation coefficient has been proposed to measure the correlations between feedback CSI and actual CSI in data transmissions [15]. The experiments in [16] further analyzed the effects of correlation coefficients on adaptive OFDMA.

In machine learning (ML), AM mode selection can be regarded as a classification problem for which the mode selection metric is produced by machine learning-based models [17–20]. Compared to threshold-based AM methods, ML-based AM can easily incorporate multi-dimensional CSI as the classification parameters. Since the ML-based classification process is trained to identify the features of channels, it is less sensitive to potential mismatches in environmental information than direct decision-making methods [8,12]. In [17], a decision tree was trained to associate channels with modulation schemes under a target bit error rate (BER) and the relationships between the BER and all relevant channel characteristics were extracted from large amounts of transmissions from a phase-shift keying (PSK) modem. From the perspective of statistical analysis plus ML, a UWA adaptive modulation and coding scheme that is based on sparse principal component analysis has been proposed [18] to explore key CSI as a more efficient switching metric.

Based on environmental information and prior training using various channel types, a support vector machine (SVM)-based channel classification method has also been proposed [19] to output the optimal modulation scheme for the expected channel. Designed for channel modeling uncertainty, an attention-aided k-nearest neighbor (A-kNN) algorithm has been proposed for UWA adaptive modulation and coding [20] and a dimensionality-reduced and data-clustered A-kNN AMC classifier has been presented to reduce the complexity.

In recent years, with the development of artificial intelligence, reinforcement learning (RL) algorithms have exhibited promising learning abilities for complex UWA problems, such as path planning [21], routing [22,23], localization [24], resources allocation [25] and adaptive modulation [26–31]. RL algorithms are training learning models for decision-making. RL does not require expert knowledge to train the learning systems and the internal relationships and knowledge are dynamically created during the learning rounds. Therefore, it can handle decision-making problems with time series data very well. In UWA AM, as RL training reflects the relationships between AM modes and the available UWA CSI, it can work efficiently even under non-ideal CSI conditions. In [26], an online algorithm in a model-based RL framework was proposed to recursively estimate the model parameters of the channels, track the channel dynamics and compute the optimal transmission parameters to minimize the long-term system costs. A Dyna-Q algorithm that was based on a UWA AM strategy was developed in [27], which selected the modulation order based on the feedback CSI from the receiver to maximize the long-term throughput. The Dyna-Q algorithm jointly played two roles: predicting CSI and calculating the communication throughput of each modulation order under different channel states for AM selection. In [28], the authors proposed an RL-based adaptive modulation and coding scheme that considered multiple quality of service (QoS) factors, including information QoS requirements, previous transmission quality and energy consumption. An efficient RL-based UWA image communication algorithm was proposed in [29], which could improve image quality while reducing energy consumption and latency in fast time-varying UWA channels. In [28,29], the channels were predicted independently. Our previous investigation results proved that the performance of QL-based AM could be further improved by integrating LSTM channel prediction [30]. The research in [31] considered a set of hybrid modulation strategies, including frequency shift keying, single-carrier communication and multi-carrier communication. It presented an RL-based AM switching strategy that was based on a deep-deterministic policy gradient (DDPG) algorithm, with the goal of maximizing long-term energy efficiency and spectral efficiency.

The above RL-based UWA AM methods outperform direct threshold-based methods in time-varying channels with non-ideal CSI. However, when the dimensionality of data is high, basic RL cannot cope with that dimensionality. In order to further enhance the performance of intelligent agents, deep reinforcement learning (DRL) combines deep neural networks (DNNs) and RL to learn successful policies directly from high-dimensionality inputs [32,33]. In the field of UWA communication and networking, DRL-based adaptive media access control [34,35] can avoid additional collisions by sensing the channel status and accessing idle time slots. Meanwhile, DRL is also applied to autonomous underwater vehicle control, which can apply continuous action policies in complex UWA environments [36,37]. In the field of wireless radio communication, there have been several studies exploring the application of DRL in AM [38–41]. For energy harvesting systems, the research in [38] presented a DRL-based optimal strategy to allocate transmission power and adjust M-ary modulation levels based on the obtained causal information about the harvested energy, battery states and channel gain. When the statistics for energy harvesting were unknown, a deep Q-network (DQN) was employed to find the optimal solution in continuous state space to achieve the maximum system throughput. In the presence of unknown multi-user interference, a heterogeneous network AM method has been designed using DRL [39]. An adaptive modulation scheme that is based on DQN has also been proposed to select rate region boundaries as the states [40]. However, this scheme assumes that the distribution of CSI is known in advance and that perfect CSI can be obtained.

Considering outdated CSI, a deep reinforcement learning-based adaptive modulation (DRL-AM) approach was proposed in [41] to tackle CSI variations in complex channels in a nonlinear manner. By using outdated CSI as the input for the DRL, it could achieve a higher throughput than a linear autoregressive moving average prediction model.

In practical UWA systems, RL-based AM strategies that use single-time outdated CSI do not produce optimal solutions due to the partial observations of the UWA channels [26–30]. The performance of DRL-based AM in wireless radio communication can be improved with multi-dimensional inputs [41]. In this paper, considering complex nonlinear varying UWA channels with unknown CSI distributions, we aimed to improve the performance of RL-based AM by combining RL with a DNN and a recurrent neural network. LSTM neural networks are among the powerful recurrent neural networks that can find intrinsic correlations in information from time series data [42], even with partially observed information. They also show impressive performances in the speech recognition [43] and image processing [44] domains. In wireless communication, DRL with partially observable conditions has been studied using LSTM networks for the navigation of unstaffed surface vehicles [45] and unstaffed aerial vehicles [46].

Motivated by the problems of UWA AM and the advantages of deep learning [47,48] and reinforcement learning, this paper proposes an LSTM-enhanced DQN-based adaptive modulation (LSTM-DQN-AM) scheme for time-varying UWA channels with outdated feedback CSI. The performance of this scheme was evaluated through multiple experiments. The main contributions of this paper include:

- An end-to-end DQN-based adaptive modulation scheme with multi-dimensional outdated CSI inputs for UWA communication. In this study, the state of the DQN was set to a time series of effective SNR, which could better track channel properties compared to the single time inputs of QL for UWA AM in [27–30]. Previous studies on threshold-based methods [8,11,12] and QL-based methods [26,28–30] have generally treated channel prediction and mode selection as two separate problems. The proposed LSTM-DQN-AM scheme could act as a joint solution for prediction and mode selection. Through the combination of the DNN structure, experience replay and dynamic greedy algorithm, DQN had a better and more stable performance than the QL-based AM methods.
- An LSTM-enhanced DQN network structure for UWA AM, which utilizes LSTM to memorize and extract features from the historical input channel state information of UWA channels. By exploiting the advantages of LSTM, LSTM-DQN-AM could extract the hidden correlations between the states and the environment, even in UWA channels with nonlinear variations and partially observed CSI. Finally, LSTM-DQN-AM could more accurately map the relationships between state-action pairs and rewards for AM decision-making. The proposed LSTM-DQN-AM scheme was more effective and robust than classical DQN and QL methods [27–30] under time-varying and partially observable conditions.

The remainder of this paper is organized as follows. Section 2 presents the UWA AM system model. Section 3 proposes the LSTM-DQN-AM scheme for performance improvement. Section 4 presents the experimental results in comparison to those from other RL- and DRL-based UWA AM methods. Section 5 provides a discussion about the study and results. Finally, Section 6 concludes the paper.

2. System Model

Figure 1 illustrates the general AM model for a UWA communication system. The transmitter adaptively changes its modulation mode according to the CSI that is fed back by the receiver. First, the transmitter sends signals to the receiver. Then, on the receiver side, the channel estimation module extracts the envelope and phase information from the signals to estimate the CSI. The estimated CSI is also used in the demodulation module to output data. Then, the receiver feeds back the estimated CSI and demodulation results to the transmitter. Finally, on the transmitter side, the modulation mode selection module determines the appropriate modulation mode and the transmitter performs adaptive modulation to send out data.

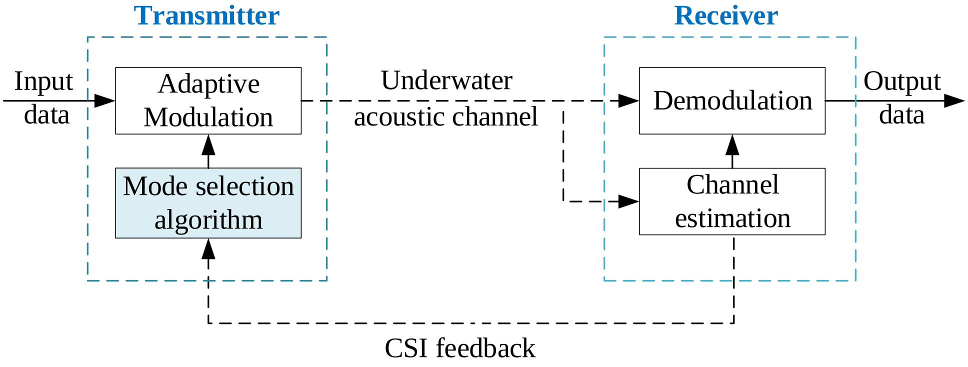


Figure 1. A system model of UWA AM.

In this study, the objective of AM was to select the appropriate modulation mode to optimize throughput with quality of service constraints, which could be expressed as:

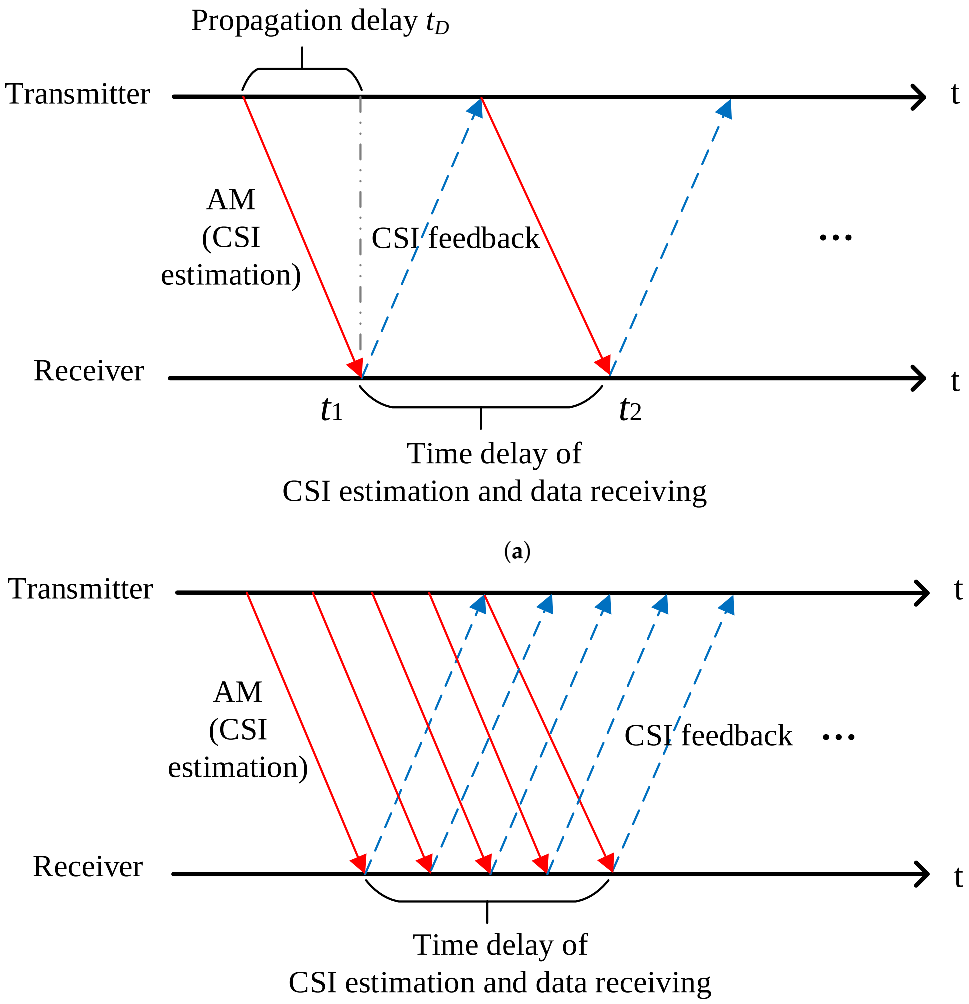
$$\begin{aligned} a_t^* &= \arg\max_{a_t \in \mathcal{A}} \left\{ R = \sum_{t=1}^{T} \gamma r_t \right\} \\ \text{s.t.} \quad & Pe_t \le BER_{\text{threshold}} \end{aligned} \tag{1}$$

where a_t^* is the action that was selected at time t, a_t^* is selected from the action space $\mathcal{A}$ (which represented the available modulation modes in the UWA AM), R is the accumulated reward, γ is the discount factor, r_t is the instantaneous reward and Pe_t is the actual BER at time t (which had to be less than the maximum constraint $BER_{\text{threshold}}$ to guarantee the quality of communication). During the AM mode selection, when the channel states were poor, the AM system selected a low-order modulation method with a lower bit rate to ensure the reliability of communication. On the contrary, in the case of good channel states, the system selected high-order modulation with a higher bit rate, which could improve the system throughput under the required BER constraints.

When the transmitter perfectly receives the CSI of the current time slot, the optimization problem of the modulation mode selection could be solved by exhausted searching. Unfortunately, due to the time-varying and long propagation delay characteristics of UWA channels, the estimated CSI became outdated after a long period of feedback. Figure 2 depicts how long propagation delays affected the AM process of UWA communication. After transmitting a data packet, the transmitter waited for the CSI feedback from the receiver. As shown in Figure 2a, the system took t_D to transmit data from the transmitter to the receiver and one more t_D to feed back the estimated CSI to the transmitter. Therefore, the transmitter had to select the AM mode of the current data packet based on the feedback CSI that was estimated $2t_D$ previously. Meanwhile, after such long delays, the actual channel state had already changed, i.e., the received feedback CSI was outdated. The decision that was based on the outdated CSI significantly downgraded the performance of the AM.

There was a total of $2t_D$ in time delay between the CSI estimation and the actual AM data reception. In UWA channels, the sound propagation speed is only about 1500 m/s. For example, when the communication distance is 750 m, the time that is required for CSI to become outdated is 1 s. As shown in Figure 2b, along with triggering more intensive transmissions [28], the overall throughput was improved by sending more data packets at the network protocol level. However, the effects of the propagation delays on CSI were still the same as those in Figure 2a.

When dealing with outdated CSI, the typical approach is to predict the current channel based on the available past CSI. While it is assumed that systems can be well approximated using linear models with known noise distributions, this assumption is often not satisfied in practical applications as the actual UWA channel variations are usually in relatively unknown nonlinear patterns, meaning that there would be performance degradation in realistic implementations. The RL-based method could be used to address this limitation. In particular, DRL has the capability to learn nonlinear dependencies between past and present channel state information and can also adapt to changes in the channel statistics.



(**a**)

(**b**)

Figure 2. The propagation delays in the UWA adaptive modulation process: (**a**) the propagation delays in classical UWA AM [12,27]; (**b**) the propagation delays in UWA AM with continuous transmissions [28].

3. LSTM-DQN-AM for UWA Communication

3.1. Framework

From the description of the system model, it can be seen that the system actions were discrete, which corresponded to the selection of different modulation modes. The UWA AM process is similar to the state transition process, in which an agent selects the AM mode based on the current system state and then acquires the next system state. The system state in the next round is related to the current state. Therefore, the AM process can be modeled as a Markov decision process.

Reinforcement learning can derive optimal strategies for Markov decision processes using a "trial and error" mechanism when interacting with the surrounding environment. The basic principle is that when the chosen action receives a large reward from the environment, the probability of the agent adopting this action in the future increases. On the contrary, when a low reward is obtained, the probability of the agent choosing the action decreases. Theoretical AM strategies usually assume that the channel state information is ideal. However, in dynamically changing UWA channels, the channel states are time- and frequency-varying and have outdated features. Reinforcement learning can train the optimal action strategy based on the available channel states and corresponding rewards. It can work without prior channel knowledge or system models and even with delayed imperfect CSI, it can still train the action under specific available states, which reveals the relationships between the outdated channel states and the actions.

Q-learning is one of the most classic algorithms in RL. In Q-learning, the agent aims to find the optimal policy π^* to maximize the accumulative reward. In the decision-making process of Q-learning, a two-dimensional table (named a Q-table) is used to store the state-action pairs and their corresponding Q-values for Q-value update and action selection. The states need to be discretized to construct the Q-table. When the number of states is particularly large, the convergence becomes dimensional. In order to overcome the shortcomings of Q-learning in complex multi-dimensional environments, we employed DQN to map the relationships between the state-action pairs and the Q-values.

Instead of a Q-table, DQN employs a DNN to generate the Q-values. By inputting the time series channel states into a DNN, it can output Q-values that produce different actions. This process is the relationship mapping of available state-action pairs and Q-values. To better derive effective actions from the channel state features, the authors of this study further designed an LSTM-DQN-AM for UWA communication. It could efficiently work with outdated feedback CSI to improve long-term throughput performance.

Figure 3 shows the basic framework of the designed DQN AM method. The learning environment was a UWA environment and the intelligent agent was a transmitter that learned optimal action AM modes under certain states. The specific design of the states, actions and rewards were as follows.

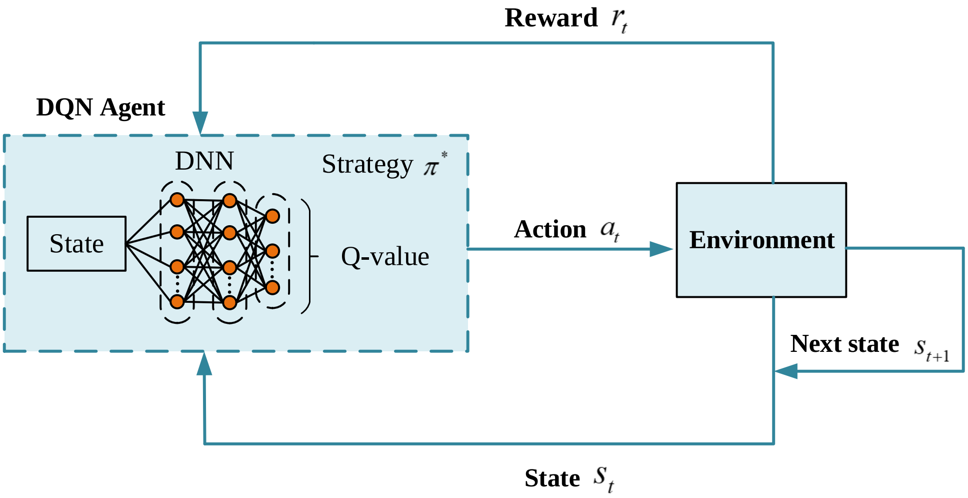


Figure 3. The framework of the DQN-based AM.

State. The state space of time slot t was a combination of the ESNR values of the previous z time slots. The ESNR values were estimated by the receiver and fed back to the transmitter. Different from input SNR and pilot SNR, the ESNR values were calculated after the channel estimation. ESNR takes the channel estimation error and the equalization error into account as noise; so, it was a more reasonable and consistent indicator of communication performance than SNR. The ESNR values were calculated as:

$$ESNR = \frac{E\left|\hat{b}(n)\right|^2}{E\left|b(n) - \hat{b}(n)\right|^2} \tag{2}$$

where $b(n)$ is the transmitted symbol and $\hat{b}(n)$ is the equalized outcome.

The state of the current time slot t was defined as:

$$\mathbf{s}_t = [ESNR_{t-z}, ESNR_{t-(z-1)}, \dots, ESNR_{t-1}] \tag{3}$$

where z is the historical length of the states. The current system state consisted of the previous ESNR values from time slot $(t-z)$ to time slot $(t-1)$. It should be noted that due to the long propagation delays, the latest feedback CSI was the ESNR at time $t-1$.

Action. The action space $\mathcal{A}$ contained all of the available modulation modes and was expressed as:

$$a_t \in \mathcal{A} = \left\{a^1, a^2, \dots, a^m, \dots, a^M\right\} \tag{4}$$

where a^m represents the different modulation modes and M is the total number of modulation modes.

Reward. The value of a reward was defined as:

$$r(t) = \begin{cases} b(1 - Pe_{s_t,a_t}) & Pe < BER_{\text{threshold}} \\ f_{\text{penalty}} & Pe \geqslant BER_{\text{threshold}} \end{cases} \tag{5}$$

where b is the number of modulation bits per symbol and Pe represents the BER when the transmitter selected a certain modulation mode in the current state. Equation (5) represents a reward after taking an action. When the BER was greater than the threshold, the reward was set to a penalty factor f_{penalty}, which was much less than 0. Then, the intelligent agent was less probable to choose the mode that resulted in a penalty. The first line in Equation (5) also indicates the normalized effective throughput (in bit/symbol) that could be achieved under the constraints of BER.

In basic reinforcement learning schemes, the corresponding reward of a state-action pair is an element of the Q-table. Through the interactions between $\mathbf{s}_t$, a_t, r_t and $\mathbf{s}_{t+1}$ and the environment, the action value function could be updated as:

$$Q(\mathbf{s}_t, a_t) \leftarrow Q(\mathbf{s}_t, a_t) + \alpha\left[r_t + \gamma \max Q\big(\mathbf{s}_{t+1}, a'\big) - Q(\mathbf{s}_t, a_t)\right] \tag{6}$$

where α is the learning rate, γ is the discounter factor, $Q(\mathbf{s}_t, a_t)$ is the Q-value of the selected action a_t under state $\mathbf{s}_t$, $\mathbf{s}_{t+1}$ denotes the new state that the channel transferred to after taking action a_t and $Q(\mathbf{s}_{t+1}, a')$ is the possible value after taking action a' at time slot $t+1$.

After the Q-table was updated, the modulation mode was selected according to a certain policy $\pi^*(s|a)$. In basic reinforcement learning, the policy is to directly select the action with the largest Q-value.

As shown in Equation (3), the agent could learn more channel information and make more reasonable decisions with larger values of z. However, for traditional Q-table-based RL algorithms, a larger z value results in a very large state space. Moreover, the Q-tables are extraordinarily large, resulting in very inefficient updates of $Q(s,a)$. For example, we set M to 4, representing four modulation modes, and assumed that the ESNR had 20 discrete values and was set z to 10 (which was a rather small history state length for CSI). Under these conditions, each action had 20^{10} possible state values and 4×20^{10} possible Q-values, which was very large for training. Therefore, in this paper, we employed the DRL technique to design the DRL-AM.

3.2. LSTM-DQN-AM Method

In reinforcement learning, the agent is initially unfamiliar with the environment, so it can only attempt to select actions. Through continuous interaction with the environment, the agent enhances good actions by judging the results of the selected action based on the feedback from the environment. Finally, the agent can learn the relationships between the environment and the actions to obtain the optimal decision. For learning in complex UWA environments, the agent neither needs to know the distribution of the channel statistics nor needs to assume a certain pattern of variations. The characteristics of channels can be well learned and mapped using AM systems through the power of deep learning.

The framework of the LSTM-DQN-based AM scheme is shown in Figure 4. The following subsections introduce the core ideas and structure of the scheme.

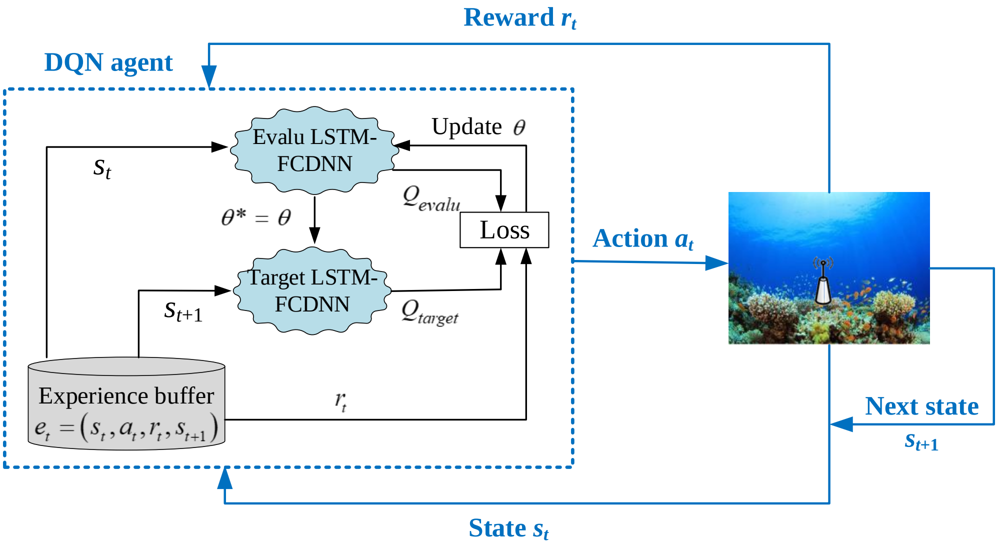


Figure 4. The framework of the LSTM-DQN-based AM scheme.

3.2.1. Generation of Q-Values

As the proposed state was a combination of time series CSI, the dimensionality of the state space was large, even when the number of discrete values of the states, actions and historical lengths of states (z) were small. Instead of the Q-tables that are used in Q-Learning, the states in DQN were used as the inputs of a DNN to generate Q-values that corresponded to all of the actions. DQN is the integration of Q-learning and a DNN, which can solve the dimensionality problem of large state spaces. In our DQN, a DNN was used to approximate the Q-values using:

$$Q(\mathbf{s}_t, a_t, \boldsymbol{\theta}) \approx Q(\mathbf{s}_t, a_t) \tag{7}$$

where $\boldsymbol{\theta}$ is the DNN parameter vectors. At time slot t, the state $\mathbf{s}_t$ was taken as a DNN input and the Q-values $Q(\mathbf{s}_t, a_t, \boldsymbol{\theta})$ were the output for each action. There was a total of M outputs for the different actions. Different from Formula (6), the DQN agent updated $Q(\mathbf{s}_t, a_t, \boldsymbol{\theta})$ by training $\boldsymbol{\theta}$ to update the Q-values by calculating the value of the Q-table. After convergence, the agent could obtain the Q-values $Q(\mathbf{s}_t, a_t, \boldsymbol{\theta})$ by inputting the states into the DQN.

3.2.2. Enhancing DQN Convergence

(1) **Experience Replay**

As shown in Figure 4, the DQN used experience replay to train the reinforcement learning processes, thereby addressing the problems of information correlation and non-static distribution. The DQN algorithm had an experience buffer $\mathbb{E}$ with a set capacity C_E to store historic learning experiences as:

$$\underset{e_t \in \mathbb{E}}{e_t} = (\mathbf{s}_t, a_t, r_t, \mathbf{s}_{t+1}) \tag{8}$$

In the training of the neural network, we usually assumed that each experience e_t was an independently and identically distributed process. However, in realistic implementations, these experiences would be correlated to some extent. The neural network was unstable when trained using correlated experiences.

When employing experience replay for the DQN, some historical experiences were randomly selected for neural network training. Then, the agent randomly selected a portion of memory from the experience pool (named a mini-batch B) to train the main network. This technique could break the correlations between datasets while improving the efficiency of the network updates.

The loss function in the dual deep neural network was calculated based on the values that were extracted from the experience replay.

(2) **Dual Deep Neural Network**

As shown in Figure 4, there were dual DNN networks in our DQN-based AM. One of the neural networks was an online evaluation network (EvaluNet), which outputted Q_{evalu}, and the other was the target network (TargetNet), which outputted Q_{target}. The DQN set a target network that generated the target Q-values separately from the online evaluation network. A specific training process could cause the neural network output to continuously approximate the target Q-values Q_{target}.

The loss function for the DQN training was:

$$Loss(\theta) = \frac{1}{N_E} \sum_{i \in B} \left(Q_{target} - Q_{evalu}\right)^2 \tag{9}$$

$$Q_{evalu} = Q(\mathbf{s}_i, a_i; \boldsymbol{\theta}) \tag{10}$$

$$Q_{target} = r_i + \gamma \max_{a'} Q\left(\mathbf{s}_{i+1}, a'; \boldsymbol{\theta}^*\right) \tag{11}$$

In Formula (9), N_E is the number of random experience samples, B is the experience buffer capacity and i is the number of episodes. Smaller loss function values indicated better network training results. In Formula (10), $\boldsymbol{\theta}$ is the parameter vector of the online evaluation network. In Formula (11), $\boldsymbol{\theta}^*$ is the parameter vector of the target network.

For parameter updates, the weight vector $\boldsymbol{\theta}^*$ of TargetNet was updated every F iterations by setting $\boldsymbol{\theta}^* = \boldsymbol{\theta}$. The dual network structure could make the network training process more stable. Because of the additional TargetNet in the network structure, the target Q-values remained unchanged for a period of time, which reduced the correlation between the current Q-value and the target Q-value, to some extent, and improved the stability of the algorithm.

(3) **Dynamic Greedy Algorithm**

In our proposed LSTM-DQN-AM, the optimal policy π^* employed a dynamic greedy algorithm to select the action that maximized $Q(s,a)$. A classical greedy-based action selection policy could be expressed as:

$$\pi^*(s|a) = \begin{cases} \arg\max(Q(s,a)), 1-\varepsilon \\ \text{random}, \qquad\qquad \varepsilon \end{cases} \tag{12}$$

where ε is a greedy factor. In order to avoid being trapped in a local optimum, the agent randomly selected one modulation mode with a probability of ε and selected the modulation mode that had the maximum Q-value with a probability of $1-\varepsilon$.

In our dynamic greedy algorithm strategy, ε gradually decayed by:

$$\varepsilon = \begin{cases} \varepsilon_{decay} \times \varepsilon, & \text{if} \quad \varepsilon > \varepsilon_{\min} \\ \varepsilon_{\min}, & \text{if} \quad \varepsilon \le \varepsilon_{\min} \end{cases} \tag{13}$$

where ε_{decay} is the decay coefficient of the greedy factor and ε_{min} is the minimum value of the greedy factor.

In the early learning stages of the DQN, the greedy factor ε was large. The agent could explore random actions with high probabilities to avoid local optimums. With the increase in the number of iterations, the greedy factor ε became smaller and the probability of the agent exploring random actions decreased. So, the agent reduced the extent of exploration and better exploited the learning results, which enhanced the learning efficiency.

The dynamic ε could maintain balance between exploration and exploitation to improve the system convergence speed and the results.

Based on the above analysis, the experience replay, dual DNN structure and dynamic greedy algorithm strategy made the DQN more efficient and stabler when dealing with complex problems.

3.2.3. Improved LSTM-Enhanced DQN

In DQN-based AM, the relationships between the state-action pairs and the Q-values are trained by inputting CSI into fully connected deep neural networks. In complex UWA environments, the state of the environment is time-varying. A transmitter with a DNN-based DQN makes decisions based on the currently sensed channel state information, which makes the transmitter forget any previous information. Therefore, the accuracy of the decision-making is limited in nonlinear and partially observable time-varying UWA channels. Compared to the fully connected DNN structure, recurrent neural networks (RNNs) have an improved memory ability. In RNNs, the recurrent neurons in the hidden layer are not only related to the output of the previous layer but also to the output of the current layer in the previous time period. LSTM is a special RNN that solves the problem of gradient disappearances in RNNs. In this study, we employed an LSTM to enhance the learning ability of the DQN.

As shown in Figure 5, the DNN structure in LSTM-DQN-AM was improved by adding an LSTM layer before the first layer of the fully connected (FC) deep neural network. The state $\mathbf{s}_t$ that consisted of the time series ESNR values was input into the LSTM of the DNN. After training, the Q-values that corresponded to the different actions were output.

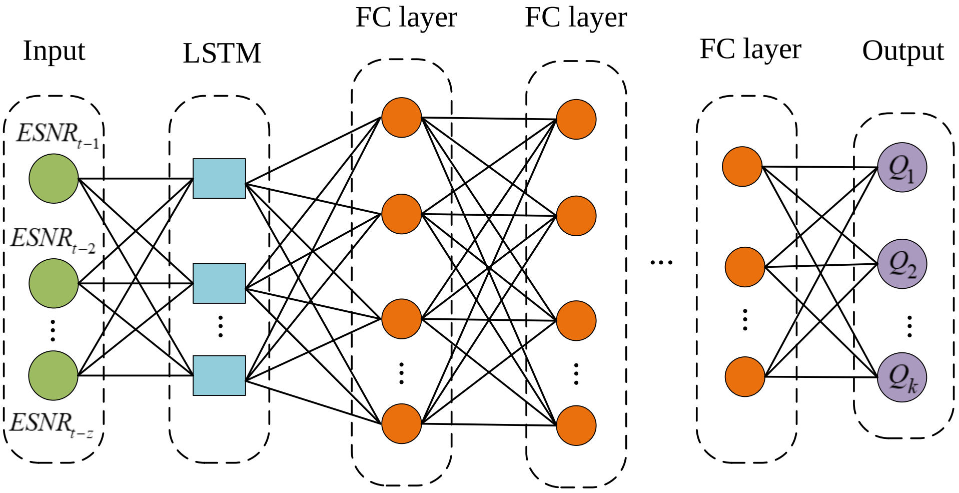


Figure 5. The LSTM-DQN network structure.

In classical DQNs, the hidden layers in the dual deep neural networks are usually FC DNNs. The output of the hidden layer could be expressed as:

$$\mathbf{y}_t = \phi(\mathbf{w} \cdot \mathbf{x}_t + \mathbf{b}) \tag{14}$$

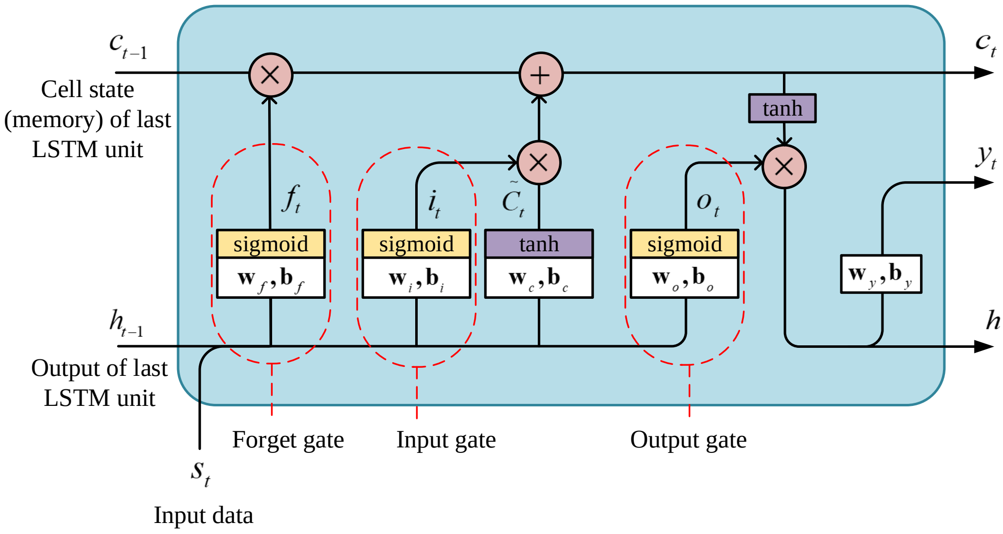
where $\mathbf{w}$ is the weight matrix of the hidden layer, $\mathbf{x}_t$ is the current input matrix, $\mathbf{b}$ is the bias term and ϕ is the activation function. From Equation (13), it can be seen that the output $\mathbf{y}_t$ only depended on $\mathbf{x}_t$. Therefore, for problems with complex temporal characteristics, the FC DNN could not track the temporal information.

Unlike the output of the FC DNN, the output of the LSTM network could be represented as:

$$\mathbf{y}_t = \phi\big(\mathbf{w}_x \cdot \mathbf{x}_t + \mathbf{w}_y \mathbf{y}_{t-1} + \mathbf{b}\big) \tag{15}$$

where $\mathbf{w}_x$ and $\mathbf{w}_y$ are the weight matrices and $\mathbf{y}_{t-1}$ is the output of the current layer at the previous time slot. It can be seen that the output was not only related to the input of the current time but also to the output of the previous time. The LSTM-enhanced DQN enabled the transmitter to effectively remember the previous states and extract the internal correlations, thereby enhancing the mapping effectiveness for the action-state pairs and rewards. For the UWA AM, it could finally achieve better action selection results by effectively learning from the partial available UWA CSI.

Figure 6 illustrates the structure of a single LSTM cell. The cell state $\mathbf{c}_t$ in an LSTM acts as a conveyor belt for data processing. An LSTM has three key gates, which control the cell states to optionally let information through: a forget gate, input gate and output gate. The gates consist of sigmoid neural net layers and pointwise multiplication operations. The three gates allow LSTM networks to save and update information in the long-term memory and the neural output acts as the short-term memory. The combination of long-term memory and short-term memory is the greatest advantage of LSTM networks in time series problems.



c_{t-1}
Cell state (memory) of last LSTM unit
c_t
tanh
y_t
f_t
i_t
$\tilde{C}_t$
o_t
sigmoid
$\mathbf{w}_f, \mathbf{b}_f$
sigmoid
$\mathbf{w}_i, \mathbf{b}_i$
tanh
$\mathbf{w}_c, \mathbf{b}_c$
sigmoid
$\mathbf{w}_o, \mathbf{b}_o$
$\mathbf{w}_y, \mathbf{b}_y$
h_{t-1}
Output of last LSTM unit
h_t
Forget gate
Input gate
Output gate
s_t
Input data

Figure 6. The structure of an LSTM cell.

Specifically for our LSTM-DQN-based AM, the channel state information $\mathbf{s}_t$ was the input for the LSTM network and the hidden state $\mathbf{h}_t$ was used to store information that was related to the channel states for the final output $\mathbf{y}_t$. For each time slot, the value of $\mathbf{h}_t$ was determined by the forget gate, the input gate and output gate and the cell state $\mathbf{c}_t$.

As shown in Figure 6, the forget gate was used to determine how much historical input state information to forget from the historical memory $\mathbf{h}_{t-1}$. The input gate layer decided which values were updated as $\mathbf{i}_t$ and a tanh layer created a vector for the new candidate values $\tilde{\mathbf{c}}_t$, which could be added to the cell state. Then, $\mathbf{i}_t$ and $\tilde{\mathbf{c}}_t$ were combined to update the cell state as $\mathbf{c}_t$. The output gate determined the output. The output was based on the cell state in a filtered version. A sigmoid layer decided which parts of the cell state were going to be output as $\mathbf{o}_t$. Then, the cell state went through the tanh layer to push the values to be between -1 and 1 and multiplied the values by the output of the sigmoid gate to obtain new memory $\mathbf{h}_t$:

$$\mathbf{o}_t = \sigma(\mathbf{w}_o \cdot [\mathbf{s}_t, \mathbf{h}_{t-1}] + \mathbf{b}_o) \tag{16}$$

$$\mathbf{h}_t = \mathbf{o}_t * \tanh(\mathbf{c}_t) \tag{17}$$

where $\mathbf{w}_o$ is the weight matrix of $\mathbf{o}_t$ (which corresponded to $\mathbf{s}_t$ and $\mathbf{h}_{t-1}$ in the output gate) and $\mathbf{b}_o$ is the bias term of o_t. Finally, based on $\mathbf{h}_t$, we could obtain the output of the LSTM as:

$$\mathbf{y}_t = \mathbf{w}_y \cdot \mathbf{h}_t + \mathbf{b}_y \tag{18}$$

where $\mathbf{w}_y$ is the output weight and $\mathbf{b}_y$ is the output bias.

With the updating of the LSTM unit, the complex temporal correlations that were hidden behind the partially observed UWA channels could be extracted and predicted, which could work well even with nonlinear channel variations.

The pseudocode of the proposed LSTM-DQN-AM is presented in Algorithm 1.

Algorithm 1: LSTM-DQN-AM

```
Initialize $M$, $BER_{\text{threshold}}$ for UWA AM system
Initialize $z$, $\varepsilon$, $\alpha$, $\gamma$ for QL
Initialize $\mathbb{E}$, $C_E$, $N_E$ for experience buffer
Initialize $\boldsymbol{\theta}$, $\boldsymbol{\theta}^*$, and $F$ of setting $\boldsymbol{\theta}^* = \boldsymbol{\theta}$ for dual neural network
Initialize $\varepsilon_{decay}$, $\varepsilon_{min}$ for dynamic greedy selection
Collect ESNR within $z$ time slots as initial state $\mathbf{s}_1$ for learning
for time slot $t$ from 1 to $T$ do
    Input $s_t$ into LSTM FC DNN EvaluNet and output Q-values
    Select $a_t$ via dynamic $\varepsilon$-greedy strategy, and update $\varepsilon$ for next round
    Take the selected action $a_t$
    Observe feedback $ESNR_t$, $r_t$ at a new state $\mathbf{s}_{t+1}$
        Define the channel state $\mathbf{s}_{t+1}$ as a times serious of outdated ESNR as Equation (3)
        Store experience $(\mathbf{s}_t, a_t, r_t, \mathbf{s}_{t+1})$ into $\mathbb{E}$
        Sample $N_E$ random mini-Batch from experience replay memory $\mathbb{E}$
        Calculate the system loss $Loss(\boldsymbol{\theta})$ as Equations (9)–(11)
        Minimize $Loss(\boldsymbol{\theta})$ using SGD, and update wight matrix of EvaluNet $\boldsymbol{\theta}$
        For every $F$ rounds, update wight of TargetNet by $\boldsymbol{\theta}^* = \boldsymbol{\theta}$
end for
```

4. Numerical Results

This section presents the numerical results that were used to evaluate the performance of the proposed LSTM-DQN-AM scheme. The system rewards were evaluated under the BER constraints and compared to those in other AM schemes. For convenience, the accumulated rewards were plotted, which were defined as the average values over the past N_r time slots.

The parameter settings of the QL and the hyperparameters of the LSTM-DNN neural network are listed in Table 1.

Table 1. The hyperparameter settings of LSTM-DQN-AM.

Parameters	Value
Historical Length of State z	30
Discount Factor of Reward γ	0.95
Learning Rate α	0.0001
Greedy Factor (Exploration Probability) ε	from 0.9 to 0.00001
Capacity of Experience Buffer $C_{\mathbb{E}}$	500
Number of Random Samples N_E in Experience Buffer	32
Update Frequency F of TargetNet	200
Number of FC DNN Layers	6
Number of Neurons in Each FC DNN Layer	64
Number of Neurons in LSTM Layer	128
Activation Function	ReLU

We compared the performance of the proposed scheme to those of the previously proposed AM schemes:

- **Threshold-AM:** Threshold switching AM with outdated feedback CSI [12];
- **QL-AM:** Q-learning-based AM with outdated feedback CSI [27];
- **LSTM-QL-AM:** Q-learning-based AM with LSTM-predicted CSI [30];
- **DQN-AM:** DQN-based AM with outdated feedback CSI [41];
- **LSTM-DQN-AM:** LSTM-DQN-based AM with outdated feedback CSI.

As the experimental data represented a realistic channel with nonlinear variations for a better evaluation of the AM schemes, the multiple AM schemes were compared using pool and sea experimental data.

In our experiments, the source information was a randomly generated bit sequence of 0 and 1. In practical applications, various information sources can be converted into digital sequences for transmissions.

As illustrated in Figure 2a, the transmitter initially triggered AM by sending out data using a randomly selected modulation mode. After the receiver received the data, it calculated the ESNR value according to its definition formula and then demodulated the received data. To guarantee the successful reception of the CSI, the estimated ESNR value and the demodulation result were fed back in the most robust modulation mode that the modem could support, i.e., BPSK. At the transmitter, $ESNR_{t-1}$ was combined with previous feedback ESNR values into channel state $\mathbf{s}_t$ for reinforcement learning and the ESNR–BER was updated according to the demodulation result. To set up the duration of the time slot, the propagation delay information was needed. In the simulations, the propagation delays could be calculated using the communication distance and the propagation speed of sound. In our practical UWA network implementation, the delay could be tested using the message exchanges from the acoustic modems. Many commercial underwater acoustic modems have a range/delay function for experiments.

For reinforcement learning, it was hard to obtain BER directly online. Before online implementation, BER could be trained using a pre-tested dataset. In the experiments, we could obtain the rough ESNR–BER relationship under this channel condition. In the online stage, the ESNR–BER relationship was continuously updated, based on the actual received results.

4.1. Pool Experimental Results

The pool experiments were conducted in a non-anechoic water pool that was 25 m long, 6 m wide and 1.6 m deep. In the experiments, the depth of the transmitter and receiver was about 0.45 m. The distance between the transmitter and the receiver was 20 m. Figure 7 shows the estimated channel impulsive responses (CIRs). It can be seen from the figure that the CIR changed even within a small time interval. CIR variations in the time domain caused channel gain variations and eventually led to SNR variations.

For the communication system setup, the transmitter supported four modulation modes (which is the same as in reference [28]): BPSK, QPSK, 8PSK and 16QAM. The carrier frequency was 20 kHz and the duration of one modulation symbol was 0.5 ms. The one-trip propagation delay was about 26 ms.

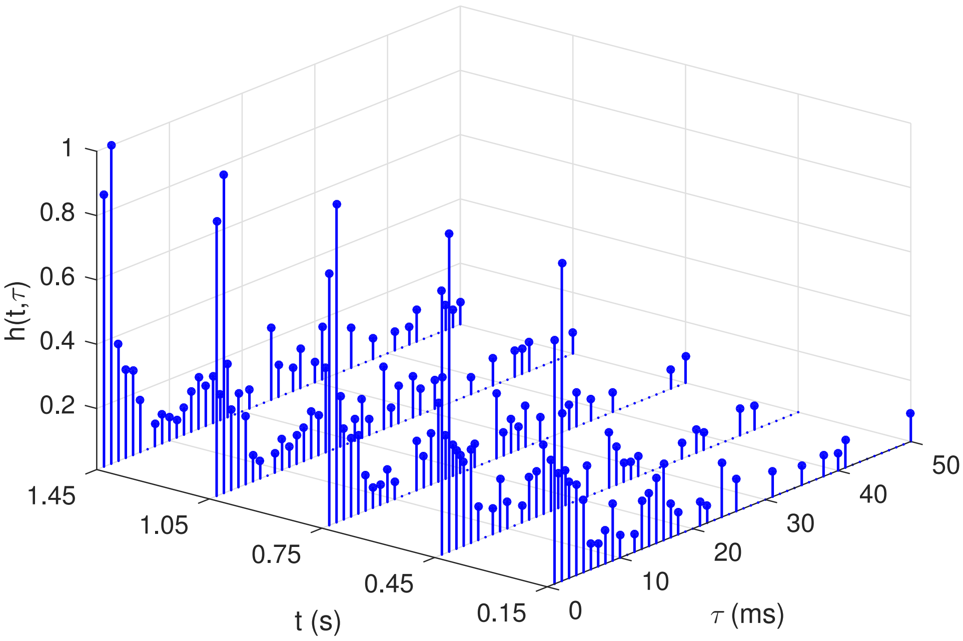


Figure 7. The channel impulsive responses from the pool experiments.

The system objective was to maximize the reward for higher throughput while maintaining the BER quality of service. From the reinforcement learning point of view, Figure 8 shows the reward performance of the multiple AM methods. A numerical comparison of the reward values is listed in Table 2. Figure 9 shows the BER performances of the multiple AM methods. From the communication point of view, Figure 10 shows the throughput of the multiple AM methods. The duration of each time slot mainly consisted of a one-trip transmission delay and the symbol duration. Especially in underwater acoustic channels, the propagation speed of sound is only about 1500 m/s. The one-trip transmission delay was usually much larger than the symbol duration, which affected the throughput results. For comparison across multiple communication scenarios, we defined the normalized throughput in bit/symbol as a unified metric, which meant that the average effective throughput could be achieved by transmitting a symbol. Figure 11 shows the number of successfully received bits per symbol after completing an action in each iteration *t*.

As shown in Figure 8, the proposed LSTM-DQN-AM had a higher reward after convergence than the other methods and delivered the fastest convergence speed. It was also the closest to the theoretical optimum value. The optimum value was calculated using perfect channel state information to select the modulation mode with the highest data rate under the BER constraints. The reward performance of the various learning-based schemes was LSTM-DQN-AM > DQN-AM > LSTM-QL-AM > QL-AM. As shown in Table 2, the long-term reward of the LSTM-DQN-AM scheme was 54% higher than that of DQN-AM, 60% higher than that of LSTM-QL-AM and 241% higher than that of QL-AM.

As shown in Figure 8, the reward of the proposed LSTM-DQN-AM outperformed that of the DQN-AM scheme [41]. LSTM-DQN-AM converged faster than the DQN-AM algorithm and the curve was more stable after convergence. This benefited from the additional recurrent neural network structure, so LSTM-DQN-AM could selectively and recurrently process the historical CSI to better track channel variations. Even when taking imperfect and/or outdated CSI as input, LSTM-DQN could map the relationships between the state-action pairs and rewards well.

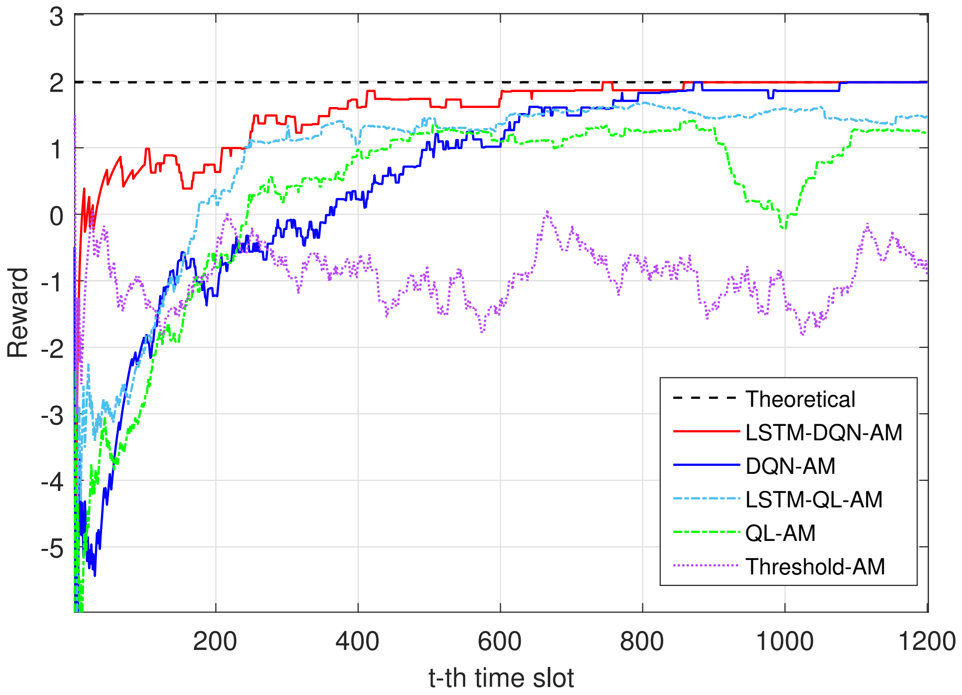


Figure 8. The rewards from the AM schemes in the pool experiments.

Table 2. A comparison of the rewards of the different AM schemes in the pool experiments.

AM Scheme	Compared AM Scheme	Improvement (%)
LSTM-DQN-AM	DQN-AM	54
LSTM-DQN-AM	LSTM-QL AM	60
LSTM-DQN-AM	QL-AM	241

As shown in Figure 8, the reward of the proposed LSTM-DQN-AM also outperformed LSTM-QL-AM [30], even though LSTM-QL-AM employed additional LSTM-based channel prediction. As the LSTM-DQN method jointly integrated an LSTM network into a DQN, the channel prediction and mode selection were jointly optimized. While in LSTM-QL-AM, the channel prediction and learning were two separate processes. For the DQN-based method, it was not necessary to employ a separate channel prediction process.

As shown in Figure 8, the proposed LSTM-DQN-AM performed better reward than the QL-AM scheme, which was proposed in [27] for UWA adaptive communication. The Q-table in QL-AM could only handle a low-dimensional finite number of states. The deep neural network in the DQN replaced the Q-table, so high-dimensional multi-feature CSI input could be processed for a more comprehensive representation of the UWA channels. Therefore, LSTM-DQN-AM could train the relationships between the state-action pairs and Q-values more effectively than QL-AM.

As shown in Figure 8, when comparing the reward of DQN-AM to those of QL-based methods, the DQN outperformed LSTM-QL-AM and QL-AM after convergence. As the deep network had better mapping abilities to express the relationships between the Q-learning and the actual UWA environment, the accumulated reward of the whole process was higher than that of the QL-based methods (although the DQN took more rounds to find the optimum solution before convergence). Since the QL-based AM could only update Q-tables with a limited capacity, it was difficult to achieve the optimal solution.

As shown in Figure 8, when comparing the two QL-AM schemes with and without channel prediction, it could be seen that the method with additional channel prediction was better than the single QL-AM. As it was combined with an LSTM for channel prediction, LSTM-QL-AM could acquire more accurate CSI for Q-table training, which ultimately led to better convergence results.

As shown in Figure 8, all of the learning-based AM methods outperformed the threshold-based AM with outdated feedback. Due to the long propagation delays and the time-varying features of UWA channels, the learning agent could only receive outdated CSI. The received outdated CSI did not match the actual data transmission CSI. The direct use of outdated CSI was inefficient for AM in UWA channels. Q-learning could map the relationships between the outdated CSI and the transmission results, which produced better selection results and reward performances than threshold-based AM. Through the improved structure of the deep neural network, the performance of the DQN-based methods could be further improved compared to the QL schemes.

Specifically, as shown in Figure 8, the convergence curve of QL-AM had a performance degradation between the 900th and 1100th time slots. This performance degradation was caused by the suddenly appearing channel variations. The variations in the realistic UWA channels could occur in nonlinear fashions. QL-based methods or linear-assumed systems could not handle these channel variations well, resulting in performance degradations in UWA AM. Benefiting from a deep neural network and a recurrent neural network, the reward performance of LSTM-DQN-AM remained robust in the face of complex channel variations. Unlike Q-learning, which only had a single CSI input, LSTM-DQN had outdated CSI sequences as the input, so the channels could be sufficiently learned and the dual deep neural network and experience replay structure in LSTM-DQN also enhanced the system's stability. In particular, LSTM-DQN incorporated an LSTM network with the capability to remember historical input sequences, which helped to extract and predict the complex hidden correlations from partially observable CSI.

Figure 9 shows the BER performances during the learning processes using the pool experimental data. It can be observed that as the number of iterations increased, the BER of the learning-based methods gradually decreased, which could then meet the requirements of the target values. However, the BER of the threshold-based scheme did not degrade below $BER_{threshold}$. Due to the threshold-based AM employing outdated CSI for its decision-making, it produced ineffective AM selection. The learning-based AM method could learn from outdated CSI for decision-making as the learning process could map the relationships between the outdated CSI and the system rewards. The BER of LSTM-DQN-AM had the fastest convergence speed. It was the first to converge below the BER threshold and it was the most stable after convergence compared to the other learning-based methods. Similar to the reward performance, the BER performance of QL-AM also had fluctuations between the 900th and 1100th time slots, while LSTM-DQN-AM produced a robust BER throughout the whole learning process.

As shown in Figure 9, in the initial stages, the BER performance of LSTM-QL-AM was better than that of QL-AM and DQN-AM, which directly adopted outdated CSI. LSTM-QL-AM used an independent prediction model to predict the current CSI using the outdated CSI sequences and then utilized the prediction CSI as the input for the Q-learning algorithm. Compared to QL-AM and DQN-AM, the improved performance of LSTM-QL-AM in the initial stages was due to the mode selection being based on predicted CSI. The proposed LSTM-DQN-AM also employed an LSTM network and DNN to learn the channels and could map the Q-values more accurately. Therefore, it could obtain a good BER performance in the initial stages.

Figure 10 shows the normalized effective throughput of the AM methods in the pool experiments. The proposed UWA AM adopted a time slot structure and the length of the time slots was related to the propagation delays. Therefore, effective throughput was not only related to successfully received bits but also to the slot length that was determined by the channel propagation delay. We normalized the throughput in bits per symbol, which represented the average successfully received bits per symbol and provided a consistent metric for comparison at different communication distances. In Figure 10, the x-axis represents the t-th slot and the y-axis represents the normalized effective throughput in bits/symbol. Similar to the reward results in Figure 8, our proposed LSTM-DQN-AM outperformed the other AM methods. After calculation, the throughput of LSTM-DQN-

AM was 5.53% higher than that of DQN-AM, 50.56% higher than that of LSTM-QL-AM and 62.12% higher than that of QL-AM.

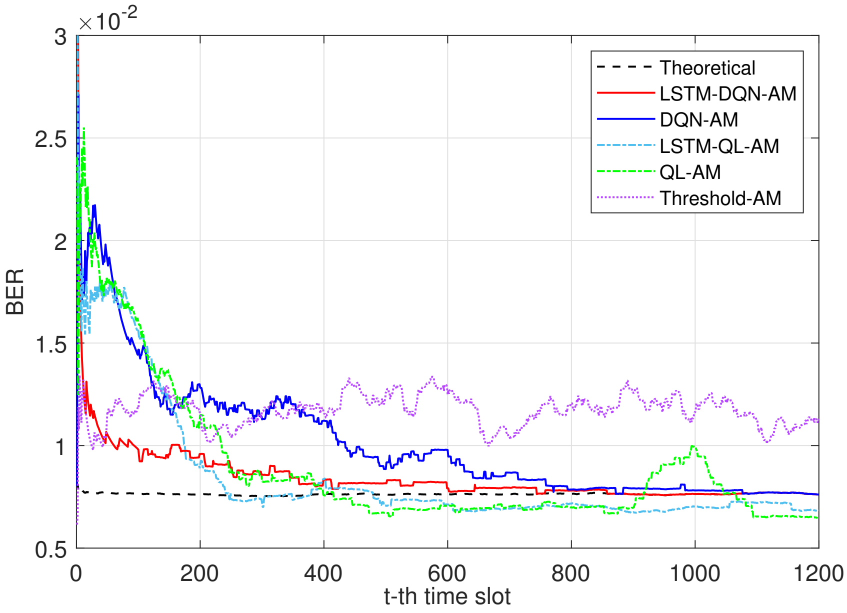


Figure 9. The BER of the AM schemes in the pool experiments.

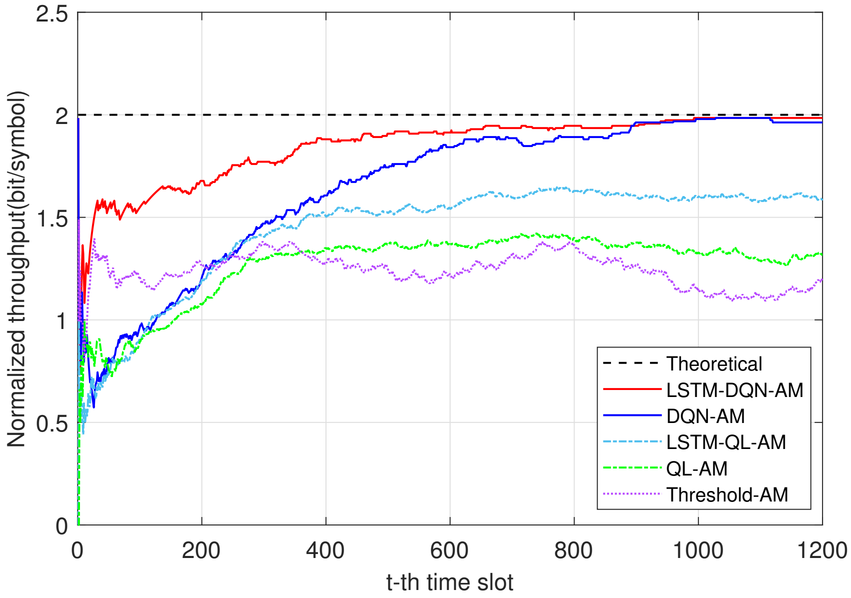


Figure 10. The throughput of the AM methods in the pool experiments.

The results in Figure 10 also prove that the optimization of the rewards in the DQN corresponded to a better throughput performance. The difference was that the rewards had negative values as a penalty factor in RL to encourage the better selection of actions. The throughput values were all positive numbers, so the fluctuations appeared to be relatively small. For example, the reward of Threshold-AM was mostly negative (shown in Figure 8), which was caused by the penalty for transmission failures. For throughput, failed transmissions corresponded to zero bits, so the throughput curves were over 0 and closer to each other.

For a detailed description of the AM reception per time slot, the number of successful bits per symbol in each slot is shown in Figure 11. The x-axis represents the *t*-th time slot and the y-axis represents the successfully received bits per symbol per time slot. Due to the limitations of the experiments, a single-carrier modulation was employed for the AM (0 indicated that the transmission failed). The successful receiving of BPSK, QPSK, 8PSK and 16QAM corresponded to 1,2,3 and 4 bits per symbol, respectively. Note that the data could be transmitted in a packet train using multiple symbols. Again, the normalized expression provided a consistent metric for comparison.

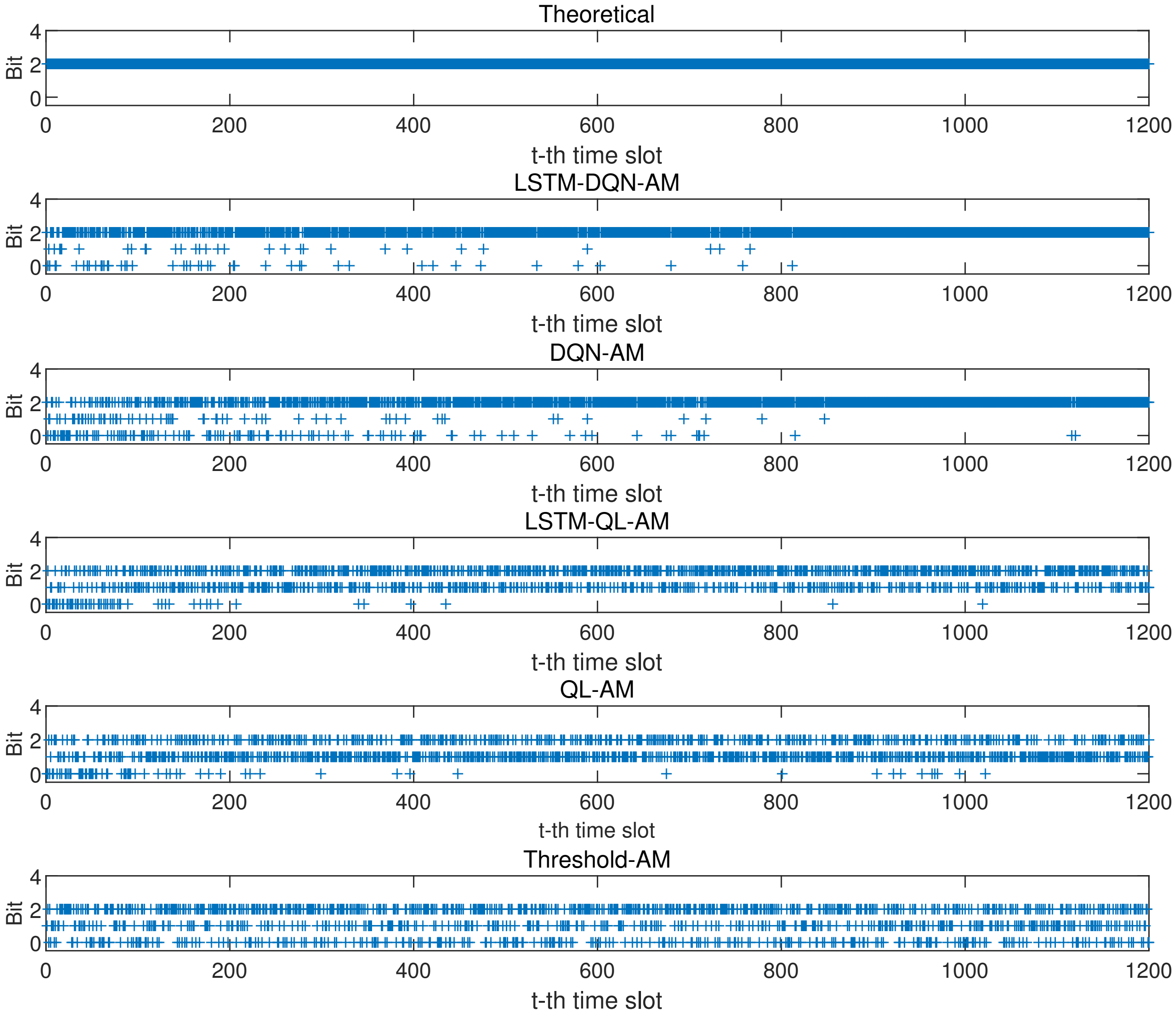


Figure 11. The received number of bits in the AM schemes in the pool experiments.

During the AM mode selection, when the channel conditions were poor, the AM systems selected a low-order modulation method with lower bit rates to maintain reliable communication. On the contrary, in good channel conditions, the systems selected high-order modulation with higher bit rates to improve the system throughput. As shown in Figure 11, the theoretical scheme had no transmission failures and selected the modulation mode with a high bit rate, according to the ideal CSI while guaranteeing the BER. In the early stages, LSTM-DQN-AM converged faster than the other schemes, which corresponded to fewer failures and the more frequent selection of higher-order modulation modes under the BER constraints. After convergence, our proposed LSTM-DQN-AM had no transmission failures. This indicated that LSTM-DQN-AM could better match the time-varying channel states in the pool experiments using adaptive selection.

It can also be observed from Figure 11 that the other adaptive schemes had more failed transmissions. The reason for this was that the transmitters selected higher-order modulation methods with low ESNR values, so the data could not be demodulated successfully. At the same time, compared to LSTM-DQN-AM, the number of high-order modulation modes that were selected by the other AM schemes was small. The reason was that in the case of high ESNR values, the other methods selected relatively low-order modulation methods. Although their selection could meet the BER constraint, it was a waste of the spectrum and resulted in a low throughput.

As can be seen from Figure 11, the threshold-based AM that used outdated CSI produced a large number of failures because the outdated CSI was ineffective for AM decisions. For the learning-based methods, the number of failed transmissions decreased with the iterations of learning after convergence.

4.2. Sea Experimental Results

The sea experiments were conducted in a shallow seawater area in Wuyuan Bay, Xiamen, China. The communication distance was about 640 m and the transmitter and receiver were placed 4 m below the water. As with the pool experiments, the transmitter supported four modulation modes: BPSK, QPSK, 8PSK and 16QAM. The carrier frequency was 20 kHz and the duration of one modulation symbol was 0.5 ms. The one-trip propagation delay was about 850 ms. The recorded data and channel state information were used for the performance evaluation of the AM schemes. Figure 12 shows the estimated channel impulsive responses (CIRs). The structures of the CIRs were more complex and the channel variations were more obvious than in the pool experiments.

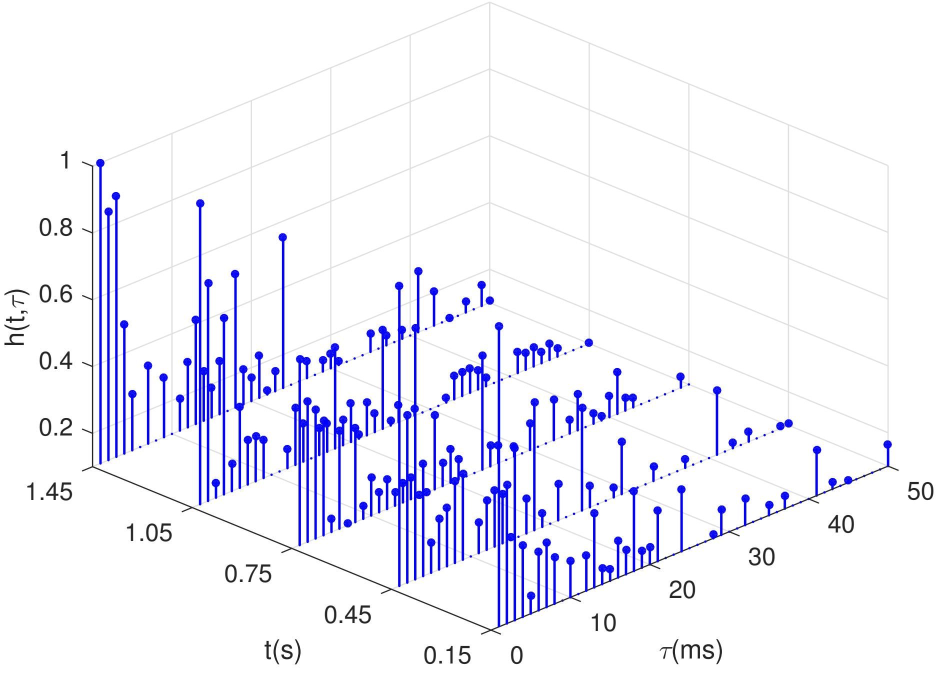


Figure 12. The channel impulsive responses in the sea experiments.

The reward curves of the various AM methods are shown in Figure 13. The proposed LSTM-DQN-AM obtained higher rewards than the other methods, had the best convergence speed and was closest to the theoretical value. The reward curve of LSTM-DQN-AM was also the smoothest after convergence. Similar to the pool experimental results, the reward performance of the various schemes was LSTM-DQN-AM > DQN-AM > LSTM-QL-AM > QL-AM. As shown in Table 3, the long-term reward values of LSTM-DQN-AM and the other methods were compared computationally. The reward of LSTM-DQN-AM was 37% higher than that of DQN-AM [41], 50% higher than that of LSTM-QL-AM [30] and 104% higher than that of QL-AM [27].

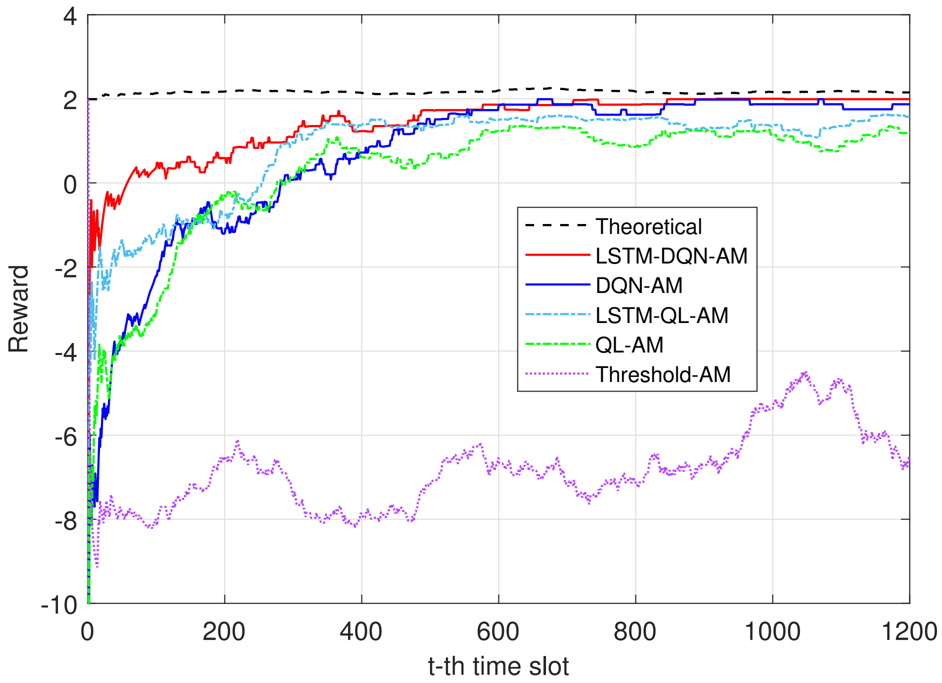


Figure 13. The rewards of the AM schemes in the sea experiments.

Table 3. A comparison of the rewards of the different AM schemes in the sea experiments.

AM Scheme	Compared AM Scheme	Improvement (%)
LSTM-DQN-AM	DQN-AM	37
LSTM-DQN-AM	LSTM-QL-AM	50
LSTM-DQN-AM	QL-AM	104

The BER curves of the various AM methods are shown in Figure 14. The results also demonstrated in the same trend as those from the pool experiments. LSTM-DQN-AM had the fastest convergence speed and the most stable performance. Except for the threshold switching-based AM method, the other intelligent learning algorithms could meet the BER constraints after convergence. In addition, we observed that the BER of the learning-based AM was sometimes even lower than the theoretical value. The theoretical optimum reward calculation was based on perfect channel state information and the selection of the modulation mode with the highest data rate under the BER constraints. The corresponding BER with the selected mode was also calculated. Whereas in the Q-learning process, the transmitter could select a modulation bit level that was too low for a certain SNR. This selection resulted in lower data rates and lower BER values than the optimum solution.

In particular, as can be seen from Figures 13 and 14, the theoretical reward and BER in the sea experiments varied more intensively than those in the pool experiments. This was caused by the complex variations in the sea channels themselves. Although the sea experiments were challenging, the proposed LSTM-DQN-AM method still achieved the best and most robust performance.

Figure 15 shows the throughput performances in the sea experiments. Similar to the pool experimental results, LSTM-DQN-AM had the best throughput performance. LSTM-DQN-AM could select high-order modulation modes and meet the BER constraints using to the feedback CSI, resulting in a high throughput performance. The throughput of LSTM-DQN-AM was 10.5% higher than that of DQN-AM, 26.7% higher than that of LSTM-QL-AM and 45.4% higher than that of QL-AM. The throughput could not reach the same level as the theoretical curve as the variations in the sea channels were much more severe than those in the pool. The proposed LSTM-DQN-AM method achieved a better performance than the other schemes, whether in the pool with relatively slow time variations or in the sea with severe channel variations, and its performance was the closest to the ideal solution.

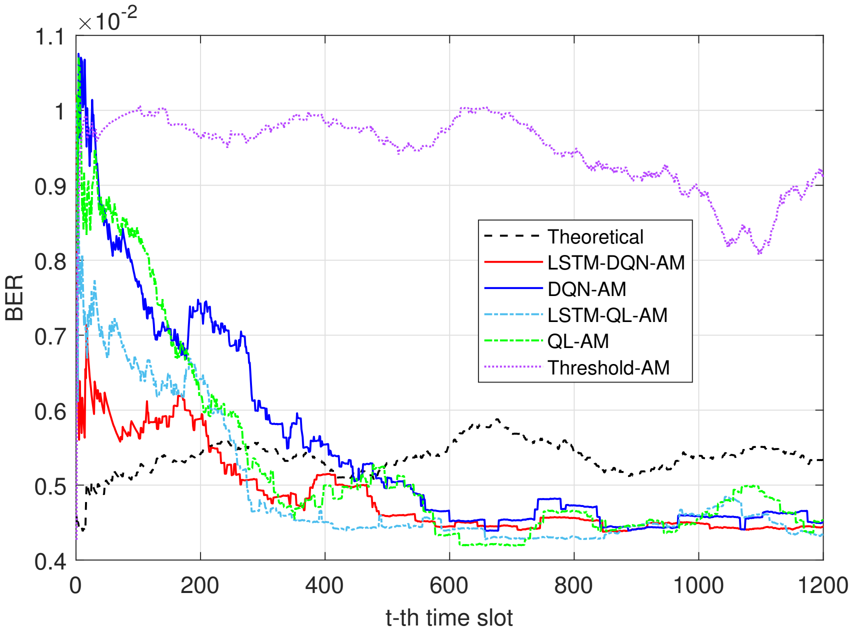


Figure 14. The BER of the AM schemes in the sea experiments.

Figure 16 shows the number of successfully received bits per symbol in each time slot in the sea experiments. The learning result of LSTM-DQN-AM was the closest to the ideal scheme. Compared to the other methods, LSTM-DQN-AM had fewer failures and the more frequent selection of higher-order modulation modes under the BER constraints. After the 700-th time slot, the transmission of LSTM-DQN-AM rarely failed. LSTM-DQN-AM had an excellent learning ability to select the appropriate modulation mode that matched the channel state, so its performance was more stable than those of the other methods.

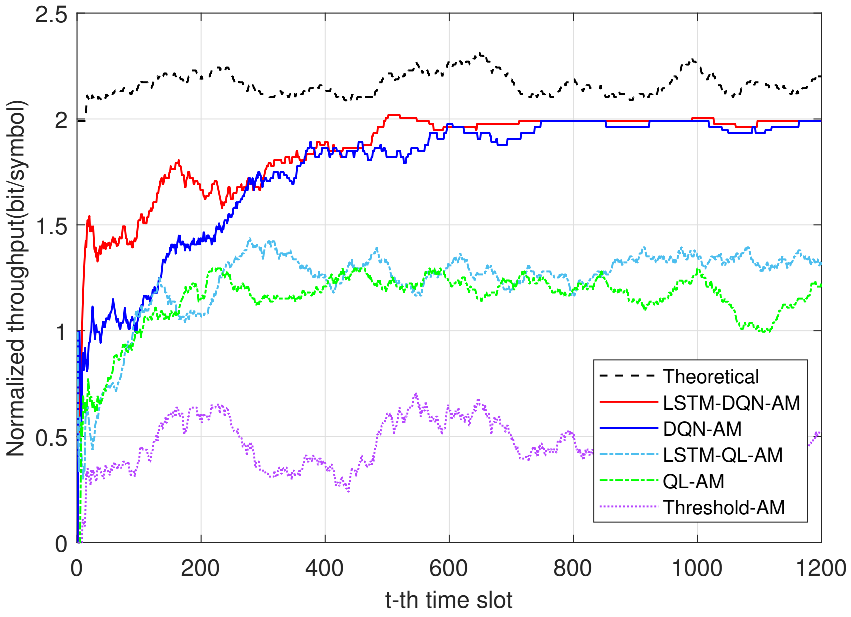


Figure 15. The throughput in the sea experiments.

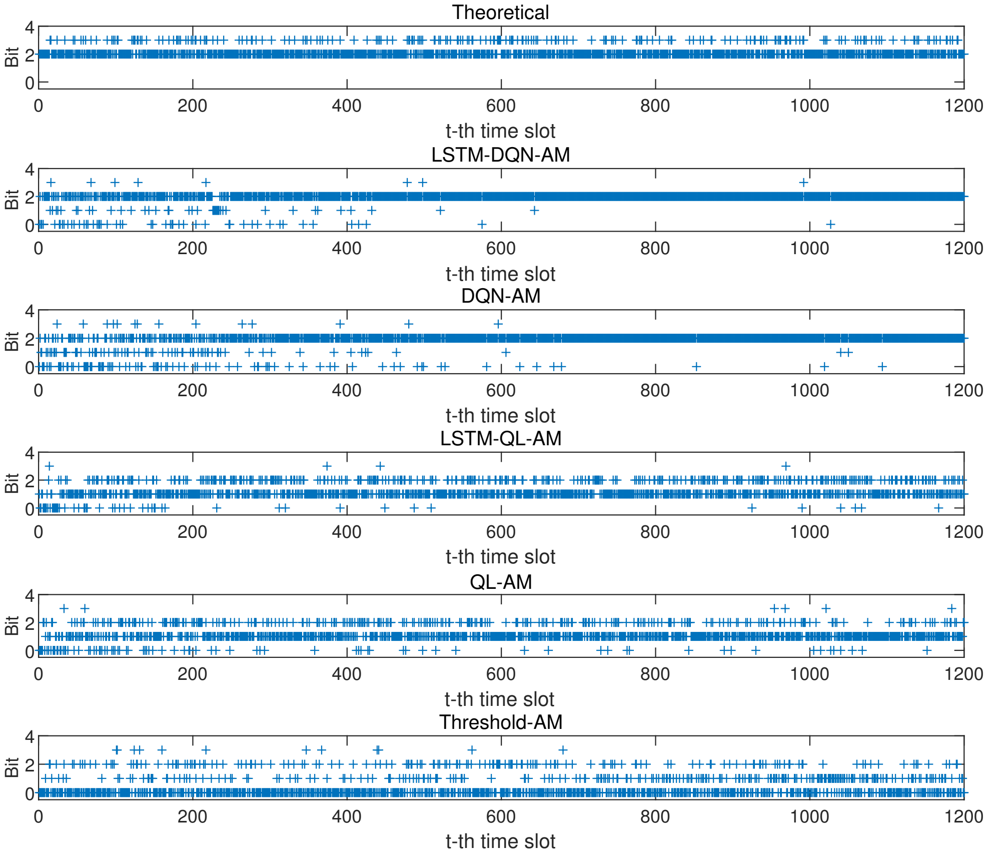


Figure 16. The received number of bits in the AM schemes in the sea experiments.

5. Discussion

In this paper, we proposed an adaptive modulation for UWA, which was based on deep reinforcement learning. The scheme aimed to improve system throughput under BER constraints in UWA channels with time-varying and long propagation delay characteristics. Specifically, a DQN-based AM structure was used to map the relationships between the state-action pairs and rewards by inputting sequential outdated feedback CSI. An LSTM neural network layer was added to the DNN in the DQN structure to enhance its ability to track the historical CSI in order to mitigate the decision bias that was caused by the partially observable CSI. The results and findings of this paper are discussed in the following section.

This paper focused on the adaptive modulation of UWA communication that was based on deep reinforcement learning in order to deal with high-dimensional CSI inputs for more accurate learning results. In the research area of RL-based UWA adaptive modulation, previous investigations [27–30] have only set one single input CSI as the state of reinforcement learning. The use of approach made the construction of Q-tables with smaller state spaces easier. In traditional RL, high-dimensional inputs produce more comprehensive information for learning but suffer from dimensionality. The DQN methods combined a DNN with QL to map the relationships between the state-action pairs and rewards using deep neural networks without the need to build very large Q-tables that are difficult to converge. Furthermore, the dual deep neural network structure, experience replay and dynamic greedy algorithm strategy made the DQN more efficient at dealing with complex problems and achieved higher rewards than QL. From multiple experimental results, it could be seen that the two DQN-based AM methods (LSTM-DQN-AM and DQN-AM) were more robust and could achieve higher rewards than the two QL-based methods (LSTM-QL-AM and QL-AM). The DQN-based AM outperformed QL, as the DNNs had the ability to handle high-dimensional time series. Combined with a DNN, the DQNs offered a potential solution for reinforcement learning when dealing with high-dimensional data.

To enhance the performance of AM in complex UWA environments, this paper proposed LSTM-DQN-AM. Compared to the DQN-AM scheme with an FC DNN that was proposed for wireless radio communication in [40,41], the proposed LSTM-DQN-AM for UWA communication combined LSTM networks. It had a powerful ability to extract channel state information and express the relationships between learning factors and realistic complex environments. In DQN-AM, the relationship mapping of QL adopted a traditional FC DNN, although UWA CSI was only partially observed because it was affected by factors such as channel variations, estimation errors, feedback errors and outdated feedback errors. LSTM had the advantage of selectively remembering input information from both long-term and short-term memory, which could mitigate the decision bias that was caused by the partially observable UWA CSI. A diagram of LSTM-DQN-AM is shown in Figure 4 and the processing of LSTM-DQN is described in Section 3.2.3. The reward results from the pool and sea experiments are shown in Figure 8 and Figure 13, respectively. The results showed that the proposed LSTM-DQN-AM method outperformed DQN-AM [40,41] under multiple channel conditions. The results also indicated that the integration of an LSTM and a DNN enhanced the system's ability to extract information that was hidden within time series data and produced a better and more robust performance than that of an FC DNN. This method could be employed in various UWA application scenarios in complex and varying UWA channels.

In underwater acoustic channels, the outdated feedback CSI does not match that of the actual channel, which severely degrades the performance of AM schemes. One typical approach to address this is channel prediction, which employs a specific model to predict the current CSI using the outdated measurements for further decision-making. LSTM networks have a unique forget gate, input gate and output gate structural design, which can play a role in channel prediction through the input of time series CSI. In LSTM-QL-AM [30], LSTM was employed for channel prediction to predict ESNR values at data transmission times using outdated feedback. The Q-learning algorithm was then used in the AM for mode selection. In this paper, the proposed LSTM-DQN used the advantages of both the DNN and LSTM to predict the Q-values of current time slots from outdated CSI sequences. It jointly played the roles of channel prediction and Q-value mapping. In comparison to the separate processes with their accumulated errors, the joint process of channel prediction and adaptive modulation could achieve global optimization results. From the pool and sea experimental results, it could be seen that the LSTM-DQN-AM method achieved higher and more robust throughput than LSTM-QL-AM, as shown in Figures 10 and 15. Compared to the separate design, the joint optimization of multiple communication modules using deep learning could achieve optimal results.

6. Conclusions

In this paper, UWA AM was investigated by considering outdated and partially available CSI, which is caused by complex channel variations and long propagation delays. The feedback CSI from transmitters usually cannot match the actual CSI in time-varying UWA channels. Therefore, we proposed the LSTM-DQN-AM scheme to train the relationships between channel states and performance and perform optimum mode selection using deep reinforcement learning. The proposed scheme avoided direct decision-making methods that are based on feedback or predicted channels and played the joint roles of channel tracking and decision-making. Through the integration of an LSTM and a DQN, it could capture channel information from CSI sequences. It could effectively learn optimal actions, even when the available CSI was outdated. Both pool and sea experimental results showed that our proposed scheme could improve system throughput while maintaining BER constraints. Compared to the DQN-AM scheme, the proposed LSTM-DQN-AM had a powerful ability to extract channel state information and express the relationships between learning factors and realistic complex UWA environments. In the sea experiments, the throughput of LSTM-DQN-AM was 10.5% higher than that of DQN-AM. Compared to the separate channel prediction scheme, LSTM-DQN-AM performed a joint learning process and produced a 26.7% throughput improvement compared to LSTM-QL-AM in the sea experiments. Compared to QL-AM, LSTM-DQN-AM exploited the advantages of deep neural networks and recurrent neural networks to process high-dimensional CSI, which produced a throughput that was 45.5% higher than that of QL-AM in the sea experiments. After convergence, both the rewards and the BER values of LSTM-DQN-AM were the most stable. In addition, when there were channel fluctuations, the rewards and BER values of LSTM-DQN-AM were still the most robust. In future research, we will extend this approach to a mobile scenario with mixed modulation modes.

Author Contributions: Conceptualization, Y.Z. and H.W.; data curation, Y.Z.; funding acquisition, Y.Z., H.W. and B.W.; investigation, Y.Z. and J.Z.; methodology, X.S. and Y.D.; resources, H.W. and B.W.; software, J.Z.; supervision, Y.Z. and X.S.; validation, J.Z.; writing—original draft preparation, Y.Z.; writing—review and editing, Y.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under grant numbers 61801372, U19B2015, 62031021 and 62001360.

Data Availability Statement: Not Applicable.

Acknowledgments: The authors would thank Wei Su from Xiamen University for sharing the experimental data.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

UWA	Underwater Acoustic
AM	Adaptive Modulation
CSI	Channel State Information
OFDM	Orthogonal Frequency Division Multiplexing
OFDMA	Orthogonal Frequency Division Multiple Access
CNN	Convolutional Neural Network
LSTM	Long Short-Term Memory
ESNR	Effective Signal to Noise Ratio
SNR	Signal to Noise Ratio
ML	Machine Learning
BER	Bit Error Rate
PSK	Phase-Shift Keying
A-kNN	Attention-Aided K-Nearest Neighbor
AMC	Adaptive Modulation and Coding
RL	Reinforcement Learning
DDPG	Depth-Determined Policy Gradient
DRL	Deep Reinforcement Learning
DNN	Deep Neural Network
DQN	Deep Q-Network
MDP	Markov Decision Process
FC	Fully Connected
EvalNet	Evaluation Network
TargetNet	Target Network

References

1. Stojanovic, M. On the relationship between capacity and distance in an underwater acoustic communication channel. *ACM SIGMOBIle Mob. Comput. Commun. Rev.* **2007**, *11*, 34–43. [CrossRef]
2. Song, A.; Stojanovic, M.; Chitre, M. Editorial Underwater Acoustic Communications: Where We Stand and What Is Next? *IEEE J. Ocean. Eng.* **2019**, *44*, 1–6.
3. Stojanovic, M.; Preisig, J. Underwater acoustic communication channels: Propagation models and statistical characterization. *IEEE Commun. Mag.* **2009**, *47*, 84–89. [CrossRef]

4. Van Walree, P.A. Propagation and Scattering Effects in Underwater Acoustic Communication Channels. *IEEE J. Ocean. Eng.* **2013**, *38*, 614–631. [CrossRef]
5. Roudsari, H.M.; Bousquet, J.F.; McIntyre, G. Channel model for wideband time-varying underwater acoustic systems. In Proceedings of the IEEE OCEANS, Aberdeen, UK, 19–22 June 2017; pp. 1–7.
6. Zhang, Y.; Venkatesan, R.; Dobre, O.A.; Li, C. Efficient Estimation and Prediction for Sparse Time-Varying Underwater Acoustic Channels. *IEEE J. Ocean. Eng.* **2020**, *45*, 1112–1125. [CrossRef]
7. Xu, B.; Wang, X.; Guo, Y.; Zhang, J.; Razzaqi, A. A Novel Adaptive Filter for Cooperative Localization Under Time-Varying Delay and Non-Gaussian Noise. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–15. [CrossRef]
8. Radosevic, A.; Ahmed, R.; Duman, T.; Proakis, J.; Stojanovic, M. Adaptive OFDM Modulation for Underwater Acoustic Communications: Design Considerations and Experimental Results. *IEEE J. Ocean. Eng.* **2014**, *39*, 357–370. [CrossRef]
9. Goeckel, D.L. Adaptive coding for time-varying channels using outdated fading estimates. *IEEE Trans. Commun.* **1999**, *47*, 844–855. [CrossRef]
10. Falahati, S.; Svensson, A.; Ekman, T.; Sternad, M. Adaptive Modulation Systems for Predicted Wireless Channels. *IEEE Trans. Commun.* **2004**, *52*, 307–316. [CrossRef]
11. Liu, L.; Cai, L.; Ma, L.; Qiao, G. Channel State Information Prediction for Adaptive Underwater Acoustic Downlink OFDMA System: Deep Neural Networks Based Approach. *IEEE Trans. Veh. Technol.* **2021**, *70*, 9063–9076. [CrossRef]
12. Wan, L.; Zhou, H.; Xu, X.; Huang, Y.; Zhou, S.; Shi, Z.; Cui, J. Adaptive modulation and coding for underwater acoustic OFDM. *IEEE J. Ocean. Eng.* **2015**, *40*, 327–336. [CrossRef]
13. Qiao, G.; Xiao, Y.; Wan, L.; Guo, X.; Jia, H. Analysis of SNR Metrics for a Typical Underwater Acoustic OFDM System. *IEEE Access* **2019**, *7*, 183565–183579. [CrossRef]
14. Zhang, R.; Ma, X.; Wang, D.; Yuan, F.; Cheng, E. Adaptive Coding and Bit-Power Loading Algorithms for Underwater Acoustic Transmissions. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 5798–5811. [CrossRef]
15. Zhang, Y.; Huang, Y.; Wan, L.; Zhou, S.; Shen, X.; Wang, H. Adaptive OFDMA with Partial CSI for Downlink Underwater Acoustic Communications. *J. Commun. Netw.* **2016**, *18*, 387–396. [CrossRef]
16. Qiao, G.; Liu, L.; Ma, L.; Yin, Y. Adaptive Downlink OFDMA System With Low-Overhead and Limited Feedback in Time-Varying Underwater Acoustic Channel. *IEEE Access* **2019**, *7*, 12729–12741. [CrossRef]
17. Pelekanakis, K.; Cazzanti, L.; Zappa, G.; Alves, J. Decision tree-based adaptive modulation for underwater acoustic communications. In Proceedings of the IEEE Ucomms, Lerici, Italy, 30 August–1 September 2016; pp. 1–5.
18. Huang, L.; Zhang, Q.; Zhang, L.; Shi, J.; Zhang, L. Efficiency Enhancement for Underwater Adaptive Modulation and Coding Systems: Via Sparse Principal Component Analysis. *IEEE Commun. Lett.* **2020**, *24*, 1808–1811. [CrossRef]
19. Huang, J.; Diamant, R. Adaptive Modulation for Long-Range Underwater Acoustic Communication. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 6844–6857. [CrossRef]
20. Huang, L.; Zhang, Q.; Tan, W.; Wang, Y.; Zhang, L.; He, C.; Tian, Z. Adaptive modulation and coding in underwater acoustic communications: A machine learning perspective. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 1–25. [CrossRef]
21. Bhopale, P.; Kazi, F.; Singh, N. Reinforcement Learning Based Obstacle Avoidance for Autonomous Underwater Vehicle. *J. Mar. Sci. Appl.* **2019**, *18*, 228–238. [CrossRef]
22. Jin, Z.; Zhao, Q.; Su, Y. RCAR: A Reinforcement-Learning-Based Routing Protocol for Congestion-Avoided Underwater Acoustic Sensor Networks. *IEEE Sens. J.* **2019**, *19*, 10881–10891. [CrossRef]
23. Valerio, V.; Presti, F.; Petrioli, C.; Picari, L.; Spaccini, D.; Basagni, S. CARMA: Channel-Aware Reinforcement Learning-Based Multi-Path Adaptive Routing for Underwater Wireless Sensor Networks. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 2634–2647. [CrossRef]
24. Yan, J.; Gong, Y.; Chen, C.; Luo, X.; Guan, X. AUV-Aided Localization for Internet of Underwater Things: A Reinforcement-Learning-Based Method. *IEEE Internet Things J.* **2020**, *7*, 9728–9746. [CrossRef]
25. Xiao, L.; Jiang, D.; Chen, Y.; Su, W.; Tang, Y. Reinforcement-Learning-Based Relay Mobility and Power Allocation for Underwater Sensor Networks Against Jamming. *IEEE J. Ocean. Eng.* **2020**, *45*, 1148–1156. [CrossRef]
26. Wang, C.; Wang, Z.; Sun, W.; Fuhrmann, D. Reinforcement learning-based adaptive transmission in time-varying underwater acoustic channels. *IEEE Access* **2018**, *6*, 2541–2558. [CrossRef]
27. Fu, Q.; Song, A. Adaptive modulation for underwater acoustic communications based on reinforcement learning. In Proceedings of the IEEE OCEANS, Charleston, SC, USA, 22–25 October 2018; pp. 1–8.
28. Su, W.; Lin, J.; Chen, K.; Xiao, L.; En, C. Reinforcement learning-based adaptive modulation and coding for efficient underwater communications. *IEEE Access* **2019**, *7*, 67539–67550. [CrossRef]
29. Su, W.; Tao, J.; Pei, Y.; You, X.; Xiao, L.; Cheng, E. Reinforcement Learning Based Efficient Underwater Image Communication. *IEEE Commun. Lett.* **2020**, *25*, 883–886. [CrossRef]
30. Zhang, Y.; Zhu, J.; Liu, Y.; Wang, B. Underwater Acoustic Adaptive Modulation with Reinforcement Learning and Channel Prediction. In Proceedings of the ACM WUWNet'21, New York, NY, USA, 22–24 November 2021; pp. 1–2.
31. Fan, C.; Wang, Z. Adaptive Switching for Multimodal Underwater Acoustic Communications Based on Reinforcement Learning. In Proceedings of the ACM WUWNet'21, New York, NY, USA, 22–24 November 2021; pp. 1–2.
32. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.; Veness, J.; Bellemare, M.; Graves, A.; Riedmiller, M.; Fidjeland, A.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]

33. Arulkumaran, K.; Deisenroth, M.; Brundage, M.; Bharath, A. Deep Reinforcement Learning: A Brief Survey. *IEEE Signal Process. Mag.* **2017**, *34*, 26–38. [CrossRef]
34. Ye, X.; Yu, Y.; Fu, L. Deep Reinforcement Learning Based MAC Protocol for Underwater Acoustic Networks. *IEEE Trans. Mob. Comput.* **2022**, *21*, 1625–1638. [CrossRef]
35. Liu, E.; He, R.; Chen, X.; Yu, C. Deep Reinforcement Learning Based Optical and Acoustic Dual Channel Multiple Access in Heterogeneous Underwater Sensor Networks. *Sensors* **2022**, *22*, 1628. [CrossRef]
36. Cao, X.; Sun, C.; Yan, M. Target Search Control of AUV in Underwater Environment with Deep Reinforcement Learning. *IEEE Access* **2019**, *7*, 96549–96559. [CrossRef]
37. Liu, T.; Hu, Y.; Xu, H. Deep Reinforcement Learning for Vectored Thruster Autonomous Underwater Vehicle Control. *Complexity* **2021**, *2021*, 1–25. [CrossRef]
38. Li, M.; Zhao, X.; Liang, H.; Hu, F. Deep Reinforcement Learning Optimal Transmission Policy for Communication Systems With Energy Harvesting and Adaptive MQAM. *IEEE Trans. Veh. Technol.* **2019**, *68*, 5782–5793. [CrossRef]
39. Zhang, L.; Tan, J.; Liang, Y.; Feng, G.; Niyato D. Deep reinforcement learning-based modulation and coding scheme selection in cognitive heterogeneous networks. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 3281–3294. [CrossRef]
40. Lee, D.; Sun, Y.G.; Kim, S.H.; Sim, I.; Hwang, Y.M.; Shin, Y.; Kim, D.I.; Kim, J.Y. DQN-Based Adaptive Modulation Scheme over Wireless Communication Channels. *IEEE Commun. Lett.* **2020**, *24*, 1289–1293. [CrossRef]
41. Mashhadi, S.; Ghiasi, N.; Farahm, S.; Razavizadeh, S.M. Deep Reinforcement Learning Based Adaptive Modulation with Outdated CSI. *IEEE Commun. Lett.* **2021**, *25*, 3291–3295. [CrossRef]
42. Gers, F.; Schraudolph, N.; Schmidhuber, J. Learning Precise Timing with LSTM Recurrent Networks. *J. Mach. Learn. Res.* **2002**, *3*, 115–143.
43. Li, J.; Mohamed, A.; Zweig, G.; Gong, Y. LSTM time and frequency recurrence for automatic speech recognition. In Proceedings of the IEEE Workshop ASRU, Scottsdale, AZ, USA, 13–17 December 2015; pp. 187–191.
44. Zhao, M.; Yan, L.; Chen, J. LSTM-DNN Based Autoencoder Network for Nonlinear Hyperspectral Image Unmixing. *IEEE J. Sel. Top. Signal Process.* **2021**, *15*, 295–309. [CrossRef]
45. Yan, N.; Huang, S.; Kong, C. Reinforcement Learning-Based Autonomous Navigation and Obstacle Avoidance for USVs under Partially Observable Conditions. *Math. Probl. Eng.* **2021**, *2021*, 5519033. [CrossRef]
46. Liu, S.; Bai, Y. UAV Intelligent Coverage Navigation Based on DRL in Complex Geometrical Environments. *Int. J. Comput. Intell. Syst.* **2021**, *14*, 177. [CrossRef]
47. Wang, X.; Liu, A.; Zhang, Y.; Xue, F. Underwater Acoustic Target Recognition: A Combination of Multi-Dimensional Fusion Features and Modified Deep Neural Network. *Remote Sens.* **2019**, *16*, 1888. [CrossRef]
48. Zhao, Y.; Wang, M.; Xue, H.; Gong, Y.; Qiu, B. Prediction Method of Underwater Acoustic Transmission Loss Based on Deep Belief Net Neural Network. *Appl. Sci.* **2021**, *11*, 4896. [CrossRef]