Adaptive Modulation and Coding with Feedback Scheduling for an Underwater Acoustic Link

Wu Shuangshuang[1], Mandar Chitre[2], Prasad Anjangi[3]

[1]ARL, Tropical Marine Science Institute, National University of Singapore

[2]Department of Electrical and Computer Engineering & ARL, Tropical Marine Science Institute, National University of Singapore

[3]Halliburton, Singapore

Abstract

Underwater acoustic channels exhibit significant temporal and spatial variability, making it challenging to design a single communication scheme that works well everywhere and at all times. Adaptive Modulation and Coding (AMC) techniques offer a solution by dynamically selecting the optimal Modulation and Coding Scheme (MCS) for specific channel conditions but require an accurate model to predict communication performance. We propose a Bit Error Rate (BER) estimation model that fuses domain knowledge to aid the evaluation of MCSs. In complex sea conditions, we enhance the reliability of AMC by extending our BER prediction model from a point prediction to an interval predictor. This extension involves incorporating Gaussian Process Regression (GPR) to address the uncertainty in BER. Predictions from such an algorithm are used to drive AMC to maximize communication throughput reliably. For AMC, regular feedback from the receiver to the transmitter is necessary to gather Channel State Information (CSI). On the one hand, obtaining feedback too often reduces the communication throughput in channels with long propagation delays, but on the other hand, insufficient feedback leads to suboptimal AMC decisions and hence poor throughput. We propose an algorithm that integrates Tree Search and Deep Q-Network (DQN) for feedback scheduling to automatically find the right balance and optimize communication performance. We demonstrate the advantages of our algorithm through experiments in a test tank and at sea in Singapore. Furthermore, our algorithm also exhibited reliability and achieved optimal throughput in various underwater environments in simulation.

Index Terms

Underwater Acoustic Communication, Adaptive Modulation and Coding, Feedback Scheduling, Bit Error Rate Estimation, Tree Search, Deep Q-Network

I. Introduction

The Underwater Acoustic Communication (UAC) channels present challenges due to their limited bandwidth, huge propagation delays, and frequency-selective fading effects [1]. High-speed communication in UAC is crucial for various industries such as marine operations, offshore oil and gas, and defense applications. However, the dynamic characteristics of UAC channels make it impractical to find a single communication scheme that maintains robust performance in the long-term deployment of UAC systems [2]. As a result, there is a growing demand for Adaptive Modulation and Coding (AMC) techniques, which enable the selection of Modulation and Coding Schemes (MCSs) based on the current channel conditions to achieve both reliability and high throughput [3]–[6].

Significant progress has been made in the development of AMC techniques in wireless communication. Among these studies, data-driven AMC algorithms have gained prominence due to their simple input requirements, capability for various problems with limited knowledge about underlying physics, and ability to extract insights from datasets. For example, in [7], [8], supervised learning strategies aided in the SNR estimation of different MCSs and enabled AMC operation accordingly. Studies [9], [10] utilized Machine Learning (ML) tools to find out the relationship between the channel measurements and Bit Error Rate (BER) performance or SNR to make MCSs decisions based on the predicted Channel State Information (CSI). Work in [11] classified the channels into different types and identified the best MCSs for each channel type in long-range UAC. However, data-driven AMC algorithms, as exemplified in existing works, require substantial training data sets to cover diverse channel conditions. Collecting such extensive data in underwater environments is time-consuming and contradicts the goal of transmitting large files within the shortest possible time.

On the contrary, physics-informed methods, which incorporate knowledge about the underlying channel physics, can be an alternative to data-driven methods for operating AMC. An illustrative example from [3] revealed how the sparse structure of the channel impulse response can be harnessed to enhance AMC while reducing computational demands and memory overhead. In [12], channel physics information in an Orthogonal Frequency-Division Multiplexing (OFDM) system helped narrow down the MCSs space and build correlations among MCSs, fostering faster convergence speed for the AMC model. Inspired by [12], our previous work in [13] proposed a heuristic BER estimation model based on channel physics knowledge

in an OFDM system that enhanced AMC performance even when dealing with a high-dimensional MCSs space. These studies underscore the potential of channel physics-informed approaches for facilitating the adaptability and performance of AMC in UAC systems.

CSI is a fundamental component of AMC, enabling the UAC system to dynamically tune MCSs based on the current channel conditions [3], [14], [15]. The speed of sound in water is approximately 1500 m/s, resulting in propagation delays that are $200,000\times$ higher than those experienced in terrestrial radio communication networks [16]. These propagation delays are comparable to the typical frame duration in UAC. Extensive research has addressed the negative effects of large propagation delays, impacting handshaking protocols and retransmission schemes [17], as well as medium-access control layer protocols preventing data collisions [18]. In a one-to-one communication system, where data frames are exchanged between a transmitter node and a receiver node, the transmitter awaits feedback from the receiver node regarding CSI before performing AMC and initiating frame transmission. In such scenarios, the introduction of two-way propagation delays can substantially degrade channel throughput. To our knowledge, the research on dynamically scheduling feedback to optimize channel throughput is still relatively limited. We propose a feedback scheduling strategy and determine relevant decision parameters to address the trade-off between communication performance and resource utilization.

AMC in any communication system, including UAC, necessitates evaluation metrics like BER [19], [20], data rate [21]–[23], or throughput [24], [25]. These metrics evaluate MCSs across variable channel conditions, thereby informing the AMC choice. Throughput assessment is vital in optimizing data transmission rates among these metrics, especially when accounting for two-way propagation delays [24], [25]. Thus, we adopt throughput to evaluate our AMC and feedback scheduling performance. Given the propagation distance between the transmitter node and receiver node, MCSs with higher coded data rates tend to achieve higher channel throughput. The coded data rate comprises uncoded data rate and error correction overhead. BER knowledge of MCSs aids in the selection of appropriate error correction methods [26]–[28], like Forward Error Correction (FEC) [29], [30]. Therefore, we consider the problem of estimating BER to enhance AMC to achieve optimal throughput in UAC systems. However, the time-varying behavior of UAC channels introduces significant fluctuations in the actual BER. Consequently, a BER distribution predictor is needed to provide a range of possible BER values given any modulation configuration for reliable MCSs determination. Researchers have proposed several models and techniques to estimate the BER in wireless communication systems, like the empirical BER model [31], statistical methods in [32]–

[34] via assuming a specific distribution prior or Monte Carlo error count [35], [36]. Recently, ML-based approaches have become popular which employ algorithms like Gaussian Process Regression (GPR) [37], [38], Neural Networks (NN) [39], or support vector machines [40] to estimate BER. They aim to learn the complex relationships between input parameters (such as transmission parameters, channel conditions, and noise levels) and the corresponding BER. Usually, ML is applied in a purely data-driven manner that relies on the availability and quality of data. With channel physics knowledge incorporated, a BER estimation model is proposed in [13] which relaxes the demand for the data availability.

In our prior study [13], we modeled the sequential MCS decision and feedback scheduling as a Markov Decision Process (MDP). We used throughput over multiple frames as the reward to train a NN that predicted throughput for any timings of feedback reception. However, when optimizing throughput in the long-term transmission, this NN might be sub-optimal since it leans towards immediate rewards. A look-ahead tree can potentially optimize long-term rewards, but its time-consuming construction limits its real-time applicability [41]. Merging tree search frameworks with Deep Learning (DL) and Reinforcement Learning (RL) has emerged as a prominent approach for real-time optimization in planning and scheduling tasks [42]. Specifically, [43] and [44] both present integration of tree search with RL. In the former, RL aids in the development of value and policy functions within the tree structure, while the latter uses tree search to guide RL exploration in intricate tasks. As such, we underscore the significance of incorporating RL methodologies with tree search configurations to resolve MDP challenges, particularly in scenarios with extensive action or state spaces, to optimize long-term rewards.

The rest of the paper is organized as follows. The problem formulation is elaborated in Section II. The proposed AMC strategy is described in Section III. In Section IV, details of a dynamic feedback scheduling strategy are explained. Then, we test our algorithm in a test tank and sea trials and demonstrate the advantage of our method in Section V. In Section VI, we show the advantages of our algorithm via simulations in diverse channel environments. We then draw our conclusions and consider the possibilities for future works in Section VII. The acronyms and symbols used are listed in Table I and Table II, respectively.

Notation: Bold symbols and $(\cdot)$ denote vectors. Symbols in a calligraphic font like $\mathcal{A}$ denote tuples. Symbols in $\{\cdot\}$ denote sets. We use the interval notation $[a, b) = \{x \in \mathbb{Z} | a \leq x < b\}$. $|\mathcal{A}|$ denotes the size or cardinality of a tuple $\mathcal{A}$. The symbol $\equiv$ represents equivalence. The symbol $\lceil a \rceil$ represents the smallest integer greater than or equal to a.

TABLE I: LIST OF ACRONYMS.

Acronym	Description
AMC	Adaptive Modulation and Coding
BER	Bit Error Rate
CSI	Channel State Information
DQN	Deep Q-Network
FHBFSK	Frequency-Hopping Binary Frequency Shift Keying
FRI	Feedback Report Interval
FSK	Frequency Shift Keying
GPR	Gaussian Process Regression
LDPC	Low-Density Parity Check
MAE	Mean Absolute Error
MCS	Modulation and Coding Scheme
MCTS	Monte Carlo Tree Search
MDP	Markov Decision Process
ML	Machine Learning
NN	Neural Network
OFDM	Orthogonal Frequency Division Multiplexing
PSK	Phase-Shift Keying
QAD	Quantile Absolute Deviation
QAM	Quadrature Amplitude Modulation
RL	Reinforcement Learning
RX	Receiver
TX	Transmitter
UAC	Underwater Acoustic Communication

II. PROBLEM FORMULATION

A. Problem Overview

Consider an UAC system where information frames are exchanged between a transmitter (TX) node and a receiver (RX) node. The objective is to transmit N bits from the TX node to the remote RX node located at a distance l within the shortest possible time over time-varying UAC channels. Modulation and Coding techniques are used to encode these bits onto frames for reliable communication. There are multiple variants of modulation schemes in UAC systems. For example, the modulation schemes can include phase, frequency, or amplitude modulation, such as Phase-Shift Keying (PSK), Frequency Shift Keying (FSK), or Quadrature Amplitude Modulation (QAM). Or in an OFDM system, the modulation schemes can represent various OFDM parameters, such as the number of subcarriers, the cyclic prefix length, etc. After modulation, the coding techniques, like the FEC, add redundant bits to the modulated frames, allowing the RX node to detect and correct errors. Due to the variability in UAC channels, it is hard

TABLE II: LIST OF SYMBOLS.

Symbol	Description
j	Index of state
$\boldsymbol{a_j}$	Modulation scheme st state $\boldsymbol{s_j}$
$\mathcal{A}$	Modulation scheme space
$d(\boldsymbol{a_j})$	Uncoded data rate of $\boldsymbol{a_j}$
h_j	Number of frames in the j^{th} FRI
$\mathcal{H}$	Set of possible FRI values
k_j^1	Number of frames for which "test" mode is enabled
k_j^2	Number of frames for which "test" mode is disabled
k_j	Ratio: $\frac{k_j^1}{k_j^1+k_j^2}$
l	Distance between TX node and RX node
n_j'	Percentage of transmitted bits
N	Total number of bits to be transmitted
r_j	Throughput of the j^{th} FRI
$\boldsymbol{s_j}$	State where the CSI of the j^{th} FRI is updated
$\mathcal{S}$	State space
τ_j	Transmission duration of each frame in the j^{th} FRI
τ_{m}	Duration of frames containing modulation information
τ_{fd}	Duration of frames containing feedback information
τ_{pd}	Propagation delay between the TX and RX nodes
$\boldsymbol{\epsilon_j}(\boldsymbol{a_j})$	Measured BER during the j^{th} FRI
$\hat{\epsilon}_j(\boldsymbol{a_j})$	Estimated BER at state $\boldsymbol{s_j}$
$\eta_j(\cdot)$	Regression analysis model for QAD prediction
$\rho(\hat{\epsilon}_j(\boldsymbol{a_j}))$	FEC rate selected based on estimated BER $\hat{\epsilon}_j(\boldsymbol{a_j})$
$\boldsymbol{\varrho}$	Set of available FEC rates
$\boldsymbol{\theta_j}$	Weight parameters for median prediction from BER distribution
$\boldsymbol{\omega_j}$	Weight parameters of FRI determination model
$\zeta(\cdot)$	Model for median prediction from BER distribution

to design a single modulation scheme that works well in all situations. Therefore, the demand for AMC techniques arises, enabling tuning the modulation scheme and coding strategy based on current channel conditions. We establish a DATA link between the TX node and RX node to transmit the modulated and coded frames, with the intent of providing as high a data rate as feasible. However, optimal communication performance in various environmental conditions necessitates fine-tuning this DATA link.

Successful transmission is achieved only when the TX and RX nodes employ identical modulation schemes. When the modulation schemes are determined at the TX node, a crucial task is to inform the remote RX node about the modulation information reliably before the DATA link frames are exchanged. A separate communication link, referred to as the CONTROL link, is first established. The CONTROL link exhibits robust communication albeit at a lower data rate than the DATA link, and the modulation and error correction parameters of the CONTROL link are pre-determined. The modulation information for the DATA link is then encoded onto frames and transmitted over this CONTROL link to the remote RX node.

Performing AMC heavily relies on obtaining accurate CSI. The CSI, such as measured BER based on the number of bits corrected during FEC decoding, is acquired through feedback from the RX node. However, employing modulation schemes and coding rates blindly may lead to failed frame receptions at the RX node. Although such failures indicate that the BER exceeds a certain threshold, they hinder acquiring accurate BER for reliable AMC. To address this challenge, a "test" mode is introduced, where frames carrying known bits are transmitted over the DATA link. In this mode, the BER can be accurately computed as the transmitted frames are known, and the CSI is updated. When the "test" mode is disabled, the N unknown bits are encoded and transmitted to the RX node over the DATA link. In this case, the BER is estimated after demodulation and decoding of the frames at the RX node. All CSI, including BER measurements, are then encoded onto frames and sent back to the TX node via the CONTROL link, thereby improving the performance of AMC.

Given the possibly long propagation delay between frames exchanged over DATA and CONTROL links, tuning modulation schemes and awaiting feedback for each frame consumes time. Conversely, employing a modulation scheme across multiple frames without timely feedback can lead to suboptimal performance, resulting in a loss of received frames at the RX node and reduced throughput. This motivates the consideration of the optimal timings for tuning modulation schemes and waiting for feedback to optimize the channel throughput. Therefore, a Feedback Report Interval (FRI), which decides the number

of transmission frames over the DATA link between two consecutive feedbacks, is proposed. In the j^{th} FRI, a specific number of frames (h_j), are transmitted using the same modulation scheme $\boldsymbol{a_j}$ and its corresponding coding rate.

B. Mathematical Formulation

We formulate the sequential decision-making of modulation scheme $\boldsymbol{a_j}$ and FRI h_j as well as the subsequent interaction with the environment to receive feedback as a MDP. In this MDP, $\mathcal{A}$ is a set containing all possible modulation schemes, i.e., $\boldsymbol{a_j} \in \mathcal{A}$ and $\mathcal{H}$ is another set including all possible values of h_j, i.e., $h_j \in \mathcal{H}$. The action space now has a cardinality of $|\mathcal{A} \times \mathcal{H}|$. An intelligent decision-making algorithm, known as the agent, engages in iterative interactions with the environment to learn and optimize a policy denoted as Π. This policy guides the agent for selecting actions from action space $|\mathcal{A} \times \mathcal{H}|$ to transmit N bits within the possible shortest time. In the state space $\mathcal{S}$, a state $\boldsymbol{s_{j+1}}$ is reached upon receiving the feedback of the j^{th} FRI. It encompasses the completion ratio of N bits, knowledge related to decision-making of actions, and communication performance metrics. The state transitions from state $\boldsymbol{s_j}$ to state $\boldsymbol{s_{j+1}}$ follows a transition function Γ. Therefore, Π is a function that maps the state space to the action space, i.e., $\Pi : \mathcal{S} \rightarrow |\mathcal{A} \times \mathcal{H}|$. After successfully transmitting all N bits, the agent reaches the terminal state $\boldsymbol{s_J}$ and hence $j \in [1, J]$. The round-trip frame exchange duration comprises the transmission duration of h_j frames, i.e., $h_j\tau_j$, the duration of frames containing modulation information τ_{m}, a two-way propagation delay $2\tau_{\text{pd}}$, and the duration of frames containing feedback τ_{fd}, as shown in Fig. 1.

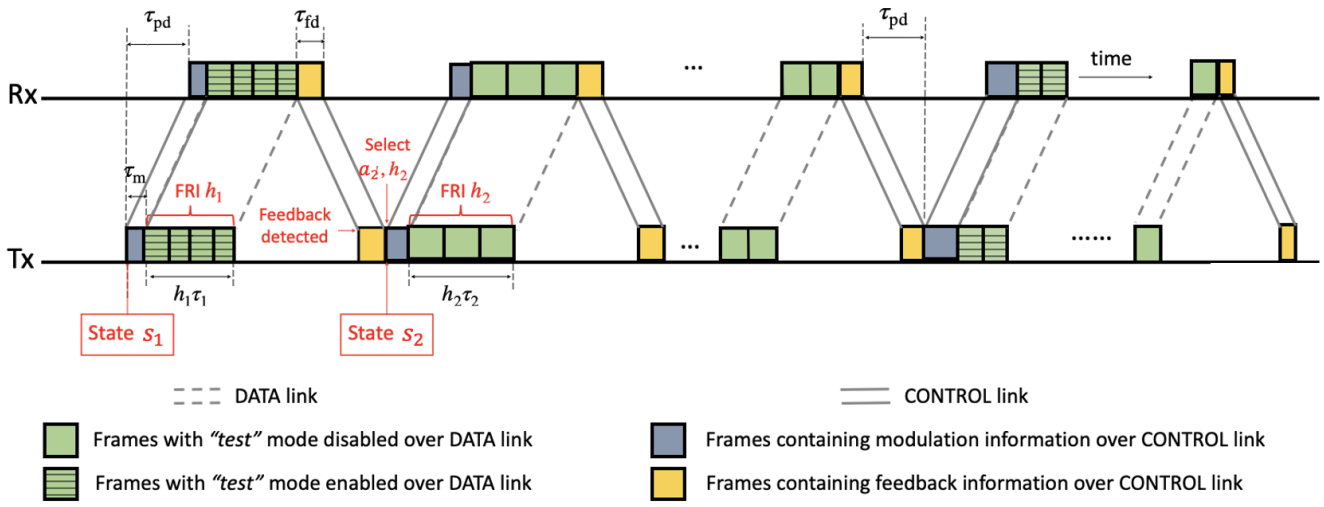

Fig. 1: An illustration of the delays in frame exchange between the TX and RX nodes.

The throughput over the transmission is the performance metric for selecting the actions in our MDP. When modulation scheme $\boldsymbol{a_j}$ is selected, the coding technique, such as FEC, is then applied. The FEC adds redundant bits to the transmitted data frames, facilitating error detection and correction during transmission. A set $\boldsymbol{\varrho}$ contains available FEC rates associated with their respective affordable BER levels. With knowledge of the BER statistics $\epsilon(\boldsymbol{a_j})$, the agent can easily determine the optimal FEC rate $\rho(\epsilon(\boldsymbol{a_j}))$. Specifically, if no FEC rate in $\boldsymbol{\varrho}$ is available for correcting the BER $\epsilon(\boldsymbol{a_j})$, the "test" mode is enabled, in which the pre-determined bits are transmitted to aid BER estimation. Given the uncoded data rate $d(\boldsymbol{a_j})$ of the modulation scheme $\boldsymbol{a_j}$, the coded data rate is calculated as $d(\boldsymbol{a_j})\rho(\epsilon(\boldsymbol{a_j}))$. The coded data rate in a communication system is closely correlated to the throughput and thus serves as a valuable metric for enhancing channel throughput.

Knowledge of BER is crucial for calculating the coded data rate in communication systems. However, in time-varying UAC channels, measuring accurate BER can be challenging, particularly given a possibly large size of $\mathcal{A}$. Obtaining an estimation of BER $\epsilon(\boldsymbol{a_j})$ over the action space $\mathcal{A}$ is hence required. A heuristic BER model based on channel physics knowledge from [13] estimates the median of the BER statistics $\zeta(\boldsymbol{a_j};\boldsymbol{\theta_j})$. $\boldsymbol{\theta_j}$ represents the model weight parameters and are updated given the feedback of the latest CSI. However, it may not capture the worst-case performance due to the inherent uncertainty of BER measurement. To ensure robust modulation selection in adverse channel conditions, η_j is proposed to help estimate the difference between the median prediction and the upperbound of the unknown BER distribution. The BER upperbound $\hat{\epsilon}_j(\boldsymbol{a_j})$ is given by

$$\hat{\epsilon}_j(\boldsymbol{a_j}) = \zeta(\boldsymbol{a_j};\boldsymbol{\theta_j}) + \eta_j(\boldsymbol{a_j}). \tag{1}$$

Obtaining CSI through feedback from the RX node plays a vital role in facilitating accurate BER estimation and subsequent AMC. However, the feedback process consumes time due to significant propagation delays between frames exchange. Therefore, the selection of FRI involves a tradeoff between the optimization speed of the AMC strategy and maximizing the throughput of transmitting N bits. When the transmission result of the last FRI is poor, choosing a smaller value for the next FRI enables faster convergence of the BER estimation model by updating the CSI more frequently. However, this may lead to increased latency due to propagation delays. On the other hand, selecting a larger FRI does not always guarantee improved channel throughput. In a varying UAC channel, selecting a larger FRI would mean that the model would operate far from the optimum and hence result in poorer performance. Thus, a dynamic

approach is required to determine the optimal sequence of FRI that balances these two objectives.

As depicted in Fig. 1, the system transitions from $\boldsymbol{s_{j-1}}$ to $\boldsymbol{s_j}$ upon the completion of the $(j-1)^{\text{th}}$ FRI. Within the state transition, the measured BER $\boldsymbol{\epsilon_{j-1}}(\boldsymbol{a_{j-1}})$ from the updated CSI is utilized to train (1). The ratio of N bits that have been transmitted till state $\boldsymbol{s_j}$, denoted by a percentage value n'_j, along with the timestamps of frames exchange is recorded. n'_j serves as a dynamic indicator of how much of the transmission task has been completed, guiding the agent to adapt its strategy by selecting shorter or longer intervals based on the remaining bits or urgency. The number of frames with and without the "test" mode enabled up to state $\boldsymbol{s_j}$ are respectively tracked by k_j^1 and k_j^2. The throughput r_{j-1} of FRI h_{j-1} is calculated as the measure of $(n'_j - n'_{j-1})N$ bits transmitted over the DATA link within a time period encompassing h_{j-1} frames with a transmission duration of τ_{j-1} each, along with the duration of frames containing modulation information τ_{m}, the feedback frame duration τ_{fd}, and a two-way propagation delay of $2\tau_{\text{pd}}$, i.e.,

$$r_{j-1} = \frac{(n'_j - n'_{j-1})N}{h_{j-1}\tau_{j-1} + 2\tau_{\text{pd}} + \tau_{\text{fd}} + \tau_{\text{m}}}. \tag{2}$$

At state $\boldsymbol{s_j}$, The parameters n'_j, r_{j-1}, and the ratio $k_j = \frac{k_j^1}{k_j^1 + k_j^2}$ provide valuable insights into the communication performance under current policy Π. The throughput r_{j-1} of FRI h_{j-1} evaluates the performance of the last selected interval, offering the agent direct feedback on the effectiveness of its prior decision and helping refine its policy for future selections. Additionally, with the inclusion of k_j into the state, the agent gains a measure of how much robust, validated information has been gathered, reflecting the reliability of the system's current understanding of the environment. To determine the next FRI h_j as a function of n'_j, r_{j-1}, and k_j, we utilize ML techniques since such a function is analytically unknown. We construct a model $\mathcal{M}(\cdot)$ with inputs n'_j, k_j, and r_{j-1} to predict the values of all possible $h_j \in \mathcal{H}$. The optimal h_j is determined by

$$h_j = \underset{h_j \in \mathcal{H}}{\operatorname{argmax}}\, \mathcal{M}\left(\boldsymbol{a_j}, h_j; \left\{n'_j, k_j, r_{j-1}\right\}, \boldsymbol{\omega_j}\right), \tag{3}$$

where $\boldsymbol{\omega_j}$ denotes the parameters of $\mathcal{M}(\cdot)$, which are updated once CSI is received. The detail of state transition between $\boldsymbol{s_{j-1}}$ and $\boldsymbol{s_j}$ is

$$\begin{aligned} \boldsymbol{s_j} &= \Gamma(\boldsymbol{s_{j-1}}, \boldsymbol{a_{j-1}}, h_{j-1}) \\ &= \{\boldsymbol{\theta_j}, n'_j, \boldsymbol{\omega_j}, \eta_j, k_j, r_{j-1}\}. \end{aligned} \tag{4}$$

To summarize, our objective is to choose the sequence of modulation schemes $\boldsymbol{a_j}$ and FRI h_j, $j = [1, J]$, where J is unknown, to transmit N bits within the shortest time and thereby maximize the throughput:

$$\begin{aligned} &\underset{\boldsymbol{a_1}, h_1, \boldsymbol{a_2}, h_2, \cdots, \boldsymbol{a_J}, h_J}{\operatorname{argmin}} \left(\sum_{i=1}^{J} h_i \tau_i + J(\tau_{\text{fd}} + 2\tau_{\text{pd}} + \tau_{\text{m}}) \right) \\ &s.t. \quad n'_J \geq 1. \end{aligned} \tag{5}$$

III. Adaptive Modulation and Coding

In this section, we delve into the adaptation strategy of MCSs, focusing on OFDM given its prevalence in modern underwater modems. Leveraging channel physics information, we construct a heuristic BER estimation model to guide AMC strategy selection. We validate this model using various datasets and offer a reference table for BER-based FEC rate selection. Once the coded data rate for each modulation scheme is ascertained, we suggest a dynamic ϵ-greedy policy to address the exploration-exploitation dilemma in modulation scheme selection given the high-dimensional MCSs space.

A. Modulation Scheme

We perform AMC in OFDM. In OFDM, there are two critical parameters: the cyclic prefix length n_{p} and the number of subcarriers n_{c}. The cyclic prefix length is required for mitigating intersymbol interference caused by the multipath effect. Meanwhile, the number of subcarriers determines the potential for each to undergo flat fading relative to the channel's coherence bandwidth, and must also adhere to constraints imposed by the channel coherence time. Bandwidth B also plays a significant role in communication performance. A wider bandwidth possibly enables higher data rates but may also affect SNR. Therefore, n_{c}, n_{p}, and B at the TX and RX nodes need to be tuned to optimize communication performance. A modulation scheme $\boldsymbol{a_j} \equiv (n_{\text{c}}, n_{\text{p}}, B)$ can be represented as a point in the modulation scheme region $\mathcal{A}$. For an OFDM system with Quadrature PSK on each subcarrier, the uncoded data rate $d(\boldsymbol{a_j})$ is

$$d(\boldsymbol{a_j}) = \frac{2Bn_{\text{c}}}{n_{\text{c}} + n_{\text{p}}}. \tag{6}$$

B. BER Estimation Model

BER information plays a pivotal role in establishing the appropriate code rate and computing the coded data rate, thereby offering valuable insights into the selection of modulation scheme $\boldsymbol{a_j}$.

1) Estimation of BER Median: In [13], a heuristic BER estimation model $\zeta(\cdot)$ based on UAC channel physics is introduced. This model defines three boundary planes, namely Bc_1, Bc_2, and Bc_3, which divide the $(n_\mathrm{c}, n_\mathrm{p}, B)$ space into different spaces. As described in [13], for good channel performance, the bandwidth of each subcarrier should remain below the coherence bandwidth to achieve flat fading, the cyclic prefix length n_p should be longer than the channel delay spread τ_ds, and the symbol duration T_s must be less than the channel coherence time τ_c to maintain channel stability throughout the symbol's duration, i.e.,

$$n_\mathrm{c} > \frac{B\tau_\mathrm{ds}}{0.423} = Bc_1, \tag{7}$$

$$n_\mathrm{p} > B\tau_\mathrm{ds} = Bc_2, \tag{8}$$

$$n_\mathrm{c} + n_\mathrm{p} < B\tau_\mathrm{c} = Bc_3. \tag{9}$$

The behavior of these boundaries with respect to a specific bandwidth B is illustrated in Fig. 2 which is reproduced from [12]. Within the triangle region, modulation schemes are more aligned with the physics constraints, leading to a potentially reduced BER. Conversely, schemes outside this region often do not satisfy these constraints, increasing the likelihood of surpassing error correction capabilities and consequent frame loss [45]. The position and size of the triangle region vary accordingly as B changes. A sigmoid function $s(d) = \frac{1}{1+e^{-b_i d}}$, $i = 1, 2, 3$ is built to characterize the BER information based on the relative position of the point $(n_\mathrm{c}, n_\mathrm{p}, B)$ with respect to the three boundaries Bc_i, $i = 1, 2, 3$. The values b_i, $i = 1, 2, 3$ correspond to the slope of the three sigmoid functions. The parametric model $\zeta(\cdot)$ is utilized to estimate the median value of the selected $\boldsymbol{a_j}$'s BER distribution at state $\boldsymbol{s_j}$ and is built as

$$\zeta(\boldsymbol{a_j}; \boldsymbol{\theta_j}) = (b_4 B + c_4) s(-d_1) s(-d_2) s(d_3), \tag{10}$$

$$d_1 = n_\mathrm{c} - Bc_1, \tag{11}$$

$$d_2 = n_\mathrm{p} - Bc_2, \tag{12}$$

$$d_3 = \frac{n_\mathrm{c} + n_\mathrm{p} - Bc_3}{\sqrt{2}}, \tag{13}$$

where d_1, d_2, d_3 are distances as shown in Fig. 2. Additionally, b_4 and c_4 represent the slope and intercept of the linear term with respect to B. At state $\boldsymbol{s_j}$, these weight parameters $\boldsymbol{\theta_j} \equiv (c_1, c_2, c_3, c_4, b_1, b_2, b_3, b_4)$

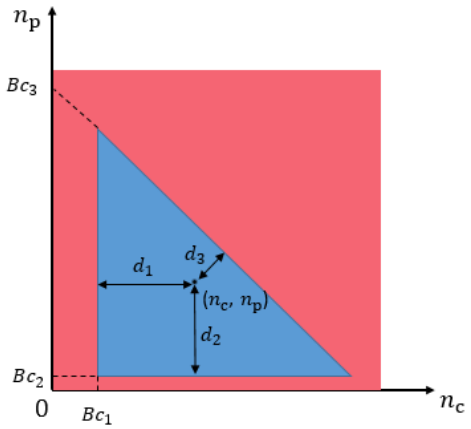

Fig. 2: Visualization of the boundaries Bc_1, Bc_2 and Bc_3 in the (n_c, n_p, B) space at a given B [12].

are updated based on the updated BER measurement over transmission.

To enhance the accuracy of the model, it is vital to measure how closely the model's estimates align with actual BER values. Hence, we introduce a loss function $L(\boldsymbol{\theta_j})$ for training $\zeta(\cdot)$. The loss function evaluates the Mean Absolute Error (MAE) between the output of $\zeta(\boldsymbol{a_i}; \boldsymbol{\theta_j})$ and the measured BER $\boldsymbol{\epsilon_i}(\boldsymbol{a_i})$, $i = [1, j]$, i.e.,

$$L(\boldsymbol{\theta_j}) = \frac{1}{|j|} \sum_{i=1}^{j} (|\zeta(\boldsymbol{a_i}; \boldsymbol{\theta_j}) - \boldsymbol{\epsilon_i}(\boldsymbol{a_i})|). \tag{14}$$

Through the minimization of $L(\boldsymbol{\theta_j})$ at state $\boldsymbol{s_j}$ using techniques like gradient descent, the model's weight parameters, $\boldsymbol{\theta_j}$, of our model, are refined, enhancing its BER estimation during transmission. This iterative refinement utilizes measured BER data, ensuring the model's predictions remain closely aligned with empirical observations.

2) Estimation of BER Uncertainty: The work presented in [13] demonstrates the capability of $\zeta(\cdot)$ for estimating the median from the time-varying BER distribution. It also illustrates that BER tends to cluster around this median value although variability is observed within the actual BER distribution. To enhance the reliability of the modulation scheme selection in view of the BER uncertainty, evaluating the upperbound in the BER distribution becomes necessary. Using the maximum BER to select a conservative FEC maximizes reliability, potentially at the cost of data throughput. We employ the Quantile Absolute Deviation (QAD) method [46] to help estimate the BER upperbound.

The QAD entails computing the q^{th} quantile of the absolute difference between the predicted median BER given by $\zeta(\boldsymbol{a_j}; \boldsymbol{\theta_j})$ and the measured BER $\boldsymbol{\epsilon_j}(\boldsymbol{a_j})$ from feedback. The training set $\mathcal{D}_j$ to train $\eta_j(\cdot)$

is composed of:

$$\{\boldsymbol{a_i} \rightarrow QAD(|\boldsymbol{\epsilon_i}(\boldsymbol{a_i}) - \zeta(\boldsymbol{a_i}; \boldsymbol{\theta_j})|; q)\}, \quad i \in [1, j], \tag{15}$$

which includes attempted modulation schemes $\boldsymbol{a_i}$, $i \in [1, j]$ and their corresponding QAD values during previous transmissions. To construct $\mathcal{D}_j$, we retain only the latest transmission's BER for each configuration, acknowledging the non-stationary nature of the underwater acoustic channel. For each unique $\boldsymbol{a}_i$, the absolute deviation between its most recent measured BER and the corresponding predicted median is used to compute the q^{th} percentile, which serves as its QAD label. As a result, $\mathcal{D}_j$ forms a table mapping each tried modulation scheme to its estimated QAD. Upon each state transition, $\eta_j(\cdot)$ is retrained on the incrementally updated $\mathcal{D}_j$ using the latest BER feedback. This enables QAD estimation for any candidate $\boldsymbol{a_{j+1}} \in \mathcal{A}$ at state $\boldsymbol{s_{j+1}}$. The choice of q in the QAD calculation (15) is set to 75 which guarantees that at least 75% of transmitted frames are successfully delivered, as it encompasses the range within which 75% of the actual BER values reside. It allows for a degree of error tolerance and also considers the trade-off between frame loss and data throughput.

Performing an exhaustive search over all possible modulation schemes in $\mathcal{A}$ and storing their corresponding QAD values is time-consuming. However, BER uncertainty tends to be highly correlated across modulation schemes that share similar characteristics in n_{c}, n_{p}, or B. Modulation schemes with the same n_{c} value tend to exhibit comparable levels of frequency diversity, and modulation schemes with similar n_{p} values would experience similar levels of protection against intersymbol interference. Similar bandwidth B values often encounter comparable channel conditions and noise levels. The QAD estimator η_j utilizes GPR [47] to learn these correlations within $\mathcal{A}$ based on the observed QAD values up to state $\boldsymbol{s_j}$. This correlation enables QAD predictions for all potential modulation schemes, eliminating the need for exhaustive evaluation.

A GPR model includes a crucial component known as the mean function, which establishes a prior expectation of the general trend in the predicted QAD. The mean of all the previously observed QAD values is used as the mean function, denoted as μ_{QAD}. Another essential component is the kernel function, which determines the similarity between data points and governs the smoothness and behavior of the GPR model. In our approach, we employ the Matérn kernel [48], known for its flexibility in modeling different levels of smoothness, to define the correlation and smoothness of the estimated QAD. The kernel function is defined as $K(\cdot)$, where its two arguments represent any two modulation schemes from the training set $\mathcal{D}_j$. The Matérn kernel has two key parameters: the length scale ℓ and the smoothness parameter ν.

The length scale parameter ℓ determines the range over which data points influence each other. A small ℓ confines the influence of a modulation scheme $\boldsymbol{a_i}$, $i = [1, j]$ to a narrow $(n_{\mathrm{c}}, n_{\mathrm{p}}, B)$ space, resulting in rapid changes in the GPR function over short distances. Conversely, a large length scale allows a modulation scheme to have a significant influence on other schemes that are farther apart, even over long distances. The smoothness parameter ν controls the flexibility of the Matérn kernel in capturing complex patterns. Higher values of ν lead to smoother functions, while lower values introduce more roughness and allow for intricate variations in the modeled functions. Euclidean distance is applied to calculate the distance between modulation schemes $(n_{\mathrm{c}}, n_{\mathrm{p}}, B)$ and $(n'_{\mathrm{c}}, n'_{\mathrm{p}}, B')$. We choose $\ell = 0.3$, approximated using the distance between the nearest neighbor modulation schemes in $\mathcal{A}$. To determine an appropriate value of ν, a range of commonly used values, namely $\{0.5, 1.5, 2.5\}$, is visualized on datasets collected from a tank and Singapore water which we list in subsection III-C. $\nu = 1.5$ is finally selected as it achieves superior performance by striking a suitable balance between demonstrating regularity in BER and allowing for fluctuations between neighboring modulation schemes.

The QAD estimator $\eta_j(\cdot)$ predicts QAD for any possible $\boldsymbol{a_j} \in \mathcal{A}$ at state $\boldsymbol{s_j}$ follows

$$\eta_j(\boldsymbol{a_j}) \sim \mathcal{GP}(\mu_{\mathrm{QAD}}, K(\cdot)). \tag{16}$$

C. Data Sets

Two datasets contain the measured BER statistics when tuning n_{c}, n_{p}, and B using Subnero M25M modems in different water environments. The Subnero M25M modem operating bandwidth is up to 12 kHz, i.e., from 20 to 32 kHz. The details of these datasets are provided below.

- A dataset collected from a test tank shown in Fig. 3. The transmission distance l between the TX and RX nodes was about 1.5 m, and the depth of TX and RX modems was 1.5 m. The BER was measured for $(n_{\mathrm{c}}, n_{\mathrm{p}}, B)$, where n_{c} was set to different values from the set $\{128, 256, 512, 1024, 2048, 4096, 8192\}$ and n_{p} ranged from 0 to 8192, B was set to different values from $\{2.4, 4.8, 7.2, 9.6, 12, 14.4, 16.8\}$ kHz.
- A dataset, denoted as SEADATA, was collected by colleagues from the Acoustic Research Laboratory in Singapore waters, with the experiment as setup from [49]. The transmission range between the TX and RX nodes, i.e., the node B and node C in Fig. 4 was about 600 m, and the water depth was between 10 and 20 m. The BER was measured for $(n_{\mathrm{c}}, n_{\mathrm{p}}, B)$, where n_{c} was set to different values

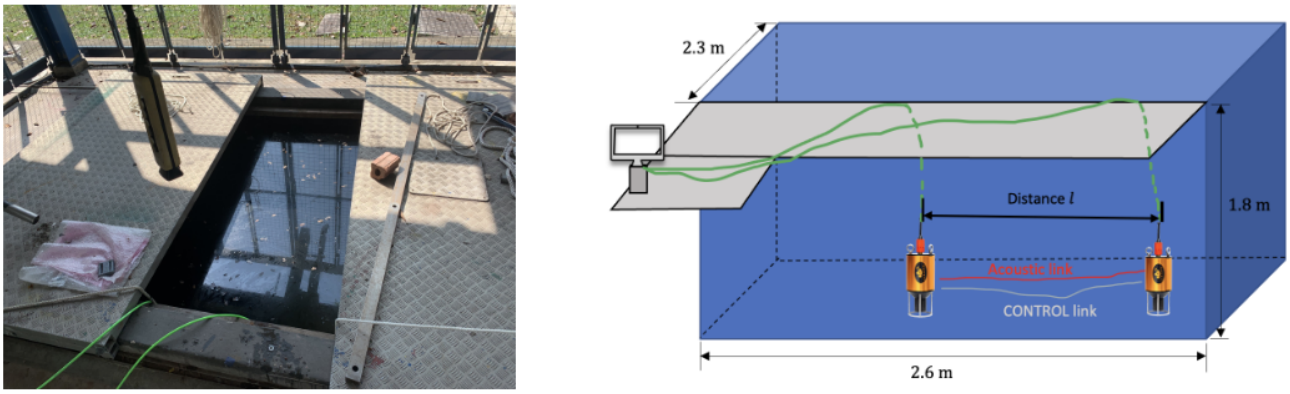

Fig. 3: Test tank and deployment of modems in the tank.

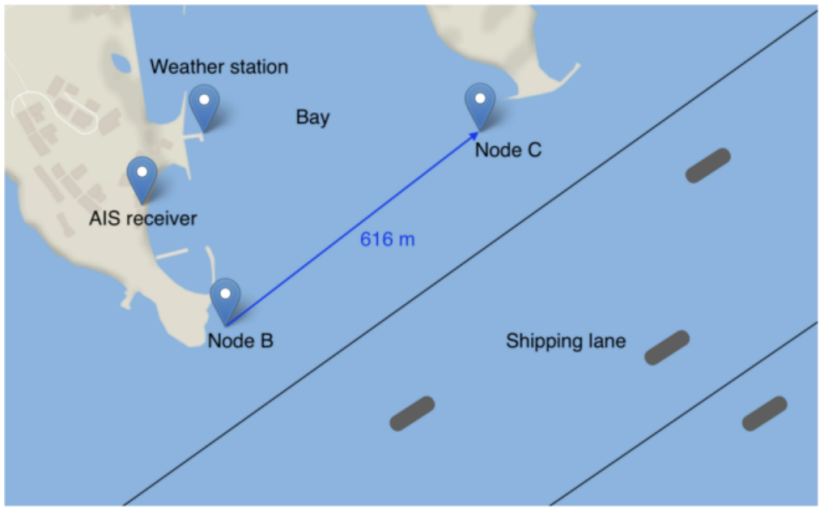

Fig. 4: Experiment setup for collecting SEADATA [49].

from the set $\{64, 128, 256, 512, 1024, 2048\}$, n_{p} ranged from 0 to 2046, B was set to different values from $\{4.8, 7.2, 9.6, 10.8\}$ kHz.

D. BER Estimation Model Validation

1) Validation on a Large Data Set: The BER estimation model is validated on the dataset SEADATA. To assess the model, SEADATA was split into a training set (SEATRAIN) and a test set (SEATEST) using a $7:3$ ratio.

Fig. 5 compares the measured BER with the estimated BER obtained from the proposed model (1) trained on SEATRAIN. To reduce redundancy, the results are presented for each n_{c}, with n_{p} binned into groups of size 128 due to similar BER values between adjacent n_{p} values. Compared to the findings reported in [13], the proposed model demonstrates better responsiveness to changes in n_{p} and greater

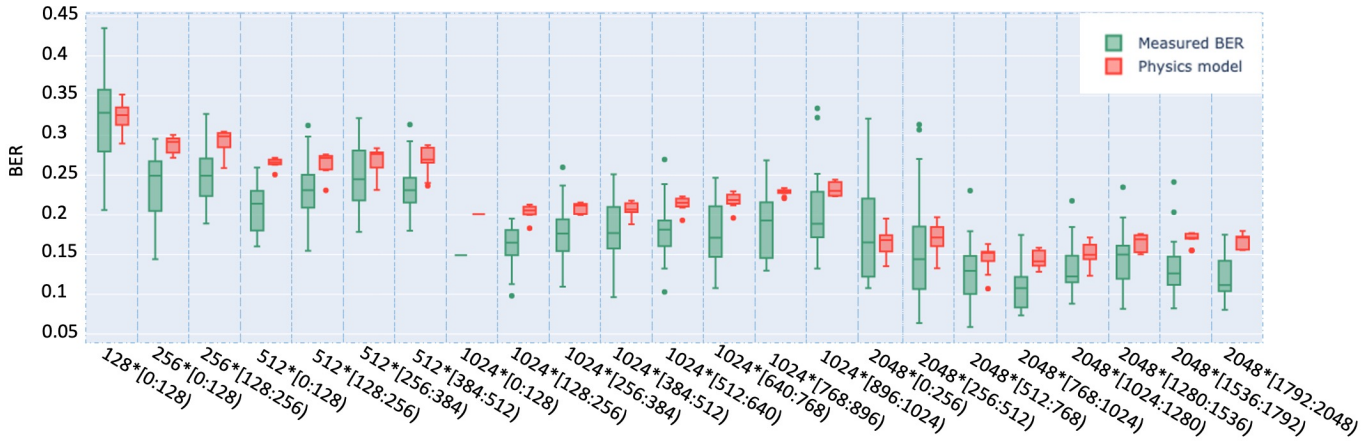

Fig. 5: Comparison of the measured BER from SEATEST (the left/green boxplots) and the BER upperbound estimation (the right/red boxplots).

Notes: The x-axis labels correspond to the values of n_{c} and a range of n_{p}, for example, $128 * [0:128) \rightarrow \{n_{\mathrm{c}} = 128, n_{\mathrm{p}} \in [0:128)\}$.

accuracy in capturing BER uncertainties.

2) Validation on a Small Data Set: During the early stage of transmission, channel BER knowledge is absent and is acquired during transmission. At the beginning of the transmission, modulation scheme and FEC rate choices should prioritize a high frame success rate to ensure dependable CSI feedback, which in turn aids the training of the BER estimation model and enhances channel throughput in the long term. A small data set is available at the beginning of transmission in realistic cold-start scenarios. Our validation on a small data set thus aims to show the capacity of our BER estimation model to guide the selection of robust modulation scheme from $\mathcal{M}$ and FEC rate with fewer training data or transmissions than the pure data-driven methods. To demonstrate its superiority, only five pairs of $(n_{\mathrm{c}}, n_{\mathrm{p}}, B)$ and its corresponding measured BER sampled from the SEATRAIN dataset randomly are utilized as the training set.

The GPR is employed as a purely data-driven comparison method, denoted as $\mathcal{GP}'$ to easily distinguish it from the previous QAD estimator $\mathcal{GP}$. The prior mean function for $\mathcal{GP}'$, μ_ϵ, is the mean of the five samples. Assuming QAD and BER estimation share identical correlation functions over $\mathcal{A}$, we employ the same Matérn kernel in the QAD estimator $\mathcal{GP}$ for $\mathcal{GP}'$. The Euclidean distance between modulation schemes is applied. This purely data-driven GPR method assumes the estimated BER $\epsilon_{\mathrm{gp}}(\boldsymbol{a_j})$ for $\boldsymbol{a_j} \in \mathcal{A}$ follows the $\mathcal{GP}'$ after trained on the five samples, i.e.,

$$\epsilon_{\mathrm{gp}}(\boldsymbol{a_j}) \sim \mathcal{GP}'(\mu_\epsilon, K(\cdot)), \tag{17}$$

where the two arguments are any two modulation schemes from the five samples. For our physics-informed

TABLE III: AN EXAMPLE OF A 5 DATA POINTS TRAINING SET.

Row	Number of Subcarriers n_{c}	Cyclic Prefix Length n_{p}	Bandwidth B (kHz)	BER
1	2048	47	10.8	0.177
2	64	59	10.8	0.333
3	1024	211	10.8	0.173
4	2048	168	7.2	0.150
5	1024	28	7.2	0.093

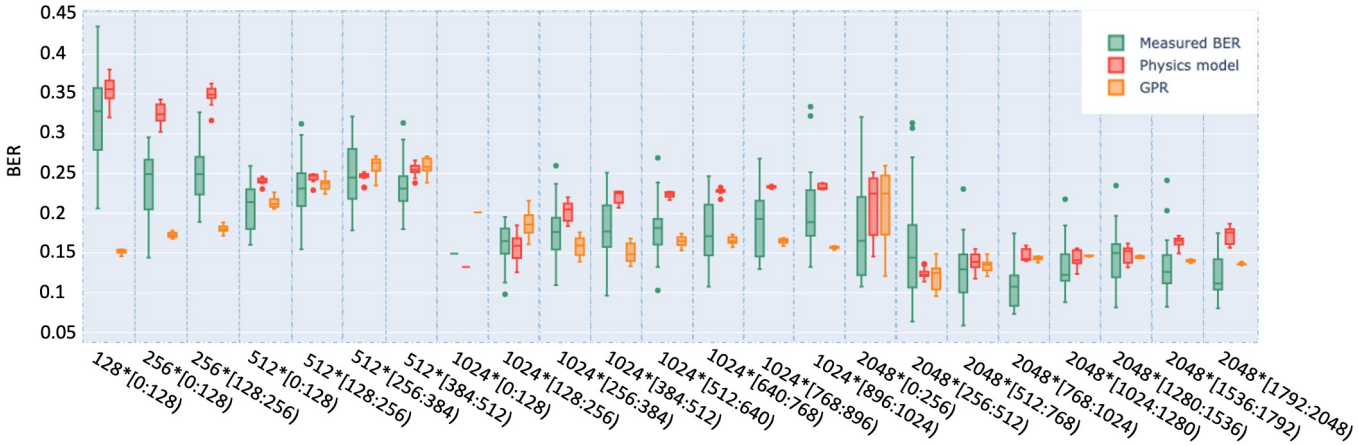

Fig. 6: Comparison of the measured BER of 5 samples in SEATEST (the left/green boxplots), the estimated BER upperbound by our physics-informed model (the middle/red boxplots) and estimated BER $\epsilon_{\mathrm{gp}'}(\cdot)$ via a pure data-driven GPR model $\mathcal{GP}'$ (the right/orange boxplots) on 5 data points training set.

Notes: The x-axis labels correspond to the values of n_{c} and a range of n_{p}, for example, $128 * [0:128) \rightarrow \{n_{\mathrm{c}} = 128, n_{\mathrm{p}} \in [0:128)\}$.

model $\zeta(\cdot) + \eta_j(\cdot)$, we first train $\zeta(\cdot)$ on the five samples from SEATRAIN dataset for 1000 epochs to approximate the BER median values over $\mathcal{A}$. $\eta_j(\cdot)$, parameters of which is aligned with (16), is then employed to estimate QAD values and help BER upperbound estimation. The comparison between our physics-informed model $\zeta(\cdot) + \eta_j(\cdot)$ and this purely data-driven GPR model $\mathcal{GP}'$ is performed on the SEATEST set. Table III and Fig. 6 illustrate an example of the training set and the comparison results for one five-sample set.

The results are presented in Fig. 6 where the measured BER (left, green), the predictions from our physics-informed model (middle, red), and the purely data-driven GPR model (right, orange), are displayed together for each modulation configuration. The advantage of our proposed physics-informed model is demonstrated when channel knowledge is limited, as indicated by the middle (red) boxplots. Compared to the right (orange) boxplots generated by $\mathcal{GP}'$, our model aims to approximate the potential upperbound of the time-varying BER distribution. Although the BER estimates provided by our model may be higher than the measured upperbound for modulation schemes, this conservative approach allows for the

selection of a more robust FEC rate during early-stage communication with incomplete channel knowledge. Consequently, a better frame success rate can be achieved while still maintaining an acceptable compromise in terms of data rate. The selection of the FEC rate leads to reliable CSI feedback for BER estimation model training initially. In contrast, the right (orange) boxplots generated by the purely data-driven GPR model $\mathcal{GP}'$ tend to underestimate the BER and are clustered around the lowerbound. This tendency leads to an overconfident selection of the FEC rate, resulting in a higher probability of frame loss. Initiating transmission with an excessively low frame success rate can hinder the convergence speed of the BER estimation model, subsequently impacting the enhancement of long-term channel throughput.

Despite being trained on only five samples, our model maintains robustness due to its physics-informed design. The base BER predictor $\zeta(\cdot)$ embeds prior knowledge to capture overall BER trends, while the QAD estimator $\eta_j(\cdot)$ adaptively bounds uncertainties in a conservative manner. This enables effective modulation decisions even under severe data scarcity, demonstrating strong cold-start capability.

E. Forward Error Correction

Given the estimated BER upperbound, we use low-density parity-check (LDPC) code [50] as the FEC technique. To choose an appropriate LDPC rate, Table IV obtained via simulations is consulted. In simulation, three different block size frames with $18, 432, 1450$ bytes are used and errors are introduced manually. We then employ 6 different LDPC rates from $\{\frac{2}{3}, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \frac{1}{6}\}$ to decode that three different block size frames with error included and tested the maximal LDPC rate for a certain BER level with 90% frame success rate. For BER values less than 0.18, they showed the 6 LDPC rates mentioned previously were capable of correcting the errors. When none of the BER ranges specified in Table IV are met, we enable the "test" mode, and known bits are sent from the TX node to the RX node.

F. Exploration & Exploitation in AMC

During the process of sequential decision-making on modulation schemes $\boldsymbol{a_j}$ while simultaneously collecting feedback to gain insights into the channel behavior, the agent faces the dilemma of whether to exploit the existing knowledge by choosing $\boldsymbol{a_j}$ or to explore new, untested schemes to enhance its understanding of the channel dynamics and possibly achieve higher rewards. To balance this trade-off, we

TABLE IV: LDPC RATE SELECTION CRITERION.

BER Estimation $\hat{\epsilon}_j(a_j)$	**LDPC Rate $\rho(\hat{\epsilon}_j(a_j))$**
$\hat{\epsilon}(\boldsymbol{a_j}) = 0$	1
$\hat{\epsilon}(\boldsymbol{a_j}) < 0.03$	$\frac{2}{3}$
$\hat{\epsilon}(\boldsymbol{a_j}) < 0.07$	$\frac{1}{2}$
$\hat{\epsilon}(\boldsymbol{a_j}) < 0.12$	$\frac{1}{3}$
$\hat{\epsilon}(\boldsymbol{a_j}) < 0.15$	$\frac{1}{4}$
$\hat{\epsilon}(\boldsymbol{a_j}) < 0.18$	$\frac{1}{6}$

employ a dynamic ε-greedy algorithm to select the modulation scheme $\boldsymbol{a_j}$. The classic ε-greedy policy is expressed as

$$\boldsymbol{a_j} = \begin{cases} \underset{\boldsymbol{a_j}\in\mathcal{A},\rho(\epsilon(\boldsymbol{a_j}))\in\boldsymbol{\varrho}}{\operatorname{argmax}} d(\boldsymbol{a_j})\rho(\hat{\epsilon}_j(\boldsymbol{a_j})), & \text{with probability } 1-\varepsilon \\ \text{Random}, & \text{with probability } \varepsilon \end{cases}, \tag{18}$$

where ε is for exploring $\boldsymbol{a_j}$ randomly to avoid being trapped in a local optimum. In our dynamic ε-greedy algorithm strategy, ε gradually decayed by

$$\varepsilon = \begin{cases} \varepsilon_{\text{decay}} \times \varepsilon, & \text{if } \varepsilon > \varepsilon_{\text{min}} \\ \varepsilon_{\text{min}}, & \text{if } \varepsilon \leq \varepsilon_{\text{min}} \end{cases}, \tag{19}$$

where $\varepsilon_{\text{decay}} = 0.9$ is the decay coefficient to control the degree of randomness and $\varepsilon_{\text{min}} = 0.1$ is the minimum value of the random factor. During the initial phase of learning, a larger ε value is applied to encourage the agent to explore untried modulation schemes with a higher probability. As the number of transmission frames increases, the agent tends to rely and exploit more on the accumulated knowledge base, which improves the learning efficiency.

IV. DYNAMIC FEEDBACK STRATEGY

Gathering CSI via feedback from the receiver is essential for AMC. However, in UAC systems with long propagation delays, tuning modulation schemes $\boldsymbol{a_j}$ and waiting for feedback on each frame consume time. Furthermore, incomplete channel knowledge in the early stage of communication or channel variability during transmission may lead to sub-optimal AMC strategies. Obtaining the feedback after a significant number of frames under a sub-optimal AMC strategy possibly results in decreased data throughput.

Therefore, we propose a dynamic feedback scheduling strategy to optimize the channel throughput by determining the optimal time to tune the modulation scheme and obtain feedback based on the channel conditions.

The sequential decision process involving the modulation scheme $\boldsymbol{a_j}$ and FRI h_j for transmitting N bits is formulated as an MDP. Tree search algorithms are suitable for solving MDPs as they optimize long-term rewards and balance exploration-exploitation trade-offs. However, traditional tree search algorithms face computational challenges when building search trees in high-dimensional action or state spaces. Uncertainty in untried state-action pairs is initially unknown. In large action and state spaces, handling uncertainty in tree search also introduces significant computational complexity, potentially resulting in suboptimal decisions.

RL is another popular method that enables the agent to learn an optimal policy in MDPs through interactions with the environment, without explicit knowledge of the environment's dynamics [51]. Deep Q-Network (DQN), a powerful RL algorithm, aims to learn a mapping function that predicts the expected reward for state-action pairs, known as the Q-value function [9], [13]. The use of DNN as function approximators in DQN enhances its ability to generalize to unseen states. However, in DQN, quick but possibly biased action selections without planning the potential consequences and future states may result in short-sighted decisions and suboptimal long-term outcomes [52].

Therefore, we propose Tree Search with DQN (TS-DQN) to benefit from the planning capabilities of tree search and the generalization capabilities of DQN. DQN focuses on learning the optimal Q-value function for state-action pairs using observed experiences, even leveraging its generalization abilities to approximate the value function in a continuous state space. Tree search utilizes the updated Q-values to guide the exploration process, prioritize promising state-action pairs, and provide a way to estimate long-term rewards. Fig. 7 illustrates the fundamental framework of the proposed TS-DQN algorithm and Fig. 8 further depicts the details of the tree search procedure. We will explain them in the rest of this section.

A. Q-Value Function

Agent aims to select $\boldsymbol{a_j}$ and h_j at state $\boldsymbol{s_j}$ to maximize the expected throughput until reaching the terminal state $\boldsymbol{s_J}$. In Fig.7, the output of DQN provides a prediction of the Q-value function, i.e., $\hat{Q}(\boldsymbol{s_j}, \boldsymbol{a_j}, h_j)$. $\hat{Q}(\boldsymbol{s_j}, \boldsymbol{a_j}, h_j)$ approximates the throughput till terminal state $\boldsymbol{s_J}$ given any state-action pair

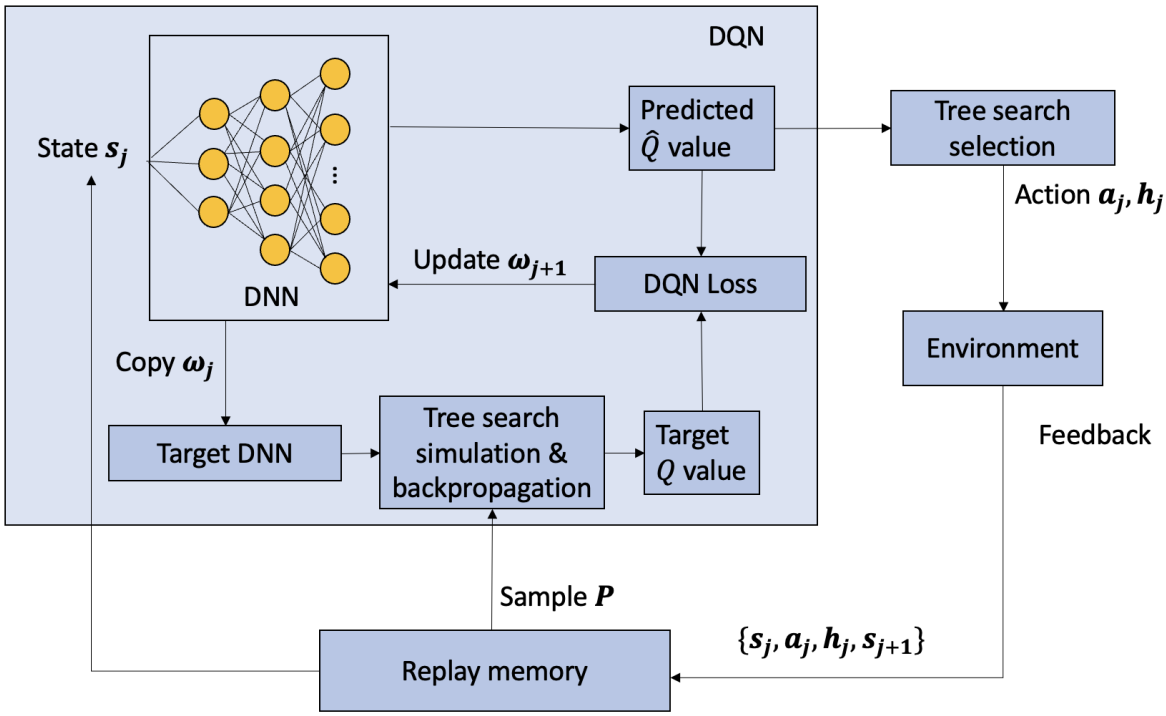

Fig. 7: The framework of the TS-DQN algorithm.

$\{a_j, h_j\}$. The agent prioritizes actions that are likely to yield favorable long-term rewards. Specifically, at state s_j, the average throughput, R_j, from the current state s_j to the terminal state s_J is calculated as:

$$R_j = \frac{(1-n'_j)N}{\sum_{i=j}^{J} h_i\tau_i + (J-j)(\tau_{\text{fd}} + 2\tau_{\text{pd}} + \tau_{\text{m}})}. \tag{20}$$

The feedback of FRI h_j from the receiver node updates $(n'_{j+1} - n'_j)N$ bits transmitted during FRI h_j within the time period encompassing h_j frames with a transmission duration of τ_j each, the duration of frames containing modulation information τ_{m} and feedback τ_{fd}, and a two-way propagation delay of $2\tau_{\text{pd}}$. However, from state s_{j+1} to the terminal state s_J, the number of transmitted bits and their corresponding duration are unknown, as they have not yet been attempted. In TS-DQN, the agent navigates a search tree and uses the target DNN to simulate the transmission process from state s_i to the terminal state, thereby enabling the calculation of R_i, $i = [j+1, J]$. Consequently, our Q-value aims to approximate R_j based on the actions a_j and h_j selected at state s_j. In the upcoming subsection IV-D, we will present the tree search approach for approximating the target Q-value $Q(s_j, a_j, h_j)$.

B. State-value Approximation

Traditional tree search methods lack explicit policies and require repeated tree building at each state. This procedure can be time-consuming and memory-intensive. Our proposed TS-DQN algorithm addresses these limitations by leveraging the approximated Q-value function to learn the rewards associated with potential state-action pairs during repeated look-ahead tree construction.

In Fig. 7, the DNN consists of one input layer, three hidden layers, and one output layer, which is used to decide $h_j \in \mathcal{H}$. This DNN is utilized to model the analytical function between the state $\{n'_j, k_j, r_{j-1}\} \in \boldsymbol{s_j}$, and the estimated reward given the selected $\boldsymbol{a_j}$ and h_j which represents the predicted Q-value $\hat{Q}(\boldsymbol{s_j}, \boldsymbol{a_j}, h_j)$, i.e.,

$$\hat{Q}(\boldsymbol{s_j}, \boldsymbol{a_j}, h_j) = \mathcal{M}\left(\boldsymbol{a_j}, h_j; \left\{n'_j, k_j, r_{j-1}\right\}, \boldsymbol{\omega_j}\right) \tag{21}$$

where $\boldsymbol{\omega_j}$ is the weights of model $\mathcal{M}(\cdot)$ and updated once CSI received.

C. Replay Memory

In our TS-DQN framework, as shown in Fig. 7, the agent utilizes a Replay Memory buffer to store its experiences $p_i = \{\boldsymbol{s_i}, \boldsymbol{a_i}, h_i, \boldsymbol{s_{i+1}}\}$ for training. This allows the model to reuse past experiences for gradient updates, promoting stability and efficient learning. The target DNN serves as a fixed reference model during each training iteration, with its parameters $\boldsymbol{\omega_{target}}$ copied from the main DNN model periodically to stabilize training. Specifically:

- The target DNN remains frozen throughout a training batch to mitigate instability arising from rapidly changing target values.
- A batch of 32 transitions, denoted as $\mathbf{P}$, is sampled from the Replay Memory to compute target Q-values. For each transition $p_i \in \mathbf{P}$, $i = [1, 32]$, the target throughput from the current state $\boldsymbol{s_i}$ to the terminal state $\boldsymbol{s_J}$ is approximated based on the tree search trajectories depicted in Fig. 8.

The DQN weights $\boldsymbol{\omega_j}$ at state $\boldsymbol{s_j}$ are updated using the ADAM optimizer, which minimizes the mean squared error between the predicted $\hat{Q}$-value and the computed target Q-value. The optimization objective for updating the weight parameters $\boldsymbol{\omega_{j-1}}$ to $\boldsymbol{\omega_j}$ in the state-action-value function approximator $\mathcal{M}(\cdot)$ is formulated as:

$$\boldsymbol{\omega_j} = \underset{\boldsymbol{\omega_j}}{\operatorname{argmin}} \left(\sum_{p_i \in \mathbf{P}} \left(Q(\boldsymbol{s_i}, \boldsymbol{a_i}, h_i) - \hat{Q}(\boldsymbol{s_i}, \boldsymbol{a_i}, h_i)\right)^2 \right). \tag{22}$$

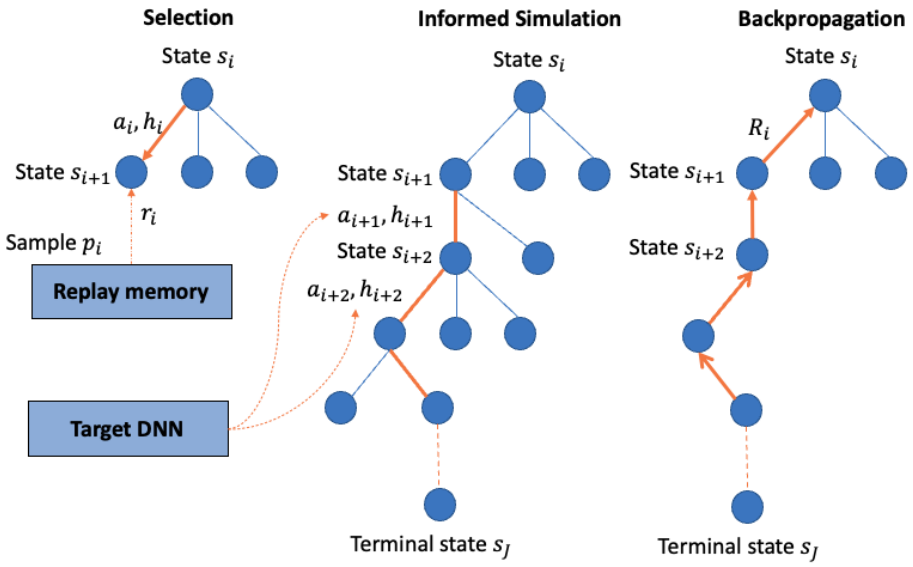

Fig. 8: The framework of the tree search algorithm.

The replay memory ensures diverse and decorrelated training samples, thereby improving the convergence of the TS-DQN framework.

D. Tree Search

Fig. 8 illustrates the structure of our lookahead tree search framework, which comprises three main stages: Selection, Simulation, and Backpropagation. When the agent reaches the state $\boldsymbol{s_j}$, the three stages proceed as follows:

- Selection: Agent selects a modulation scheme $\boldsymbol{a_j}$ using (18) first and determines FRI h_j to maximize the predicted $\hat{Q}$-value, i.e.,

$$h_j = \underset{h_j \in \mathcal{H}}{\operatorname{argmax}}\, \hat{Q}(\boldsymbol{s_j}, \boldsymbol{a_j}, h_j), \tag{23}$$

  with a probability of 0.9. With the remaining probability of 0.1, the agent performs random exploration in the decision tree. After $\boldsymbol{a_j}$ and FRI h_j are determined, frames are encoded for transmission, and feedback is used to update the AMC strategy in (18) and the next state $\boldsymbol{s_{j+1}}$. The tuple $p_j = \{\boldsymbol{s_j}, \boldsymbol{a_j}, h_j, \boldsymbol{s_{j+1}}\}$ is then stored in the Replay Memory.
- Simulation: For each $p_i = \{\boldsymbol{s_i}, \boldsymbol{a_i}, h_i, \boldsymbol{s_{i+1}}\}$, $i \in [1, j]$, sampled from the Replay Memory, the agent performs simulations starting from the newly added node $\boldsymbol{s_{i+1}}$ to the end of the transmission. During this simulation phase, the target DNN estimates the value of possible state-action pairs and selects actions following the same exploration strategy during Selection stage until a terminal state is reached.

- Backpropagation: The outcomes of the simulation, including FRI values, throughput, and timestamps of each FRI between visited states, are backpropagated up the tree to update the target Q-value.

The cumulative reward obtained through this informed simulated rollout is used as the reference Q-value, i.e., target Q-value, serving as a training target for the DQN. By minimizing the difference between the predicted and target Q-value for each $p_i \in \mathbf{P}$ as defined in (22), the DQN learns to accurately estimate long-term expected throughput over the episode.

V. Experiment Design and Results

In order to evaluate the performance of our algorithms, we tested our algorithms in a test tank and at sea in Singapore. This section provides a detailed description of the experiment setup, testing procedure, and results.

A. Experimental Setup

In our experiments, the TX and RX nodes are WNC-M25MSS3 modems [53] from Subnero. These modems support two types of links: CONTROL and DATA. The CONTROL link employs Frequency-Hopping Binary Frequency Shift Keying (FHBFSK) as the modulation technique and LDPC code as the FEC method. In different environments, the LDPC code rate and power level of the CONTROL link are pre-tuned and remain static. On the DATA link, the LDPC code is used as FEC, and OFDM serves as the modulation technique. For each FRI, the number of subcarriers, cyclic prefix length, and bandwidth in OFDM are tuned adaptively. The "test" mode on the DATA link in the WNC-M25MSS3 modems can be enabled or disabled.

Frames transmitted over the CONTROL link serve two purposes. The first purpose involves sending frames containing modems' modulation and setup information at the beginning of each FRI. Specifically, this information encompasses the number of subcarriers, cyclic prefix length, and bandwidth in the OFDM system, the LDPC code rate, "test" mode information, and FRI value. Secondly, the CSI feedback from the RX modem to the TX modem is sent over the CONTROL link. The feedback frame contains measured BER statistics along with the number of bits transmitted during one FRI.

To ensure sufficient reception of frames over the DATA link without compromising throughput unnecessarily, we should consider an appropriate wait duration on the RX modem after the modulation setup. This wait duration, denoted as τ_{wait}, is calculated as $h_j \tau_j$, where τ_j represents the transmission duration of each frame in the j^{th} FRI. Upon receiving the frame with the modulation and setup information of

modems over the CONTROL link, the RX modem triggers the waiting phase for the upcoming h_j frames over the DATA link, lasting for $\tau_{\text{wait}} = h_j \tau_j$.

The operational procedures on the TX modem and RX modem are detailed in Algorithm 1 and Algorithm 2, respectively.

Algorithm 1 Operations on TX Modem over Transmission

INITIALIZATION: state $\boldsymbol{s_1} = \{\boldsymbol{\theta_1}, \boldsymbol{\eta_1}, \boldsymbol{\omega_1}, n'_1 = 0, k_1 = 0.5\}$ where $\boldsymbol{\theta_1}$ is randomized, $\boldsymbol{\omega_1}$ is pre-trained in Section V-B and $\eta_1(\boldsymbol{a_1}) = 0.18$ for $\boldsymbol{a_1} \in \mathcal{A}$.
for $j \in [1, J]$ **do**
  Estimate BER $\hat{\epsilon}_j(\boldsymbol{a_j})$ for $\boldsymbol{a_j} \in \mathcal{A}$ using (1).
  Determine FEC rate $\rho(\hat{\epsilon}_j(\boldsymbol{a_j}))$ for $\boldsymbol{a_j} \in \mathcal{A}$ using Table IV.
  Select $\boldsymbol{a_j}$ using (18) and h_j using (23).
  Determine whether to enable or disable "test" mode.
  Transmit frame carrying modulation and "test" information over CONTROL link.
  Transmit h_j frames over DATA link.
  Detect and decode feedback frames from the RX modem over the CONTROL link.
  perform state transition $\boldsymbol{s_j} \rightarrow \boldsymbol{s_{j+1}}$.
  if $n'_{j+1} \geq 1$ **then**
    Stop transmission.
  end if
end for

Algorithm 2 Operations on RX Modem over Transmission

while receive frame over CONTROL link **do**
  Decode and Modulate.
  Enable or disable "test" mode according to the "test" mode information.
  Calculate τ_{wait}.
  while in τ_{wait} **do**
    if receive frame over DATA link **then**
      Decode and store BER.
    end if
  end while
  Send feedback frame to TX modem.
end while

B. Pre-training of Feedback Model

The tunable OFDM parameters offer a wide range of potential values, resulting in a considerably large action space $[\mathcal{A} \times \mathcal{H}]$. Training of the model $\mathcal{M}(\cdot)$ in sea trials setting becomes challenging and time-consuming. To address this, we opt to pre-train $\mathcal{M}(\cdot)$ in a test tank, as depicted in Fig. 3. This controlled and safe environment allows us to learn the initial features and patterns of model $\mathcal{M}(\cdot)$, denoted by $\hat{\boldsymbol{\omega}}$. For the CONTROL link, we utilized LDPC with a $\frac{1}{3}$ rate FEC, and we set the power level for both DATA

and CONTROL links to $155\,\text{dB re}\,1\,\mu\text{Pa}$ on TX and RX modems. Each pre-training iteration followed the operational procedures outlined in Algorithm 1 and Algorithm 2 and was terminated once N bits had been transmitted. The $\boldsymbol{\omega_J}$ obtained at the terminal state $\boldsymbol{s_J}$ of each iteration was retained and served as the initial $\boldsymbol{\omega_1}$ for the subsequent iteration. The propagation delay is negligible in the test tank. During the pre-training of $\hat{\boldsymbol{\omega}}$, we assumed different distances l between the TX modem and RX modem, specifically 1, 2, or 3 km. Consequently, we incorporated $\tau_{\text{pd}} = \frac{2}{3}$, $\frac{4}{3}$, 2 s into each FRI to account for the impact of various propagation delays. After 100 pre-training iterations, the refined $\hat{\boldsymbol{\omega}}$ was employed as the initial value of $\boldsymbol{\omega_1}$ in subsequent experiments, including those conducted in the test tank, sea trials, and simulations.

C. Tank Experiment

We first tested our algorithm in the same test tank shown in Fig. 3. We used the same two WNC-M25MSS3 modems from Subnero, positioning them in four different locations as illustrated in Fig.9. In this setup, the CONTROL link employed LDPC with a $\frac{1}{3}$ rate for FEC. The power levels for both CONTROL and DATA links were set to $155\,\text{dB re}\,1\,\mu\text{Pa}$ on TX and RX modems. The selection of modulation schemes consistently followed (18). Meanwhile, we compared our feedback scheduling algorithm, TS-DQN, with alternative strategies: a Random strategy, which selected h randomly between 1 and 20; a Fixed strategy, where h was predetermined at 5, 10, 15, or 20; and a Time-varying policy, in which FRI grew with n'_j, the ratio of transmitted bits. Additionally, we evaluated TS-DQN against the classical DQN approach implemented as described in [20] (referred to here as "DQN"). The traditional DQN employs the Bellman equation for Q-value updates, defined as follows:

$$\hat{Q}(\boldsymbol{s_j}, \boldsymbol{a_j}, h_j) = (1-\alpha)\hat{Q}(\boldsymbol{s_j}, \boldsymbol{a_j}, h_j) + \alpha\big(r_j + \delta V(\boldsymbol{s_{j+1}})\big), \tag{24}$$

$$V(\boldsymbol{s_j}) = \max_{\boldsymbol{a_j} \in \mathcal{A}} \hat{Q}(\boldsymbol{s_j}, \boldsymbol{a_j}, h_j), \tag{25}$$

where the learning rate $\alpha = 0.1$ and the discount factor $\delta = 0.9$. In this implementation:

- The state is the same as $\boldsymbol{s_j}$.
- The action is the FRI h_j, determined according to (23).
- The reward is the immediate throughput achieved during each FRI, as defined in (2).

This setup mirrors the framework outlined in [20], allowing a direct comparison between DQN's iterative Bellman-based updates and TS-DQN's tree search strategy for reward evaluation. Each transmission run employing different feedback strategies was terminated until $N = 100,000$ bits have been transmitted.

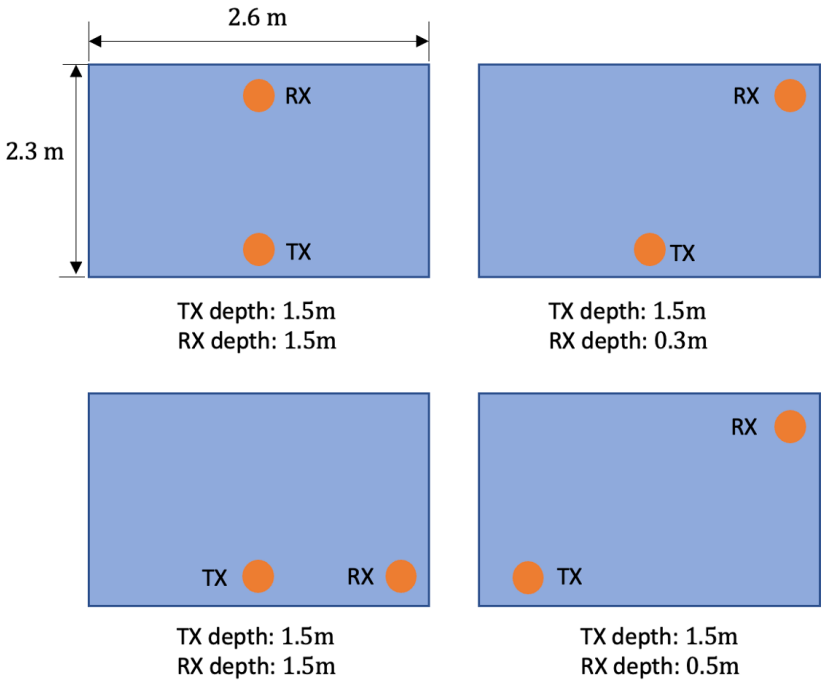

Fig. 9: 4 different deployments in the test tank.

When calculating with negligible propagation delays τ_{pd} in the test tank, the throughput results are demonstrated in Fig. 10. To observe the impact of propagation delays, we introduced a transmission distance l as 3 km, resulting in a propagation delay of $\tau_{\text{pd}} = 2$ s. The throughput results with propagation delays added are presented in Fig.11. Notably, when propagation delays are negligible, selecting a smaller FRI value enables the AMC model to converge more quickly by acquiring CSI feedback with fewer transmissions, effectively functioning as an optimal strategy in such scenarios. For the Fixed FRI policies, we can see the throughput decreases gradually as the fixed FRI value increases. The experimental results demonstrate that the proposed TS-DQN algorithm effectively selects smaller FRI values during the initial training stage of the AMC model, enabling quicker feedback updates. Along with the transmission, both TS-DQN and Time-varying strategies increment the FRI value to save time on propagation while still ensuring close-to-optimal AMC performance. However, the slower FRI adjustment speed of the Time-varying strategy, compared to our TS-DQN, results in lower throughput. When propagation delays are introduced, the advantages of TS-DQN become prominent because of its intelligence in saving time on propagation delays and feedback duration when the AMC strategy performs well. Fig.11 also shows the throughput results of the Fixed policy with a small value, like 5, and the Time-varying policy are significantly reduced. This is because they waste time requesting unnecessary feedback when the

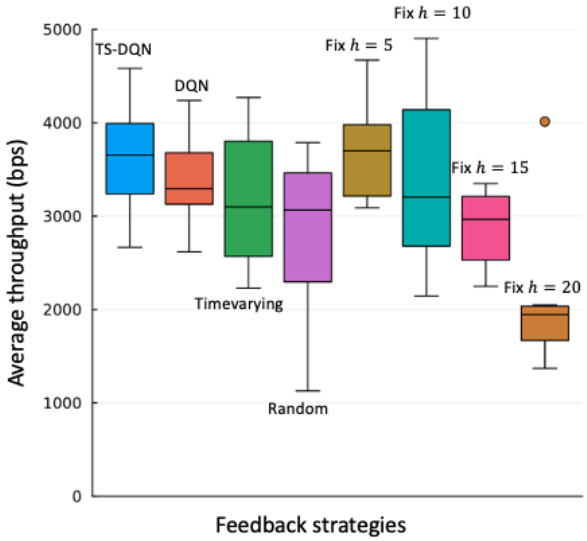

Fig. 10: Tank throughput comparison before propagation delays added.

AMC strategy is already performing well. Conversely, a large fixed FRI value, like 20, leads to slow convergence when the agent has limited channel knowledge at the beginning of transmission, thereby hindering throughput. The baseline DQN model from [20] demonstrates lower throughput compared to TS-DQN. The classical DQN algorithm relies on the Bellman equation to update Q-values, which incorporates immediate rewards and the maximum estimated future reward. Experiment results demonstrate that the classical DQN tends to select conservative, smaller FRI values at each state. While smaller FRIs provide quicker feedback and facilitate faster system exploration, the time lost due to extensive propagation delays reduces overall throughput. In contrast, TS-DQN evaluates the reward more comprehensively by expanding a search tree to terminal states. This approach accounts for the cumulative effects of decisions over entire trajectories, resulting in higher throughput and improved system performance.

D. Sea Trials

Sea trials were conducted in Singapore waters as shown in Fig.12, with an experimental setup illustrated in Fig.13. Two WNC-M25MSS3 modems were used as TX and RX nodes. The TX modem was controlled from a nearby ground station via a laptop. A separate laptop connected to the RX modem was responsible for receiving modulation setup information, guiding the RX modem in modulating and receiving frames, and instructing the RX modem to transmit feedback frames. On Day 1, both TX and RX modems were deployed in a marina, with the RX modem positioned 100 m from the TX modem. The depths of TX and RX modems were $d_1 = 5$ m and $d_2 = 3$ m, respectively. On Day 2, the RX modem was deployed from a

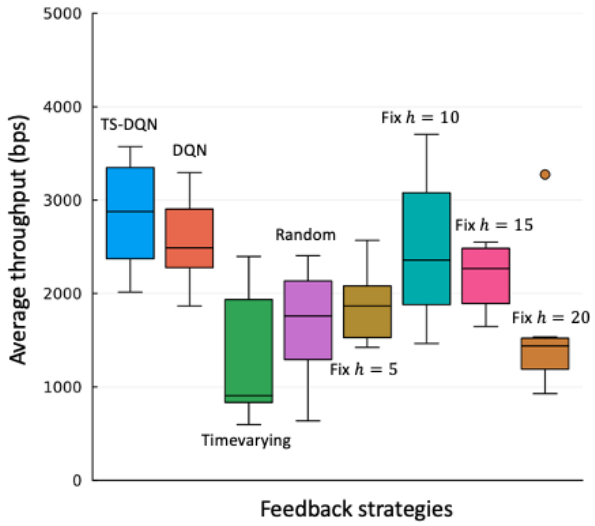

Fig. 11: Tank throughput comparison after propagation delays added.

Fig. 12: Sea trial environment: day 1 (left) and day 2 (right).

boat at a distance of 223 m from the TX modem. The depths of TX and RX modems were $d_1 = 3$ m and $d_2 = 6$ m, respectively. In our experimental setup, both TX and RX modems employed LDPC with a $\frac{1}{6}$ rate for FEC on the CONTROL link. The power levels were configured at $175\,\text{dB re}\,1\,\mu\text{Pa}$ for the TX modem and $185\,\text{dB re}\,1\,\mu\text{Pa}$ for the RX modem. Each run was terminated after transmitting $N = 100,000$ bits, following the procedures outlined in Algorithm 1 and 2 for the TX and RX modems, respectively. On Day 1, the recorded throughput for each feedback strategy, the TS-DQN, Random, Fixed, and Time-varying, is averaged over 20 transmissions. On Day 2, we did the comparison between our TS-DQN and the baseline

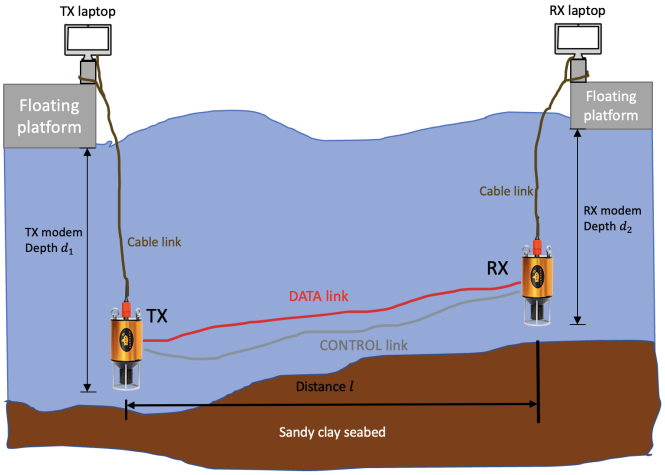

Fig. 13: Sea trial deployment.

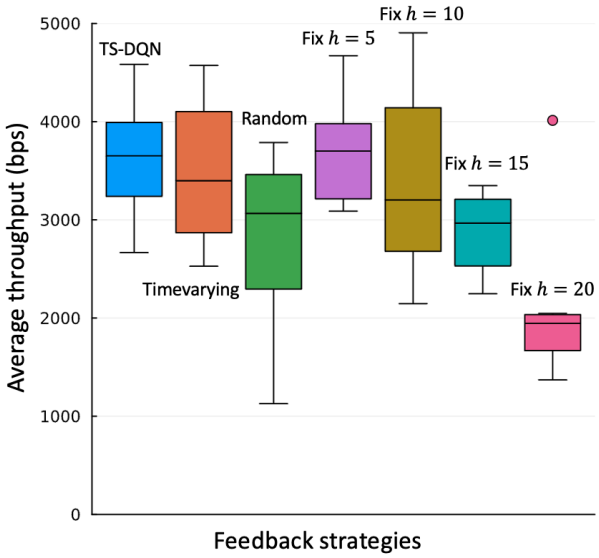

Fig. 14: Sea trial throughput comparison before propagation delays added on Day 1.

DQN model from [20]. The recorded throughput for these two feedback strategies is averaged over 10 transmissions. Fig.14 and Fig.16 depicts results incorporating the actual delay propagation ($\tau_{\text{pd}} \approx \frac{l}{1500}$ s where $l = 100$ and 223 m on Day 1 and Day 2 seperately), while Fig.15 and Fig.17 presents throughput assuming a distance $l = 3$ km.

The timing diagram of a series of transmissions from the sea trial is depicted in Fig. 18 where frames labeled “test frames” indicate the “test” mode is on and frames labeled “data frames” indicate the “test” mode is switched off. The sea trial throughput results demonstrate a decrease when compared with the tank throughput results presented in Fig.10 and Fig.11. This decrease is attributed to the increased noise from nearby shipping and construction in the marina. Under the Fixed policy, the throughput shows a

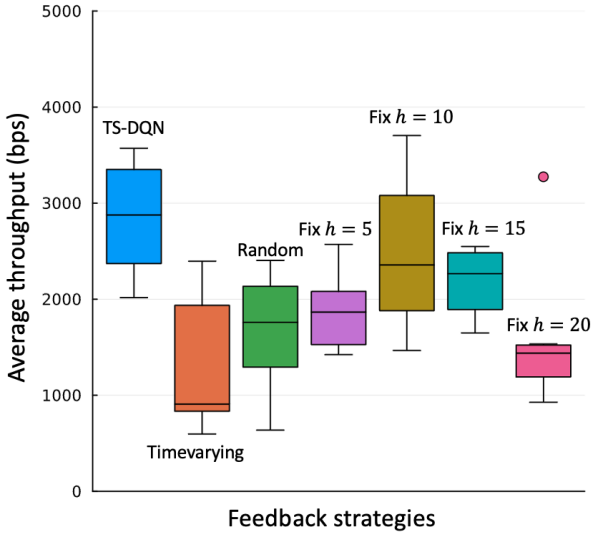

Fig. 15: Sea trial throughput comparison after propagation delays added on Day 1.

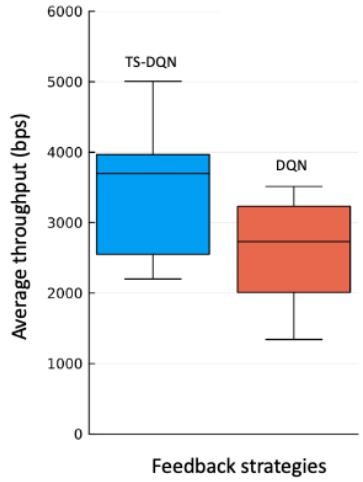

Fig. 16: Sea trial throughput comparison before propagation delays added on Day 2.

decreasing trend as the fixed FRI value increases, and the advantage of our proposed TS-DQN is not notably evident. This is due to the satisfactory convergence speed of the AMC strategy under the Time-varying and Fixed policies. When considering propagation delays for a distance of 3 km, the impact of the propagation delay increased. Our TS-DQN algorithm outperforms, as it can dynamically determine the FRI value based on the channel conditions. Specifically, TS-DQN tends to select a smaller FRI when there is limited channel knowledge or when channel conditions change, while it chooses a larger FRI when the AMC strategy is optimized or operates in a stable channel condition. With reduced ship traffic noise, the environment on Day 2 facilitated an increase in the average throughput for both the TS-DQN and DQN models. The comparison with the DQN model from [20] underscores the superior performance

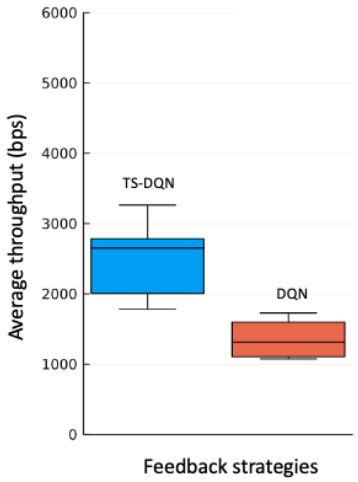

Fig. 17: Sea trial throughput comparison after propagation delays added on Day 2.

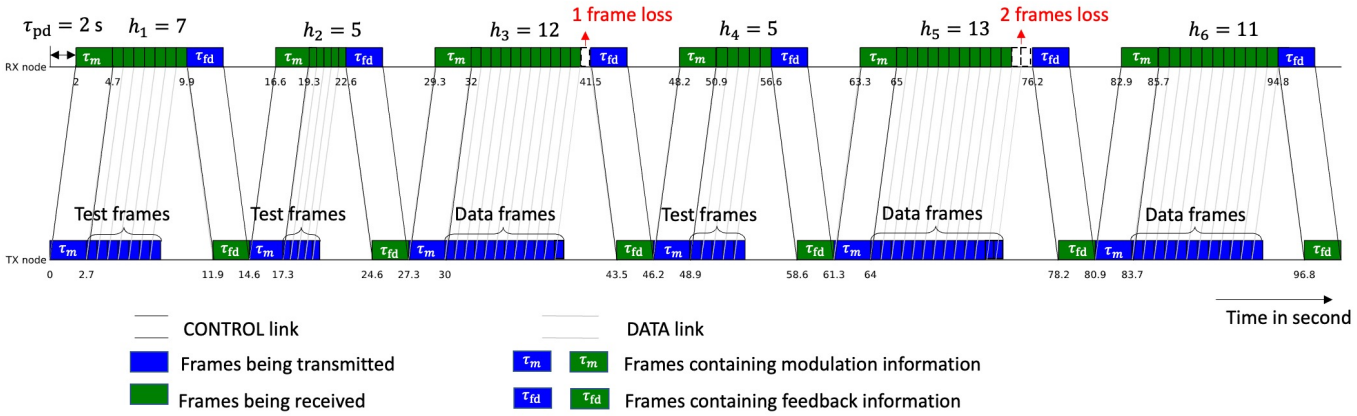

Fig. 18: An illustration of the timestamps in frame exchange between the TX and RX modems in the sea trial.

Notes: Frames labeled "test frames" indicate the "test" mode is on and frames labeled "data frames" indicate the "test" mode is switched off.

of our TS-DQN approach, particularly in adapting to dynamic, real-world environments.

VI. Simulation and Results

Sea trials validated algorithm performance in realistic UAC conditions. Because the test tank and sea trials offered limited channel conditions, we then verified our algorithm in more diverse underwater environments via simulations. We used a surrogate model to represent the DATA link of different UAC environments which was built based on [54], where the Pekeris ray model with red Gaussian noise was employed. The Pekeris ray model is a very fast fully differentiable 2D/3D ray model for isovelocity range-independent environments. There were some parameters in the Pekeris ray model we could modify, such as the bathymetry with a constant depth, and the ambient noise model with variance σ^2. We fixed the isovelocity sound speed profile with sound speed $c = 1540$ m/s, a flat sea surface, and a sandy clay

seabed. We chose the standard deviation in the red Gaussian noise to be 10^6 (Pa)^2/Hz. The transmission distance l and the depth of the TX node and RX node d_1 and d_2 for different simulation surrogate models are listed in Table. V.

In the simulation, the TX and RX nodes ran on the same machine, and the measured BER was provided directly by the surrogate model. Therefore, the duration of frames containing modulation information and feedback no longer existed. To be consistent with the practical setup in the sea trial, we assumed the $\tau_{\text{m}} = \tau_{\text{fd}} = 2.7$ s. The propagation delay τ_{pd} was determined by the distance l, i.e., $\tau_{\text{pd}} = \frac{l}{1540}$ s. The FEC selection rule was identical to Table IV. To better simulate the real environment, if the measured BER given by the environment surrogate model was less than the BER limit of each LDPC rate, the frame had a very high probability, 90%, to be successfully received. Meanwhile, a frame had a probability of 10% to be received successfully even when its measured BER was larger than the given BER limit. We compared our feedback strategy TS-DQN against Random, Fixed, Time-varying, and NN (our previous work presented in [13]) strategies. The simulation was stopped when $N = 100,000$ bits had been transmitted. The possible values of number of subcarrier n_{c} were selected from $\{64, 128, 256, 512, 1024, 2048, 4096, 8192\}$, while the value of n_{p} ranged from 0 to 8192. Additionally, the possible occupied ratio of the 24 KHz was chosen from $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$.

For different feedback strategies, transmission was executed 20 times given a different UAC surrogate model.

TABLE V: UAC SURROGATE MODEL PARAMETERS.

Surrogate Model	**Water depth**	**TX node depth d_1**	**Rx node depth d_2**	**Distance l**
1	50 m	25 m	25 m	3000 m
2	50 m	25 m	25 m	2000 m
3	10 m	5 m	5 m	100 m

For different surrogate models listed in Table V, the throughput results of different feedback strategies are presented in Fig. 19, Fig. 20, and Fig. 21. The Random strategy yields significantly lower throughput compared to the proposed TS-DQN strategy. For the Fixed strategy, the median throughput initially increases as the FRI increases but gradually decreases when FRI exceeds a certain point for $l = 2000$ m or 3000 m. However, for $l = 100$ m, the throughput continues to increase as FRI increases. This is due to the surrogate model being sensitive to the distance l between the TX node and RX node. The channel becomes more challenging, i.e., higher probability to include errors in the transmitted frames as

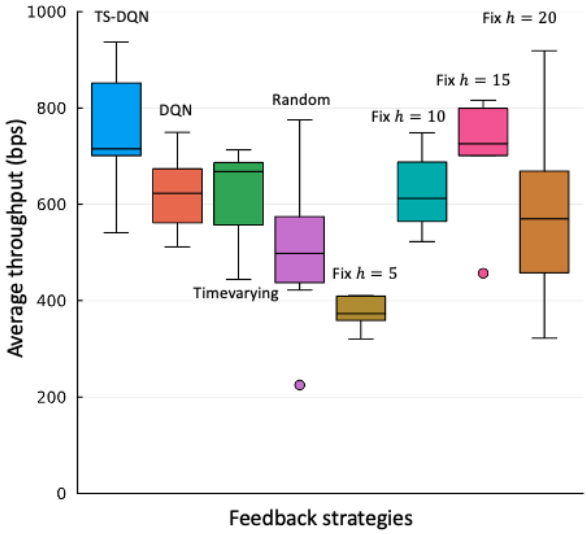

Fig. 19: Results of average throughput with different feedback strategies given surrogate model 1.

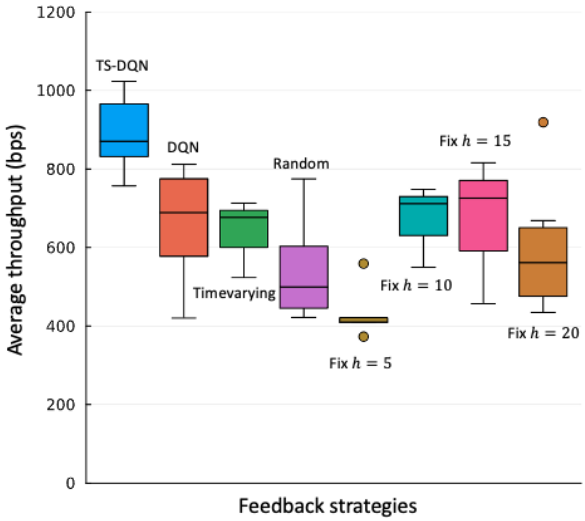

Fig. 20: Results of average throughput with different feedback strategies given surrogate model 2.

l increases. Increasing the value of FRI does not directly improve the throughput since it slows down the convergence speed of AMC when UAC channels are complex. That explains why a fixed value of FRI determined beforehand is not conducive to optimizing throughput and varies with different UAC channels. The difficulty of selecting an optimal FRI in advance emphasizes the significance of studying dynamic feedback scheduling strategies like TS-DQN to enhance AMC and optimize channel throughput. Additionally, our comparison of TS-DQN with the DQN proposed in [20] shows that TS-DQN exhibits improved robustness in complex channel conditions.

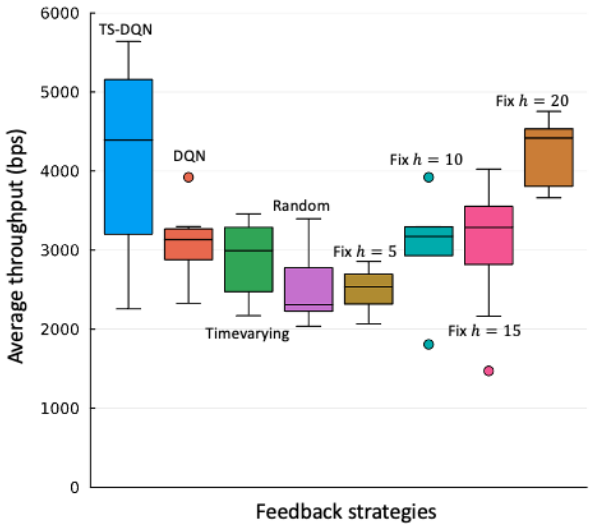

Fig. 21: Results of average throughput with different feedback strategies given surrogate model 3.

VII. Discussion and Future Work

We integrated knowledge of channel physics to design a heuristic BER estimation model for AMC, thereby reducing the dependency on extensive datasets for channel prediction. The BER estimation from this model captured the median of the measure BER gathered from Singapore waters. To account for variations and enhance robustness in practical UAC channels, we used the GPR algorithm to predict the BER upperbound, ensuring that at least 75% of the frames were successfully transmitted. Results from both tank and sea experiments, as well as simulations, affirmed the efficacy of our proposed BER distribution predictor in operating AMC. Notably, even in preliminary sea trials with limited channel knowledge, our AMC strategy can explore a variety of MCSs and procure valuable feedback CSI for AMC model training.

We also introduced an algorithm framework, TS-DQN, which incorporated tree search and DQN to dynamically determine the time to tune MCSs and obtain feedback. TS-DQN capitalized on the strategic planning capability of tree search and the generalization ability of DQN. Both experimental and simulation results underscored the advantage of TS-DQN over NN and other feedback scheduling policies in optimizing long-term channel throughput. Specifically, when compared to the best fixed-feedback policy, our TS-DQN algorithm reduced the transmission time for all N bits by up to 25%. This significant achievement underscored its potential value in the UAC market.

Our future efforts will focus on improving the MDP algorithm we introduced, while simultaneously tackling the joint exploration and exploitation during the selection of MCSs and FRI. Additionally, the existing heuristic BER estimation model encourages us to seek more universal methods for integrating

physical information into the AMC algorithm design. Potential research directions include exploring the application of our methods to modulation techniques beyond OFDM. Ultimately, we aim to improve the adaptability and robustness of our algorithm across various communication systems.

REFERENCES

[1] M. Stojanovic, "Underwater acoustic communications: Design considerations on the physical layer," in *2008 Fifth Annual Conference on Wireless on Demand Network Systems and Services*, 2008, pp. 1–10.

[2] B. Xu, X. Wang, Y. Guo, J. Zhang, and A. A. Razzaqi, "A novel adaptive filter for cooperative localization under time-varying delay and non-gaussian noise," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–15, 2021.

[3] A. Radosevic, R. Ahmed, T. M. Duman, J. G. Proakis, and M. Stojanovic, "Adaptive OFDM modulation for underwater acoustic communications: Design considerations and experimental results," *IEEE Journal of Oceanic Engineering*, vol. 39, no. 2, pp. 357–370, 2014.

[4] P. Xia, S. Zhou, and G. Giannakis, "Adaptive mimo-ofdm based on partial channel state information," *IEEE Transactions on Signal Processing*, vol. 52, no. 1, pp. 202–213, 2004.

[5] D. Love and R. Heath, "Limited feedback power loading for ofdm," in *IEEE MILCOM 2004. Military Communications Conference, 2004.*, vol. 1, 2004, pp. 71–77 Vol. 1.

[6] L. Huang, Q. Zhang, L. Zhang, J. Shi, and L. Zhangb, "Efficiency enhancement for underwater adaptive modulation and coding systems: via sparse principal component analysis," *IEEE Communications Letters*, pp. 1–1, 2020.

[7] S. Kojima, K. Maruta, and C.-J. Ahn, "Adaptive modulation and coding using neural network based SNR estimation," *IEEE Access*, vol. 7, pp. 183 545–183 553, 2019.

[8] R. C. Daniels, C. M. Caramanis, and R. W. Heath, "Adaptation in convolutionally coded MIMO-OFDM wireless systems through supervised learning and SNR ordering," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 1, pp. 114–126, 2010.

[9] L. Jing, C. Dong, C. He, W. Shi, and H. Yin, "Adaptive modulation and coding for underwater acoustic communications based on data-driven learning algorithm," *Remote Sensing*, vol. 14, no. 23, 2022. [Online]. Available: https://www.mdpi.com/2072-4292/14/23/5959

[10] Y. Zhang, J. Zhu, H. Wang, X. Shen, B. Wang, and Y. Dong, "Deep reinforcement learning-based adaptive modulation for underwater acoustic communication with outdated channel state information," *Remote Sensing*, vol. 14, no. 16, p. 3947, Aug 2022. [Online]. Available: http://dx.doi.org/10.3390/rs14163947

[11] J. Huang and R. Diamant, "Adaptive modulation for long-range underwater acoustic communication," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6844–6857, 2020.

[12] P. Anjangi and M. Chitre, "Model-based data-driven learning algorithm for tuning an underwater acoustic link," in *2018 Fourth Underwater Communications and Networking Conference (UComms)*, 2018, pp. 1–5.

[13] W. Shuangshuang, P. Anjangi, and M. Chitre, "Adaptive modulation and feedback strategy for an underwater acoustic link," in *2022 Sixth Underwater Communications and Networking Conference (UComms)*, 2022, pp. 1–5.

[14] M. R. Khan, B. Das, and B. B. Pati, "Channel estimation strategies for underwater acoustic (UWA) communication: An overview," *Journal of the Franklin Institute*, vol. 357, no. 11, pp. 7229–7265, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0016003220302325

[15] L. Liu, L. Cai, L. Ma, and G. Qiao, "Channel state information prediction for adaptive underwater acoustic downlink ofdma system: Deep neural networks based approach," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 9, pp. 9063–9076, 2021.

[16] J. P. Alan C. Farrell, "Performance of IEEE 802.11 mac in underwater wireless channels," *Procedia Computer Science*, vol. 10, no. 12, pp. 62–69, 2012. [Online]. Available: https://doi.org/10.1016/j.procs.2012.06.012

[17] A. Leon-Garcia and I. Widjaja, *Communication Networks: Fundamental Concepts and Key Architectures*, 1st ed. McGraw-Hill School Education Group, 1999.

[18] X. Guo, M. R. Frater, and M. J. Ryan, "Design of a propagation-delay-tolerant mac protocol for underwater acoustic sensor networks," *IEEE Journal of Oceanic Engineering*, vol. 34, no. 2, pp. 170–180, 2009.

[19] J. P. Leite, P. H. P. de Carvalho, and R. D. Vieira, "A flexible framework based on reinforcement learning for adaptive modulation and coding in ofdm wireless systems," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, 2012, pp. 809–814.

[20] W. Su, J. Lin, K. Chen, L. Xiao, and C. En, "Reinforcement learning-based adaptive modulation and coding for efficient underwater communications," *IEEE Access*, vol. 7, pp. 67 539–67 550, 2019.

[21] Y. R. Zheng, J. Wu, and C. Xiao, "Turbo equalization for single-carrier underwater acoustic communications," *IEEE Communications Magazine*, vol. 53, no. 11, pp. 79–87, 2015.

[22] B. Mounika and B. K. Priya, "Analysis and comparison of different channel coding techniques for underwater channel using awgn and acoustic channel," in *2018 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, 2018, pp. 1664–1669.

[23] L. M. S. L. G. Qiao, Z. Babar and J. Wu, "Mimo-ofdm underwater acoustic communication systems—a review," *Physical Communication*, vol. 23, pp. 56–64, 2017.

[24] E. Fitzgerald, M. Pióro, and A. Tomaszewski, "Energy versus throughput optimisation for machine-to-machine communication," *Sensors*, vol. 20, no. 15, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/15/4122

[25] H. Ju and R. Zhang, "Throughput maximization in wireless powered communication networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 1, pp. 418–428, 2014.

[26] Y. Zhang and Q. Ma, "Adaptive modulation and coding with cooperative transmission in mimo fading channels," in *Proceedings of the 2015 4th National Conference on Electrical, Electronics and Computer Engineering*. Atlantis Press, 2015/12, pp. 1363–1367. [Online]. Available: https://doi.org/10.2991/nceece-15.2016.240

[27] M.-F. Tsai, N. Chilamkurti, C.-K. Shieh, and A. Vinel, "Mac-level forward error correction mechanism for minimum error recovery overhead and retransmission," *Mathematical and Computer Modelling*, vol. 53, no. 11, pp. 2067–2077, 2011, recent Advances in Simulation and Mathematical Modeling of Wireless Networks. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S089571771000258X

[28] E. Lucas and Z. Wang, "Performance prediction of underwater acoustic communications based on channel impulse responses," *Applied Sciences*, vol. 12, no. 3, 2022. [Online]. Available: https://www.mdpi.com/2076-3417/12/3/1086

[29] J. Xu, K. Li, and G. Min, "Reliable and energy-efficient multipath communications in underwater sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 7, pp. 1326–1335, 2012.

[30] G. Barreto, D. H. Simão, M. E. Pellenz, R. D. Souza, E. Jamhour, M. C. Penna, G. Brante, and B. S. Chang, "Energy-efficient channel coding strategy for underwater acoustic networks," *Sensors (Basel, Switzerland)*, vol. 17, no. 4, p. 728, 2017.

[31] G. Zhu, B. Feng, and W. Liu, "A ber model for turbo codes on awgn channel," in *Proceedings of 2005 IEEE International Workshop on VLSI Design and Video Technology, 2005.*, 2005, pp. 419–422.

[32] J. Laster, J. Reed, and W. Tranter, "Bit error rate estimation using probability density function estimators," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 1, pp. 260–267, 2003.

[33] A.-A. Enescu, B.-M. Sandoi, and C.-G. Dinu, "A low-complexity bit error rate estimation algorithm for wireless digital receivers," in *2014 10th International Conference on Communications (COMM)*, 2014, pp. 1–4.

[34] S. Awino, T. J. O. Afullo, M. Mosalaosi, and P. O. Akuon, "Gmm estimation and ber of bursty impulsive noise in low-voltage plc networks," in *2019 PhotonIcs Electromagnetics Research Symposium - Spring (PIERS-Spring)*, 2019, pp. 1828–1834.

[35] R. Holzlöhner and C. R. Menyuk, "Use of multicanonical monte carlo simulations to obtain accurate bit error rates in optical communications systems," *Optics letters*, vol. 28, no. 20, pp. 1894–1896, 2003.

[36] B. Mazzeo and M. Rice, "On monte carlo simulation of the bit error rate," in *2011 IEEE International Conference on Communications (ICC)*, 2011, pp. 1–5.

[37] S. Saoudi, T. Derham, T. Ait-Idir, and P. Coupe, "A fast soft bit error rate estimation method," *EURASIP journal on wireless communications and networking*, vol. 2010, no. 1, 2010.

[38] R. Simeon, T. Kim, and E. Perrins, "Machine learning with gaussian process regression for time-varying channel estimation," in *ICC 2022 - IEEE International Conference on Communications*, 2022, pp. 3400–3405.

[39] Gowrishankar, R. Babu H.S., and P. Satyanarayana, "Neural network based ber prediction for 802.16e channel," in *2007 15th International Conference on Software, Telecommunications and Computer Networks*, 2007, pp. 1–5.

[40] A. Charrada, "SVM based on LMMSE for high-speed coded OFDM channel with normal and extended cyclic prefix," *Physical Communication*, vol. 29, pp. 288–295, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1874490717306146

[41] T. Elomaa and T. Malinen, "On lookahead heuristics in decision tree learning," in *Foundations of Intelligent Systems: 14th International Symposium, ISMIS 2003, Maebashi City, Japan, October 28-31, 2003. Proceedings 14*. Springer, 2003, pp. 445–453.

[42] M. Świechowski, K. Godlewski, B. Sawicki, and J. Mańdziuk, "Monte carlo tree search: a review of recent modifications and applications," vol. 56, no. 3, pp. 2497–2562. [Online]. Available: https://doi.org/10.1007/s10462-022-10228-y

[43] K. Takada, H. Iizuka, and M. Yamamoto, "Reinforcement learning to create value and policy functions using minimax tree search in hex," *IEEE Transactions on Games*, vol. 12, no. 1, pp. 63–73, 2020.

[44] A. Agarwal, K. Muelling, and K. Fragkiadaki, "Model learning for look-ahead exploration in continuous control," vol. 33, no. 1, pp. 3151–3158. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/4181

[45] G. Chen, E. Rodriguez-Villegas, and A. J. Casson, "Chapter 5.1 - wearable algorithms: An overview of a truly multi-disciplinary problem," in *Wearable Sensors*, E. Sazonov and M. R. Neuman, Eds. Oxford: Academic Press, 2014, pp. 353–382. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780124186620000052

[46] A. Akinshin, "Quantile absolute deviation," 2022. [Online]. Available: https://arxiv.org/abs/2208.13459

[47] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Cambridge, Mass: MIT Press, 2006.

[48] M. G. Genton, "Classes of kernels for machine learning: A statistics perspective," *J. Mach. Learn. Res.*, vol. 2, p. 299–312, mar 2002.

[49] M. Chitre, T.-B. Koay, G. Deane, and G. Chua, "Variability in shallow water communication performance near a busy shipping lane," in *2021 Fifth Underwater Communications and Networking Conference (UComms)*, 2021, pp. 1–5.

[50] A. Shokrollahi, "Ldpc codes: An introduction," in *Coding, cryptography and combinatorics*. Springer, 2004, pp. 85–110.

[51] W. B. Powell, "From reinforcement learning to optimal control: A unified framework for sequential decisions," Dec 18 2019. [Online]. Available: http://libproxy1.nus.edu.sg/login?url=https://www.proquest.com/working-papers/reinforcement-learning-optimal-control-unified/docview/2323280441/se-2

[52] A. Thomas, Z. Tian, and D. Barber, "Thinking fast and slow with deep learning and tree search," Dec 03 2017. [Online]. Available: http://libproxy1.nus.edu.sg/login?url=https://www.proquest.com/working-papers/thinking-fast-slow-with-deep-learning-tree-search/docview/2076517562/se-2

[53] "UnetStack3: the underwater networks project," https://unetstack.net/, accessed: 2022.

[54] M. CHITRE, "A high-frequency warm shallow water acoustic communications channel model and measurements," *The Journal of the Acoustical Society of America*, vol. 122, no. 5, pp. 2580–2586, 2007.