Deep Joint Source-Channel Coding for Underwater Image Transmission

Khizar Anjum
Rutgers University
New Brunswick, NJ, USA
khizar.anjum@rutgers.edu

Zhuoran Qi
Rutgers University
New Brunswick, NJ, USA
zhuoran.qi@rutgers.edu

Dario Pompili
Rutgers University
New Brunswick, NJ, USA
pompili@rutgers.edu

ABSTRACT

Traditional methods for coding underwater acoustic communications are bound to be surpassed by methods optimizing for source-channel coding jointly. However, the complexity of joint-optimization has thwarted successful breakthroughs in this area. We, therefore, present a novel approach, where we model the coding problem as the translation problem of the input sequence to another 'language', depending on the estimated channel conditions. We use Long Short-Term Memory (LSTM)-based sequence-to-sequence models to enable this and explain our approach in detail.

ACM Reference Format:
Khizar Anjum, Zhuoran Qi, and Dario Pompili. 2022. Deep Joint Source-Channel Coding for Underwater Image Transmission. In *The 16th International Conference on Underwater Networks & Systems (WUWNet'22), November 14–16, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3567600.3568138

1 INTRODUCTION

The transmission of multimedia data such as images and videos is of foremost importance for researchers working in the field of underwater exploration and monitoring as such data could provide vital information about the number, health and distribution of various species in the underwater environment. However, such transmission is challenging as underwater acoustic channel is challenging and has low bandwidth. It's usually modelled as a Rician fading channel for short-range shallow water communication (with a depth of less than 100 m, where the power of the Line-of-Sight (LOS) signal is stronger than the multipath delay signals) [15, 22]. Furthermore, it is non-stationary on time scales relevant to usual communication applications. These fluctuations can be due to seasonal changes in temperature profiles, fish populations, storms, tidal changes, and internal or surface gravity waves [28]. This means that even static images require innovative communication solutions to be transmitted successfully. Furthermore, the physical properties determining sound propagation underwater are myriad and their complex interactions make the modelling of multipath propagation and Doppler effects theoretically intractable. Due to this reason, a system that uses one kind of coding and modulation scheme parameters will underperform over an extended period of time and hence an adaptive system is desired [19]. Therefore, we present our adaptive scheme in Figure 1, which adapts the coding based on both the estimated Channel State Information (CSI), received as feedback, and the current input data.

Motivation: Most of the work towards realizing such an adaptive communication protocol has been directed towards optimizing source coding and channel coding separately, or rather optimizing parameters of hand-made codes [19], such as Joint Photographic Experts Group (JPEG) coding, Turbo coding, etc. This has been mainly motivated by Shannon's separation theorem [24], which states that under unlimited delay and complexity, separate optimization is as good as global optimization. However, this assumption breaks down in multi-user scenarios, and non-ergodic source or channel distributions [29], making it a subpar policy for designing communication pipelines, especially underwater. This is the reason a Joint Source-Channel Coding (JSCC) scheme is necessary for achieving the best performance in challenging conditions such as underwater acoustic channel. Conventionally, the transmitter changes its coding and modulation scheme to achieve the lowest Bit Error Rate (BER) and the best effective data rate possible. This conventional approach, however, has two major deficiencies: i) It does not take into account the input data distribution, leading to subpar compression, and ii) It attempts to optimize an inherently separate source-channel coding design. We, therefore, propose a data- as well as the channel-aware approach (see Figure 1) to encode an image. Our approach, in addition to benefiting from the joint optimization, also benefits from being data-aware, in so as we use both nature of underwater images as well as adaptively change the error-protection (channel coding) based on the CSI feedback to achieve optimum trade-offs.

Our Contributions: Overall, we make the following contributions to the state-of-the-art research in the area of adaptive joint source-channel coding:

- We propose a Convolutional Neural Network (CNN) based encoder and decoder structure that is used to extract useful and important features out of the images.
- We propose a novel approach for joint source-channel coding by posing it as a translation problem and using sequence-to-sequence learning to solve it, which is the first time it has been tried for this application.
- We evaluate our proposed approach against all types of conventional approaches (both model-based and Neural Network (NN)-based) and present a detailed analysis under different channel conditions and data.

Outline: The rest of this article is organized as follows: In Sect. 2, we examine the relevant literature and position our work in regards to it. In Sect. 3, we introduce our proposed approach, while Sect. 4

WUWNet'22, November 14–16, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9952-4/22/11…$15.00
https://doi.org/10.1145/3567600.3568138

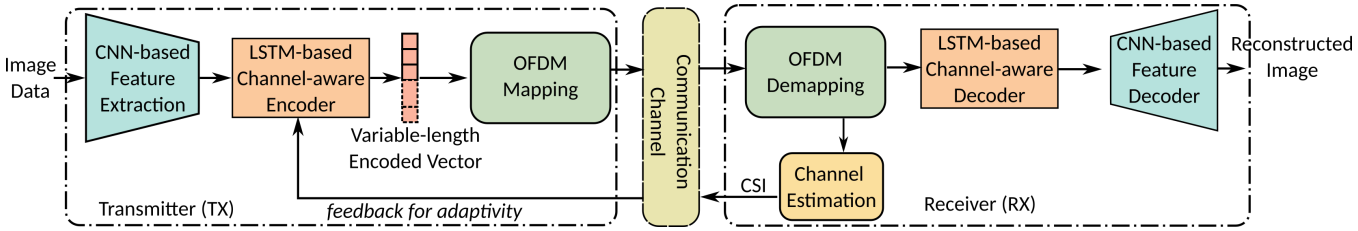

Figure 1: An outline of our proposed acoustic deep JSCC scheme as well as the variable length encoder, which is both data- and channel-dependent.

details the evaluation of our approach with comparisons of both the state-of-the-art as well as comparable techniques. Finally, Sect. 5 concludes the paper with a final summary and discussion on the future directions of this work.

2 RELATED WORK

Our approach is related to several topics, including compression, channel coding, JSCC, and channel-aware adaptivity. Accordingly, we present the comparison under three headings: i) JSCC, ii) Image Compression and ii) Adaptive Modulation and Coding (AMC).

JSCC: Existing works in the area of JSCC have been mostly limited to theoretical analysis under idealistic source and channel distributions [11, 12], or to the joint optimization of inherently separate source and channel encoders. Bourtsoulatze et al. [7] propose a deep autoencoder-based architecture to map an image to complex symbols to be transmitted through the channel, and show that their scheme outperforms separate source-channel encoding schemes like JPEG, or Low-Density Parity-Check (LDPC) codes [9], etc. This work is then also shown to do well when the available channel inputs are limited (QAM, BPSK, etc.), e.g., due to the transmitter's physical limitations [27]. Yan et al. [33] propose a similar deep autoencoder network for a multi-input network with one receiver (MISO) with multiple encoders at the transmitter side. These works, however, lack the concept of feedback to enable adaptation at the transmitter side. Towards this end, Kurka et al. [16] use channel feedback to enable better-quality image transmission. However, their adaptation consists of re-transmissions of the partial image samples, based on channel feedback, hence enhancing the received image quality at the receiver. Xu et al. [32] propose an end-to-end deep JSCC scheme that adapts the Signal-to-Noise Ratio (SNR) using the CSI obtained through feedback, and reports a performance boost as compared to other feedback-based separate source-coding schemes. However, their testing environment is limited to a terrestrial radio channel, and their approach has not been tested for a challenging underwater acoustic channel. On the other hand, our approach generates variable length transmission codes, which depend on the input image and CSI. This leads to better usage of the bandwidth and superior adaptation to the variable underwater acoustic channel.

Image Compression: A lot of work has been done towards making the compression (source coding) of images selective to the content and the structure of the image. Model-based techniques like JPEG [30], or JPEG 2000 [26] do not take into account the nature of the input image or the data distribution, leading to subpar performance. JPEG 2000 has a mode that accepts the specification of a Region of Interest (RoI) and optimizes its compression at the expense of the background. This, however, still needs a specification of the RoI by the user or some other algorithm. On the other hand, better performances have been achieved by using neural networks to learn the relatively important content in images. For example, Akutsu et al. [3] propose an additional selective detail decoder that pays more attention to the generation of smaller, finer details in order to reconstruct images with higher sharpness for elements such as text or faces, and hence a higher performance in the Structural Similarity Index (SSIM) [31]. Post-processing enhancement using generative techniques has also been used in several other techniques [6, 8, 18] and serves well to fill in the gaps in the recovered distorted image. However, even though such compression is data-aware, it is not designed with any communication challenge in mind. For example, JPEG exhibits a "cliff effect" [7] whereby under a certain Signal-to-Noise Ratio (SNR), it is almost never able to recover the image leading to a sudden deterioration in performance. Similar concerns may exist for generative methods as it is known that noise added to latent representations can change the generated output. Anjum et al. [4] devise a neural network based approach to deal with such "cliff effects" and showed that a smooth deterioration can be achieved using this approach but assume perfect knowledge of the acoustic channel. In this paper, on the other hand, we go one step further and account for both source-coding (compression) and channel-coding at the same time, and evaluate it for different SNR regimes for both Additive White Gaussian Noise (AWGN) channels and multipath fading channels.

Adaptive Modulation and Coding (AMC): Adaptation of transmitter as well as receiver parameters (modulation, transmit power, message size, equalizer taps, etc.) in a communication system depending on the condition of the communication channel to achieve the best performance is a well-known topic in underwater communications [10, 21]. In this design, the algorithm for this adaptation takes the central stage, and therefore, many kinds of solutions have been proposed, including model-based as well as NN-based solutions. Pelekanakis et al. [19] propose a decision-tree-based approach to determine modulation and coding schemes for a required BER using estimates of channel delay, Doppler spread and the received SNR. Furthermore, Shankar and Chitre [23] frame the adaptation as a multi-armed bandit problem, where the goal is to *select* between different schemes to maximize the expected reward. They present a Dynamic Programming (DP) solution to this problem by carefully balancing the exploration and exploitation in the search-space to optimize both expected BER and code rate. Petroccia et al. [20] propose a Cross-Entropy (CE) based algorithm

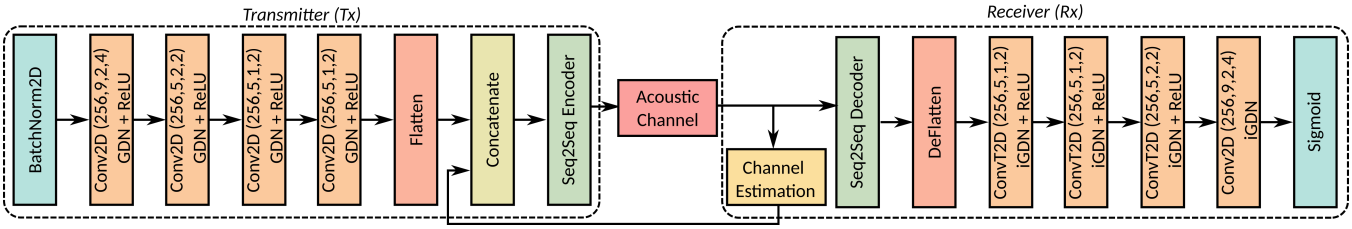

Figure 2: Detailed structure of our JSCC network with feedback and description of the parameters used. Conv2D layers are parameterized as (*output channels, kernel size, stride, padding*).

to select between schemes. Using a machine learning perspective, Huang et al. [13, 14] frame the adaptation as a classification problem. All of these works, however, operate in a *selection* space that is discrete, and try to *choose* a combination of optimal parameters given some channel indicator (SNR, Doppler spread, etc.). Furthermore, they do not take into account the nature of the input data. On the other hand, our proposed approach takes care of both of these short-comings by: i) Learning a network that generates complex coded symbols depending on CSI and SNR, and ii) Taking into account the content of a given image and extracting relevant features out of it.

3 PROPOSED APPROACH

Our proposed approach is illustrated in detail in Figure 1, and consists of three main components, which are explained exhaustively in this section. First we explain the structure of our proposed Convolutional NN-based feature extractor, and then we move on to Long Short-Term Memory (LSTM)-based joint source-channel encoder. Finally, we describe the proposed loss functions and training scheme for our approach and then present a comprehensive evaluation in Sect. 4.

3.1 CNN-based Feature Extraction

Given the nature of underwater data, the images taken underwater vary considerably in their nature. A vast number of images in the underwater scene are unclear as the water is either muddy or has a number of particles dissolved in it. Furthermore, such passive photography is only possible in shallow water as natural light rapidly scatters when entering the water through the surface. Furthermore, a vast majority of underwater images have large parts of the images containing only water, or plain background. This presents an opportunity to extract and compress the underwater images accordingly, i.e., by first extracting the features and then using them to unequally code different parts of the image.

We propose a CNN encoder $\mathcal{E}$ to extract the important parts of the image in an unsupervised manner. The architecture of our CNN-encoder is illustrated in detail in Figure 2. It consists of first a batch-normalization layer, which is then followed by a convolution layer with Generalized Divisive Normalization (GDN) [5], and Rectified Linear Unit (ReLU) non-linearities. This block is then repeated three more times with slightly different parameters, as shown in Figure 2. Finally, the flatten layer converts the features from a matrix of size (C, H, W) to size $(C, H \times W)$, where C is the number of channels in the last layer, and H and W denote the height and width of the resulting features. The final encoded representations are then passed through an LSTM-based JSCC encoder which generates variable-length latent-vector encodings given the feedback CSI of the channel. After going through the LSTM-based JSCC encoder, the signal is also quantized to INT8 representation to be then encoded into a given scheme (based on CSI) and transmitted over the acoustic channel. The parameters for the Orthogonal Frequency-Division Multiplexing (OFDM)-based transmission are also estimated using another feed-forward network which determines the mode of transmission of the image. Hence, finally, the decoder receives quantized and distorted representations to be restored. The decoder is designed as an inverse multi-scale transform network that is also composed of multiple convolutional layers. The decoder consists of a deflatten layer, and then four deconvolutional blocks, finally resulting in the reproduction of the original image. First, we de-flatten from $(C, H \times W)$ to (C, H, W), and then each deconvolutional block executes transposed-convolutional layer, followed by inverse-GDN and ReLU non-linearities. The last layer of the decoder uses Sigmoid as the activation function, which is interpreted as an image.

3.2 LSTM-based JSCC

One of the main drawbacks of regular deep neural-network-based JSCC schemes is that they predict a constant-sized vector to be transmitted through the channel. We imagine our input (from encoder $\mathcal{E}$) as a pseudo-sequence of embedded features from the image concatenated with information from receiver-side (CSI). Features from CNN of the size $(C, H \times W)$ are first considered as a sentence of C words of embedding dimension $H \times W$. Then we extend this representation in two ways: i) We extend the embedding dimension by adding SOS (start of sentence) and EOS (end of sentence) tokens on the first and last indices, making the new feature dimensions $(C, H \times W + 2)$, and ii) We transform the CSI of size (N_P, N_{FFT}), where N_P is the number of pilot packets and N_{FFT} is OFDM FFT size, to $(N_P, H \times W + 2)$ using a dense neural network. Finally, we concatenate both sources of information and have a final pseudo-sequence of size $(C + N_P, H \times W + 2)$. In order to feed this pseudo-sequence to the LSTM model, we use another dense layer to map onto the size $(C + N_P, h)$, where h is the hidden-size of the LSTM-layer. We then apply sequence-to-sequence model [25] to learn to transform this pseudo-sequence to another one which is robust to the channel. Just as languages have redundancy and have the ability to correct themselves in the presence of noise, we expect our *translator* to translate multi-scale features of our image into a language that is redundant enough to correct itself given

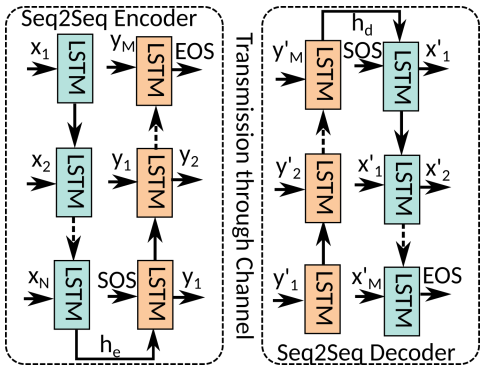

Figure 3: Proposed sequence-to-sequence encoder and decoder for the channel.

the current channel conditions. Since, a larger redundancy usually corresponds to a lengthier sentence, we expect a trade-off in terms of channel conditions and length of the latent vector, i.e., the worse the channel conditions, the longer the code-word to recover from the expected distortion. Our architecture is shown in Figure 3. Based on the Recurrent Neural Network (RNN)'s internal state, it is decided if more code words of hidden-size h should be output or if the sequence is finished. In this way, the final code-word has a length of $L = Nh$, where N differs for each channel condition and is learned during back-propagation. At the decoder, the received message (which is distorted due to the channel multi-path effects) is passed through the network, which performs the reverse translation task, i.e., converts the received message back to the multi-scale features then used to reconstruct the image. This network is also expected to perform as a channel decoder, i.e., correct errors in the received message by using redundancy as encoded by the RNN encoder on the transmitter side. For our experiments, h is equal to 1024 and $H = W = 50$ for an image-size of $(200, 200, 3)$.

3.3 Loss Function and Training

In order to train our multi-component neural network-based approach, we employ a complex loss function and training process ensuring correct training of the network. Our loss function is composed of four components in total, which are then added together to compose an aggregate loss function to be optimized. In these components, the first component is the Mean Squared Error (MSE) of the encoder-decoder network: $\mathcal{L}_{MSE}(y, \hat{y}) = \frac{1}{H \times W} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2$, where y_i and $\hat{y}_i$ are the i-th pixels of the input image y and reconstructed image $\hat{y}$ respectively. H and W denote the height and width of the image respectively. The second component of the loss function consists of the structural similarity index (SSIM) [31]. This metric is based on the assumption that human vision perceives structural information in a scene more robustly than the individual pixels. For this reason, it is modelled using luminance, contrast and structure common between two images (ground truth and reconstructed image). The structure is modeled using covariance matrix of both images. We add this metric into the loss function as well, as MSE alone is a poor indicator of how useful and clear an image is. Furthermore, we use the multi-scale version of this metric (MS-SSIM), which is defined in the range $[0, 1]$, and is directly proportional to image quality. In order to use it as a loss function, we define: $\mathcal{L}_{SIM}(y, \hat{y}) = 1 - \text{MSSSIM}(y, \hat{y})$.

The third component of our proposed loss function concerns itself with the length of the code-word generated for transmission across the channel. This component depends on two major factors: i) The length of the code-word transmitted through the channel must be as short as possible to facilitate the highest rate at which data can be transmitted, but at the same time, ii) The features must also be reconstructed fairly accurately, which requires a longer code-length, hence acting as a counterbalance. Let f be the multi-scale features input into the RNN, f' be the reconstructed features, and L be the length of the sequence being transmitted. The loss function is given as: $\mathcal{L}_{TR} = L||f - f'||_2^2$. Finally, the overall network is trained in the following manner: The CNN-based autoencoder is trained first and so is the RNN for the code generation using the losses $\mathcal{L}_{MSE}$ and $\mathcal{L}_{SIM}$ respectively. After these sub-networks are pre-trained, the final network is trained overall with a combination of the losses, which is, $\hat{\mathcal{L}} = \lambda_{MSE}\mathcal{L}_{MSE} + \lambda_{SIM}\mathcal{L}_{SIM} + \lambda_{TR}\mathcal{L}_{TR}$, where λ_{MSE}, λ_{SIM}, and λ_{TR} depend on the dataset being used.

Table 1: Tunable parameters in the underwater simulation for Separate Source-Channel Coding scheme.

Coding Type	Parameter	Tunable Values
Source Coding	Type	0 (JPEG), 1 (JP2)
	Quality	$\{q : q \in \mathbb{Z}; 1 \leq q \leq 100\}$
Channel Coding	Type	0 (Turbo), 1 (LDPC)
	Coding Rate	{1/2, 1/3, 1/4}
Transmission	Mod. Symbols	$2^m, m \in \{1, 2, 3, 4, 6\}$
	Ratio of D to P	{1,2,3,4,5}
	No. of Sub-carriers	{64, 128, 256, 512, 1024}

4 PERFORMANCE EVALUATION

We present in detail the comparison of our proposed method with three other baselines in this section, i) model-based disjoint parameter selection and joint data-driven parameter selection via ii) NN and iii) Reinforcement Learning (RL). First, we present our experimental setup, and then move on to the baseline comparison, and then show further experiments on the technical intricacies of our approach.

4.1 Experimental Setup

We employ both simulations and real-life testbed experiments as conducted on Rutgers University, New Brunswick, NJ premises, to test our approach and compare against several baselines. Below we detail our setup for both the simulations and experimental setup.

Simulations: In our simulations, the Rician channel is chosen to simulate the underwater channel. We set up this environment with the help of both MATLAB and Python. Underwater channel, source coding, channel coding, OFDM-based transmission, and channel estimation are all implemented in MATLAB while tuning algorithms are implemented in Python. Table 1 shows the parameters that could be tuned in order to get the best data rate under a given channel condition. Looking at the total number of customizable parameters, we can see that there could be a total of 150,000 possible Separate

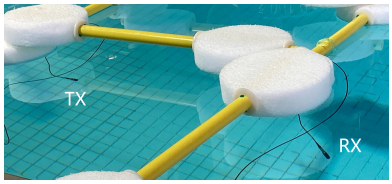

Figure 4: Testbed in the pool experiments at Sonny Werblin Recreation Center in July 2022. The depth of the pool is 4.0 ft, while the depths of the transducer (TX) and the hydrophone (RX) are equal to 0.8 ft. The distance is about 3.3 ft between the transducer and the hydrophone.

Source Channel Coding (SSCC) schemes. In the parameters, modulation symbols denote how many bits are encoded in each symbol, with $m = 1$ denoting the BPSK, $m = 2$ denoting the QPSK scheme and so on. Furthermore, each OFDM frame is composed of both data symbols D and pilot symbols P. The ratio of D to P denotes how many data symbols are transmitted for each data symbol in a frame. A higher ratio means a low number of pilots and hence, a weaker channel estimation at the receiver. Figs. 5 depicts the received BER and PSNR of JPEG and JPEG 2000 with different channel coding methods in simulated Rician channels versus normalized SNR (Eb/No). We can observe that when BER is higher than 10^{-4}, the received PSNR is very low and 'cliff effects' happen. We can also find that a low channel coding rate leads to low BER, and a high compression ratio leads to high PSNR. With the same compression ratio, the received image quality of JPEG 2000 is higher than that of JPEG, but the size of JPEG 2000 is larger than JPEG.

Pool experiments: Based on the simulation results, we further evaluate our proposal by conducting several rounds of pool experiments, based on a high-performance and scalable platform using a programmable Kintex-7 FPGA designed by Ettus Research Group with the NI Corporation, called Universal Software Radio Peripheral (USRP) X-300 [**?**]. Teledyne Marine RESON TC4013 omnidirectional transducers [**?**] with a frequency range from 50 to 150 kHz are used in our testbed. The specifications of the system are summarized in Table 2. In our experiments, the transducer and the hydrophone are placed in a large pool as shown in Figure 4. The image data is passed to the acoustic modem and transducer to be sent to the hydrophone on the other side of the link. The transmit power is adjusted mutually by power amplifier to get different levels of SNR. The transmission is then done with the symbol rate of 100 kBaud. The BER and Peak Signal-to-Noise Ratio (PSNR) performance of JSCC in the pool is shown in Figure 6. We can observe that the results in pool experiments are very close to those in simulated Rician channels. To mitigate the multipath effect as well as to enhance the spectrum efficiency, the OFDM modulation is applied in the underwater transmissions. The OFDM FFT size is chosen to be 6144. Given a bandwidth of 100 kHz, the symbol rate is 100 kBaud and the FFT duration is 6144/100 = 61.44 ms. We choose the cyclic prefix length to be 10.24 ms. Overall the OFDM symbol length is 61.44 + 10.24 = 71.68 ms, and the subcarrier spacing is 1/71.68ms = 16.28 Hz.

Datasets Used: For training our neural networks, we use both Underwater Image benchmark dataset [17] and a large dataset of our own collected underwater images using BlueROVs in Raritan river, NJ, US. For all our experiments, the input image-size is always set to (200, 200, 3) and any images that do not conform to this size are resized using Python Imaging Library (PIL). Furthermore, we use the channel taps (with multiple paths contributing to multiple taps) estimated and collected during tests conducted at Sonny Werblin Recreation Center (see Figure 2) for emulating the communication channel during training time.

Table 2: Hardware Specifications.

Part	Parameter	Value
Transducer	Frequency range	50–150 kHz (Omnidirectional)
	Receiving sensitivity	−211 dB ± 3 dB re 1 V/μPa
	Transmit sensitivity	130 dB ± 3 dB re 1 V/μPa
PreAmp.	Frequency (Gain)	0.5–500 kHz (0–50 dB)
	HP/LP filters	1 Hz–250 kHz/1 kHz–1 MHz
PowerAmp.	HP filters (Gain)	1 Hz–20 kHz (0–36 dB)
Modem	Mainboard	Kintex-7 FPGA
	Frequency (Clock)	0–30 MHz (10 MHz/1 PPS)
	ADC sample rate	1 channel, 200 MS/s (14 bits)
	DAC sample rate	1 channel, 800 MS/s (16 bits)

4.2 Comparison with AMC literature

We present here the comparisons with three baselines, namely, i) model-based disjoint parameter selection and joint data-driven parameter selection via ii) NN and iii) RL.

Model-based Disjoint Parameter Selection: In order to compare our technique to manual parameter selection which can be controlled by a tuner based on the feedback obtained from the receiver, we mapped this parameter selection problem as a classification problem. Hence, we trained a decision-tree classifier based on the approach presented in [19] for the parameters we stated in Table. 1. We first created a dataset with inputs of the dataset being CSI recovered from our simulations, and the ground truth being the index of a possible permutation of the parameters, which gave the best data-rate given that 37 packets are transmitted. That scheme is then labelled as the ground truth, and the decision-tree classifier is trained on this dataset. Figure 7(b) shows the comparison of this method with other methods. Given that the number of output schemes is high, the decision-tree model performs poorly because of a lack of data, and is not scalable as the number of available schemes increases.

NN-based Disjoint Parameter Selection: For this baseline, we trained a neural network classifier to predict the best-performing schemes for a given CSI, as proposed in [13], and labelled the dataset a little differently than for the model-based parameter selection. In this scenario, we labelled 5 top performing schemes for a given SNR value as the ground truth in order to compensate for less available data and increase the probability of guessing a 'good-enough' scheme. The NN architecture used is the following: a convolutional Layer with 32 output filters, a kernel size of 5, and a sigmoid activation, another convolutional layer with 90 output filters and a kernel size of 5, a flatten layer and finally a dense layer with a Sigmoid activation predicting probabilities of each class. Figure 7(b) shows the comparison of this method with other methods. Similar to the decision-tree model, this method is also not scalable as the number of available schemes increases.

RL-based Disjoint Parameter Selection: Another way to design a link-tuning algorithm is to let it experiment directly on a

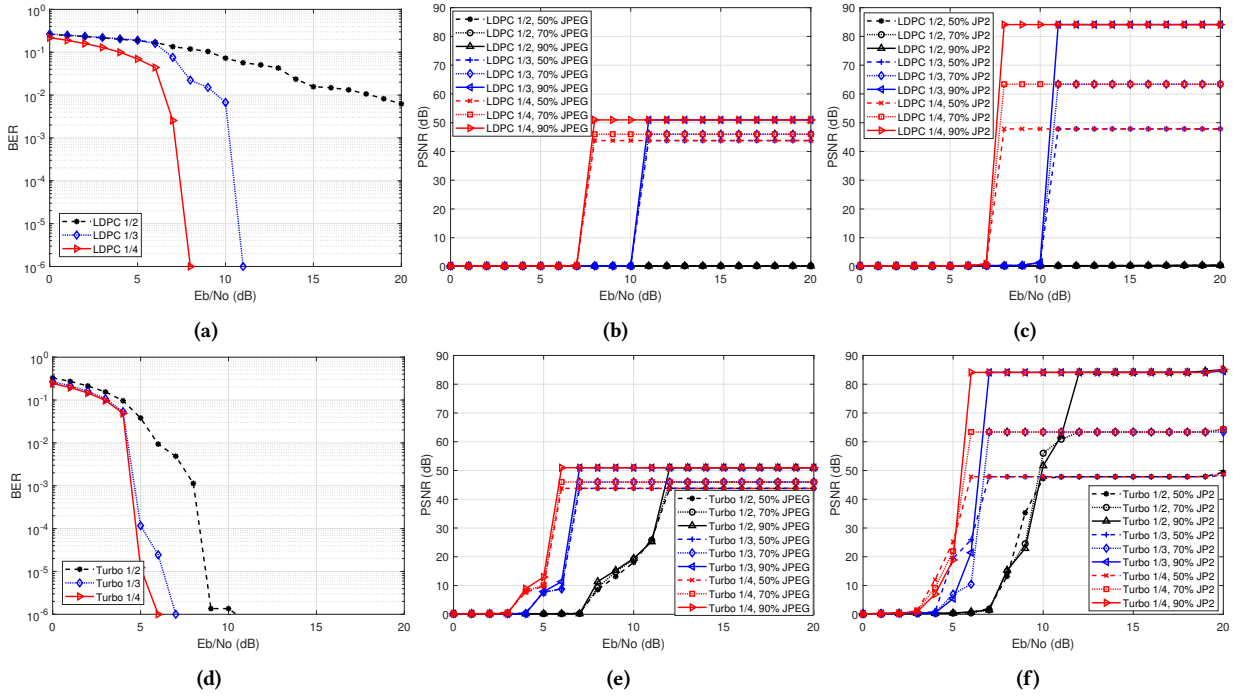

Figure 5: In simulated Rician channels with QPSK: (a) BER of LDPC coding with different code rates; (b) PSNR of received images compressed by JPEG with LDPC coding; (c) PSNR of received images compressed by JPEG 2000 (JP2) with LDPC coding; (d) BER of Turbo coding with different code rates; (e) PSNR of received images compressed by JPEG with Turbo coding; (f) PSNR of received images compressed by JPEG 2000 with Turbo coding.

live acoustic channel and then betters itself using the feedback it obtains using the average data-rate achieved and the BER, implicitly modelling the current channel conditions. This is the approach proposed by Shankar and Chitre [23], and we use it as a baseline in our own method. Since, our setup is slightly different than the one described in the paper, we adapt it slightly and focus on only the Dynamic Programming (DP) based solution, as it outperforms all the rest approaches according to their evaluation. The solution is based on the method proposed in [23]. This reward function is directly proportional to the data-rate achieved by a given scheme. The reward R for transmitting a frame s_i^P, while being in state ξ_t at time t is given by: $R(\xi_t, s_i^P) = \hat{\alpha}_{(j,c),t}\beta_j r_c e_c$. Here, ξ_t and s_i^P, namely agent's state and packet transmitted using scheme i, are defined in the same fashion as [23]. Furthermore, $\hat{\alpha}_{(j,c),t}$ denotes the estimated packet-success probability for a given scheme $i \equiv (j, c)$, β_j denotes the uncoded data-rate, and r_c denotes the information rate of the channel-coding scheme being used. One adaptation introduced by us to this formula is the parameter e_c, which is defined as the compression-to-clarity ratio. Therefore, we define the compression-to-clarity ratio as: $e_c = \log \frac{K}{\text{BPP}\times\text{MSE}}$. The distribution of this metric is shown in Figure 7(a), where both codecs' performance crosses each other in the mid-quality area, while JPEG2000 ultimately provides better performance at higher quality values. We use $K = 10$ for our results. We use this final reward formula to update our agent's value function $V_i^*(\xi_t)$ using the Bellman equation: The performance of the agent is shown in Figure 7(c), where across multiple runs, the agent steadily increases performance. In lower SNR channels, the agent does a lot of exploration, because of an overall lower probability of success, while at higher SNRs, the agent does well with exploring the space and discovering schemes with higher rewards. Figure 7(b) also shows the comparison of this method with other kinds of parameter selection. Overall, the RL method is scalable and adaptive but needs time to tune its reward functions. However, it may still result in sub-optimal performance as it only slowly explores the available search space. Finally, as shown in Figure 7(b), we compare the effective datarate achieved using all the different baselines, we observe that our JSCC scheme performs better than the disjoint NN and decision tree algorithms, while RL performs better. RL, however, takes a long time to converge for different SNR values (as shown in Figure 7(c)), while our approach achieves similar performance with a few transmissions.

5 CONCLUSION AND FUTURE WORK

We presented our data-driven scheme for JSCC in underwater acoustic channel using CNN-based feature extraction and a novel variable-length encoder and decoder design based on RNNs. The variable-length encoder-decoder design has the potential to adapt to changing underwater channel depending on the feedback received from the receiver. As future work, we would further conduct experiments to establish the efficacy of our approach, namely the quality of final images received, and do ablation studies to make it more efficient. We will also investigate how training on a specific channel data (pool) generalizes to acoustic communication on other channel conditions such as a bay, or the ocean. We also plan

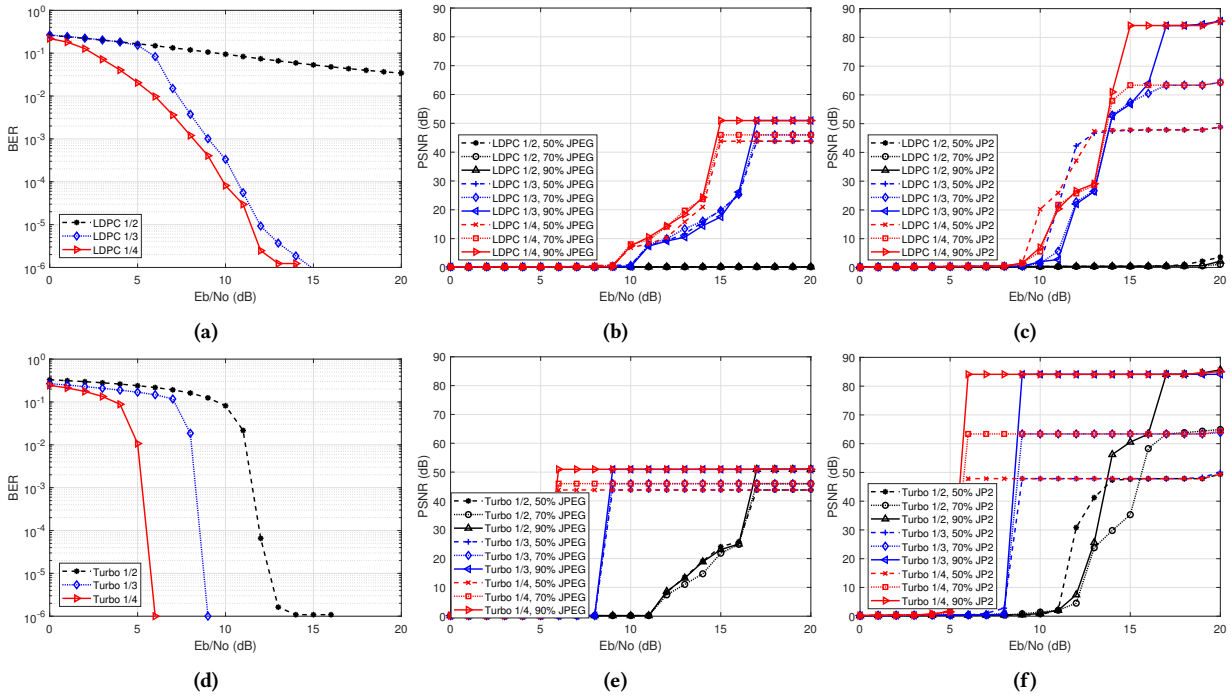

Figure 6: In pool experiments with QPSK: (a) BER of LDPC coding with different code rates; (b) PSNR of received images compressed by JPEG with LDPC coding; (c) PSNR of received images compressed by JPEG 2000 (JP2) with LDPC coding; (d) BER of Turbo coding with different code rates; (e) PSNR of received images compressed by JPEG with Turbo coding; (f) PSNR of received images compressed by JPEG 2000 with Turbo coding.

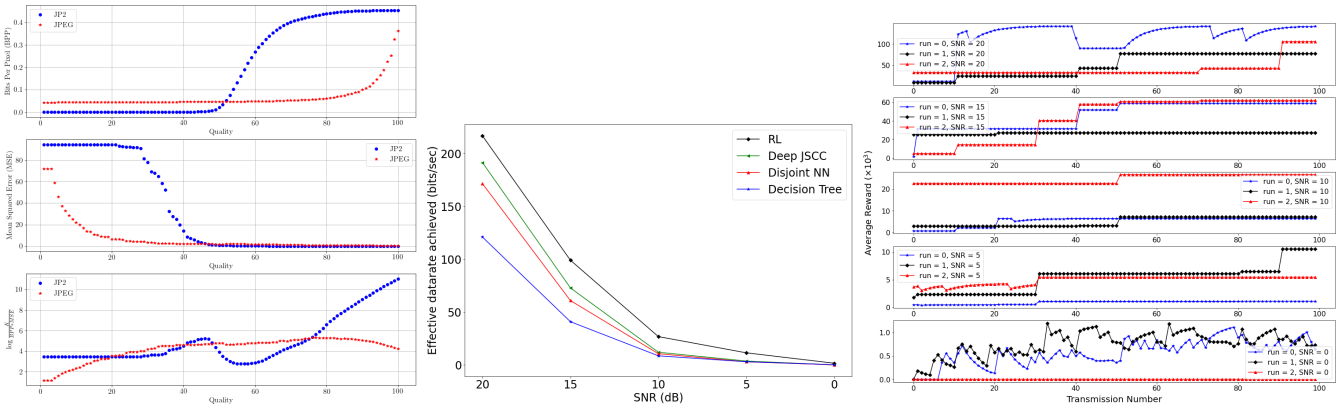

Figure 7: (a) Quality analysis of the source-coding algorithms (JPEG and JPEG2000), as tested on our own underwater dataset; (b) Comparison against various baseline schemes; (c) Average estimated data-rate achieved with the variation of SNR.

to conduct on-the-field experiments in the Barnaget bay, NJ, US which will comprise of multiple ROVs exploring the underwater environment adaptively sampling the area. We will then use this collected data to investigate the generalizability of our approach. Furthermore, we will also expand the scope of our work towards more multimedia data such as videos, which has the potential for enabling live acoustic underwater video-streams, and develop our data-driven encoding decoding technique further, by deploying it to underwater robots for acoustic communications.

ACKNOWLEDGMENTS

The authors thank Jeffrey Zeszotarski, aquatics coordinator at Rutgers University for his help and support during field experiments.

REFERENCES

[2]]Tc4013 [n. d.]. RESON TC4013 Hydrophone Product Information. http://www.teledynemarine.com/reson-tc4013. Accessed Feb 2, 2021.

[2]]USRP [n. d.]. USRP X Series. https://www.ettus.com. Accessed Feb 2, 2021.

[3] Hiroaki Akutsu, Akifumi Suzuki, Zhisheng Zhong, and Kiyoharu Aizawa. 2020. Ultra Low Bitrate Learned Image Compression by Selective Detail Decoding. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, Seattle, WA, USA, 524–528. https://ieeexplore.ieee.org/document/9150712/

[4] Khizar Anjum, Zhile Li, and Dario Pompili. 2022. Acoustic Channel-aware Autoencoder-based Compression for Underwater Image Transmission. In *The Sixth Underwater Communications and Networking Conference (UComms)*. 1–4.

[5] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. 2015. Density modeling of images using a generalized normalization transformation. *arXiv preprint arXiv:1511.06281* (2015).

[6] Yuting Bao, Yuwen Tao, and Pengjiang Qian. 2022. Image Compression Based on Hybrid Domain Attention and Postprocessing Enhancement. *Computational Intelligence and Neuroscience* 2022 (March 2022), 1–12. https://www.hindawi.com/journals/cin/2022/4926124/

[7] Eirina Bourtsoulatze, David Burth Kurka, and Deniz Gündüz. 2019. Deep Joint Source-Channel Coding for Wireless Image Transmission. *IEEE Transactions on Cognitive Communications and Networking* 5, 3 (Sept. 2019), 567–579. Conference Name: IEEE Transactions on Cognitive Communications and Networking.

[8] Zhengxue Cheng, Ting Fu, Jiapeng Hu, Li Guo, Shihao Wang, Xiongxin Zhao, Dajiang Zhou, and Yang Song. 2021. Perceptual Image Compression using Relativistic Average Least Squares GANs. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, Nashville, TN, USA, 1895–1900. https://ieeexplore.ieee.org/document/9522791/

[9] IEEE Computer Society LAN/MAN Standards Committee et al. 2007. IEEE Standard for Information Technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. *IEEE Std 802.11* (2007).

[10] Henry Dol, Koen Blom, Paul van Walree, Roald Otnes, Håvard Austad, Till Wiegand, and Dimitri Sotnik. 2020. Adaptivity at the Physical Layer. In *Cognitive Underwater Acoustic Networking Techniques*, Dimitri Sotnik, Michael Goetz, and Ivor Nissen (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 13–40.

[11] Iñaki Estella Aguerri and Deniz Gündüz. 2016. Joint Source-Channel Coding With Time-Varying Channel and Side-Information. *IEEE Transactions on Information Theory* 62, 2 (Feb. 2016), 736–753. Conference Name: IEEE Transactions on Information Theory.

[12] Fredrik Hekland, Pal Anders Floor, and Tor A. Ramstad. 2009. Shannon-kotelnikov mappings in joint source-channel coding. *IEEE Transactions on Communications* 57, 1 (2009), 94–105.

[13] Lihuan Huang, Yue Wang, Qunfei Zhang, Jing Han, Weijie Tan, and Zhi Tian. 2022. Machine Learning for Underwater Acoustic Communications. *IEEE Wireless Communications* (2022), 1–8. Conference Name: IEEE Wireless Communications.

[14] Lihuan Huang, Qunfei Zhang, Weijie Tan, Yue Wang, Lifan Zhang, Chengbing He, and Zhi Tian. 2020. Adaptive modulation and coding in underwater acoustic communications: a machine learning perspective. *EURASIP Journal on Wireless Communications and Networking* 2020, 1 (Oct. 2020), 203.

[15] Hovannes Kulhandjian and Tommaso Melodia. 2014. Modeling Underwater Acoustic Channels in Short-Range Shallow Water Environments. In *Proceedings of the International Conference on Underwater Networks & Systems* (Rome, Italy). 1–5.

[16] David Burth Kurka and Deniz Gündüz. 2020. *DeepJSCC-f: Deep Joint Source-Channel Coding of Images with Feedback*. Technical Report arXiv:1911.11174. arXiv. http://arxiv.org/abs/1911.11174 arXiv:1911.11174 [cs, eess, math, stat] type: article.

[17] Chongyi Li, Chunle Guo, Wenqi Ren, Runmin Cong, Junhui Hou, Sam Kwong, and Dacheng Tao. 2019. An underwater image enhancement benchmark dataset and beyond. *IEEE Transactions on Image Processing* 29 (2019), 4376–4389.

[18] Fabian Mentzer, George Toderici, Michael Tschannen, and Eirikur Agustsson. 2020. *High-Fidelity Generative Image Compression*. Technical Report arXiv:2006.09965. arXiv. http://arxiv.org/abs/2006.09965 arXiv:2006.09965 [cs, eess] type: article.

[19] Konstantinos Pelekanakis, Luca Cazzanti, Giovanni Zappa, and João Alves. 2016. Decision tree-based adaptive modulation for underwater acoustic communications. In *2016 IEEE Third Underwater Communications and Networking Conference (UComms)*. 1–5.

[20] Roberto Petroccia, Pietro Cassarà, and Konstantinos Pelekanakis. 2019. Optimizing Adaptive Communications in Underwater Acoustic Networks. In *OCEANS 2019 MTS/IEEE SEATTLE*. 1–7. ISSN: 0197-7385.

[21] Andreja Radosevic, Rameez Ahmed, Tolga M. Duman, John G. Proakis, and Milica Stojanovic. 2014. Adaptive OFDM Modulation for Underwater Acoustic Communications: Design Considerations and Experimental Results. *IEEE Journal of Oceanic Engineering* 39, 2 (April 2014), 357–370. Conference Name: IEEE Journal of Oceanic Engineering.

[22] Andreja Radosevic, John G. Proakis, and Milica Stojanovic. 2009. Statistical characterization and capacity of shallow water acoustic channels. In *OCEANS 2009-EUROPE*. 1–8.

[23] Satish Shankar and Mandar Chitre. 2013. Tuning an underwater communication link. In *2013 MTS/IEEE OCEANS - Bergen*. 1–9.

[24] Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal* 27, 3 (1948), 379–423.

[25] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. https://arxiv.org/abs/1409.3215

[26] D.S. Taubman and M.W. Marcellin. 2002. JPEG2000: standard for interactive imaging. *Proc. IEEE* 90, 8 (Aug. 2002), 1336–1357. Conference Name: Proceedings of the IEEE.

[27] Tze-Yang Tung, David Burth Kurka, Mikolaj Jankowski, and Deniz Gunduz. 2022. DeepJSCC-Q: Constellation Constrained Deep Joint Source-Channel Coding. *arXiv preprint arXiv:2206.08100* (2022).

[28] Paul A. van Walree. 2013. Propagation and Scattering Effects in Underwater Acoustic Communication Channels. *IEEE Journal of Oceanic Engineering* 38, 4 (2013), 614–631.

[29] S. Vembu, S. Verdu, and Y. Steinberg. 1995. The source-channel separation theorem revisited. *IEEE Transactions on Information Theory* 41, 1 (1995), 44–54.

[30] G.K. Wallace. 1992. The JPEG still picture compression standard. *IEEE Transactions on Consumer Electronics* 38, 1 (Feb. 1992), xviii–xxxiv. Conference Name: IEEE Transactions on Consumer Electronics.

[31] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (2004), 600–612.

[32] Jialong Xu, Bo Ai, Ning Wang, and Wei Chen. 2022. *Deep Joint Source-Channel Coding for CSI Feedback: An End-to-End Approach*. Technical Report arXiv:2203.16005. arXiv. http://arxiv.org/abs/2203.16005 arXiv:2203.16005 [cs, eess, math] type: article.

[33] Jintao Yan, Jianhao Huang, and Chuan Huang. 2021. Deep Learning Aided Joint Source-Channel Coding for Wireless Networks. In *2021 IEEE/CIC International Conference on Communications in China (ICCC)*. 805–810. ISSN: 2377-8644.