An End-to-End Scheme for Learning Over Compressed Data Transmitted Through a Noisy Channel

Alireza Tasdighi, Elsa Dupraz

An End-to-End Scheme for Learning over Compressed Data Transmitted Through a Noisy Channel

Alireza Tasdighi and Elsa Dupraz
IMT Atlantique, Lab-STICC, UMR CNRS 6285, France

Abstract

Within the emerging area of goal-oriented communications, this paper introduces a novel end-to-end transmission scheme dedicated to learning over a noisy channel, under the constraint that no prior training dataset is available. In this scheme, the transmitter makes use of powerful Spherical Harmonic Transform and Irregular Hexagonal Quadratic Amplitude Modulation techniques, while the receiver relies on a Complex-Valued Neural Network (CVNN) so as to realize the learning task onto the received noisy data. As a main feature of the proposed scheme, the transmitter is fixed and does not depend on the source statistics, while the receiver is trained from a first data transmission phase, thus providing an efficient transmission-versus-learning approach under the considered constraint. The proposed transmission scheme may be adapted to a variety of learning problems, and the paper specifically investigates clustering and classification, two very common learning tasks. In the last part of the paper, the source/channel coding rate of the proposed transmission scheme is evaluated theoretically and from numerical simulations. This analysis shows a clear advantage in terms of coding rate of our scheme compared to conventional coding approaches, when targeting the same learning performance level.

Index Terms

Data Transmission, Source Coding, Machine Learning, Neural Networks, Clustering, Classification

I. Introduction

Conventional schemes for data transmission over a noisy channel are designed so as to reconstruct the original information sequence without error (lossless transmission) or with a residual amount of errors (lossy transmission). However, in many applications, the objective of the receiver is not to reconstruct the original data, but rather to apply a given learning task onto the received data. As examples, one may consider health disease detection from human body sensors [1], [2], underwater activity monitoring [3], or traffic flow prediction with autonomous vehicles [4]. The problem of learning over data transmitted through a noisy channel falls into the emerging area of goal-oriented communications, and was identified as a key functionality to be integrated in the upcoming 6G standard [5].

In this paper, we consider that several sensors transmit their data to a fusion center whose objective is to apply a certain learning task onto the sensors measurements. In this context, conventional source/channel coding schemes targeting data reconstruction are known to be sub-optimal in terms of amount of data to be transmitted so as to achieve a certain learning performance [6], [7]. An alternative to conventional coding approaches consists of replacing the transmitter and the receiver by Deep Neural Networks (DNNs) trained so as to realize the learning task while taking into account the channel effect onto the transmitted data [8], [9]. However, this approach can only be implemented if an initial training dataset is available for pre-training, or if there exists some reliable feedback link allowing for significant data transmission from the receiver to the transmitter. Otherwise, the emitter should have enough resources in terms of data and power to train the DNN, or the receiver may perform the training, but it should then send back the weights to the encoder through the feedback link. On the opposite, this paper addresses the design of a practical coding scheme dedicated to learning, considering that: (i) the sensors do not have enough resources to perform the training, (ii) no initial training set is available, (iii) the feedback link only allows for limited data transmission. When considering these three constraints, the key challenges reside in devising a fixed transmitter able to work by itself with only a few feedback from the receiver, and in developing a receiver dedicated to learning and capable of online adaptation to the source and channel statistics.

In the literature, the problem of designing source and channel coding schemes dedicated to learning was first addressed from the theoretical point of view of Information Theory. In this field, the most considered learning problem was by far Distributed Hypothesis Testing (DHT) in which the receiver should decide between two hypothesis related to the statistics of two sources X and Y [10]–[13]. These works provided error-exponents for DHT under rate-limited transmission links and under various transmission setups (perfect and non-perfect channel, relaying opportunities, etc.). Apart from DHT, [6] identified a trade-off between content identification and data reconstruction from a noisy database, while [14] addressed parameter estimation over compressed data. Finally, [15] considered the problem of supervised learning of a given function f from compressed observations. Although all the above works provide meaningful insights on how to perform learning over compressed data, they are mostly theoretical and do not provide any practical code design solution.

On the practical code design side, several works proposed to perform parameter estimation [16], hypothesis testing [17], or clustering [18], [19], from only a small amount of linear combinations of the input data, following the Compressed Sensing (CS) approach [20]. However, these works are not directly suitable for digital data transmission, since they produce real-valued data and do not evaluate the effect of quantization or channel noise onto the learning performance. As an attempt to develop discrete CS approaches for learning, [21], [22] considered parameter estimation over Low Density Parity Check (LDPC) codes, and [23] investigated clustering over LDPC codes. But the above works considered only discrete sources, and they assumed a perfect (noiseless) transmission channel.

In this paper, we propose a full end-to-end transmission scheme dedicated to learning. The proposed scheme does not rely on any prior knowledge of the source statistics and can therefore be adapted online to the collected data. We first introduce a transmitter scheme which can be rate-adapted so as to ensure a good learning performance. This transmitter is built with Spherical Harmonic Transforms (SHF) [24] and Irregular Hexagonal Quadratic Amplitude Modulation (IHQAM) [25]. These two techniques used together allow us to preserve the data structure after channel transmission, in order to efficiently apply learning after only a few reconstruction operations at the receiver. We then propose a receiver dedicated to learning and built from a Complex-Valued Neural Network (CVNN) [26] trained from a first data transmission phase. One main advantage of the proposed strategy is that the transmitter does not depend on the considered learning task. Therefore, we consider two learning problems that are clustering and classification, and provide two versions (both based on CVNN) of the receiver, depending on the considered problem. We aim to evaluate the effectiveness of the proposed approach for these two problems.

For that purpose, the second part of the paper is dedicated to the performance evaluation of our scheme. Since standard performance metrics usually considered for data reconstruction (error probability, distortion, etc.) are not suitable in our context, we first identify metrics of interest which can be used to evaluate the learning performance of the proposed scheme. We then evaluate the source-channel coding rate of our scheme, as a function of its parameters. Unfortunately, we cannot compare this coding rate to any available Information-Theoretic achievable coding rate. Indeed, no such theoretical result exists for clustering or classification, and given the few learning tasks considered in the literature of Information Theory, this appears to be a difficult problem. This is why, here, we employ a more pragmatic approach and evaluate the coding rates of two identified baseline schemes: one theoretical and one more practical, the latter consisting of a conventional coding scheme. These two baselines will allow to position the performance of our scheme with respect to other potential approaches.

Finally, we run numerical simulations to evaluate the learning performance of the proposed scheme, and compare the coding rate of our scheme with respect to the two identified baselines. As a main result, we observe a significant gain in coding rate compared to conventional coding schemes, while maintaining the same learning performance.

The outline of the paper is as follows. Section II introduces our notation and assumptions for the problem of learning over transmitted data. Section III describes the coding scheme at the transmitter. Section IV introduces the learning scheme at the receiver. Section V evaluates the coding rate and learning performance of the proposed scheme. Finally, Section VI provides numerical results.

II. System Description

This section introduces the notation and main assumptions of our work, and presents the problem of learning over data transmitted through a noisy channel. In what follows, $[\![1, N]\!]$ denotes the set of integers from 1 to N.

A. Source and channel models

We consider a setup in which a potentially large number of sensors collect data to be transmitted to a fusion center. We assume that each sensor has access to several pieces of data, each denoted with bold-letter $\mathbf{X}_s$, where $s \in [\![1, S]\!]$, and we let $\{\mathbf{X}_s\}_{s\in[\![1,S]\!]}$ be the full dataset. We consider that each piece of data $\mathbf{X}_s$ is a matrix of size $N \times M$. This corresponds to two-dimensional data such as images, although the proposed scheme could be adapted to one-dimensional data such as measurement vectors or time series. We do not make any assumption on the source statistics, in order to develop an agnostic coding scheme which can adapt to a wide range of situations. In addition, we consider that the data is transmitted through an Additive White Gaussian Noise (AWGN) channel with variance σ^2, a common assumption in the study of communication systems.

B. Learning tasks

The objective of the fusion center is to apply a given learning task over the dataset $\{\mathbf{X}_s\}_{s\in[\![1,S]\!]}$ collected by the sensors. Clustering is an unsupervised learning task (*e.g.* no labelled data is available for training), while classification is a supervised learning task (labelled data is required for training). This will allow us to evaluate the performance of the proposed transmission scheme over two learning tasks which are very different by nature. We now briefly introduce these two tasks.

1) Clustering: Clustering consists of separating the dataset $\{\mathbf{X}_s\}_{s\in[\![1,S]\!]}$ into clusters, such that data in a cluster are similar with each other. In this work, we consider the Euclidian distance $d(\mathbf{X}_s, \mathbf{X}_{s'})$ as the similarity measure between two data $\mathbf{X}_s$ and $\mathbf{X}_{s'}$. We further consider the very popular clustering algorithm K-means [27] since our purpose is not to introduce a new clustering method, but rather to work on the design of the transmission system. The K-means algorithm requires the knowledge of the number of clusters K, and aims to minimize the following cost function:

$$J = \sum_{s=1}^{S} \sum_{k=1}^{K} c_{s,k} d(\mathbf{X}_s, \boldsymbol{\theta}_k) \tag{1}$$

with respect to cluster assignments $c_{s,k} \in \{0, 1\}$ and to cluster centroids $\boldsymbol{\theta}_k \in \mathbb{R}^{N\times M}$. K-means usually suffers from initialization issues and from the fact that the number K of clusters can be difficult to know in advance. We refer the reader to [28], [29] for methods to solve these two issues. These methods consist of simple modifications of the K-means algorithm, and could be easily incorporated in our approach.

2) Classification: Classification consists of assigning data to one of K pre-defined classes. It is realized in two phases. In the first *training* phase, a classifier is trained from a set of labelled data, where labels indicate to which class belongs each data. In the second *inference* phase, the classifier should correctly assign a new unlabelled data to the correct class. Among various methods that exist for classification, we here consider standard feedforward Neural Networks (NN) for their efficiency and adaptability [30]. In what follows, and as commonly done in classification, we will consider that the NN is trained by considering the cross-entropy as loss function [31].

C. *Transmission scheme for learning*

When developing our transmission scheme, we will consider that we do not have access to any prior training dataset, since constructing in advance such a training set is not always possible in practical transmission scenario. Therefore, in order to develop an efficient transmission scheme dedicated to learning, we consider two transmission phases. At the first phase, a fraction β of the dataset $\{\mathbf{X}_s\}_{s\in[\![1,S]\!]}$ is transmitted to the fusion center by using a conventional lossless or lossy data transmission scheme. This first data transmission will allow the fusion center to properly calibrate the learning algorithm. This is of special importance in our context, since we want the transmission scheme to be as agnostic as possible with respect to the source statistics. The second phase, which constitutes the main contribution of this paper, will be specifically designed so as to allow for learning over the transmitted data, without need to reconstruct the original data. Note that both phases will be taken into account when evaluating the overall source/channel coding rate of the proposed scheme.

In addition, we consider a feedback channel between the fusion center and the sensors. Through the feedback channel, the fusion center will get the sensors informed about the amount of data they should transmit at the second phase so as to achieve a good learning performance. However, we will pay special attention to only sending a few amount of information through the feedback channel, since setting up such down-link connection can be very costly in practical applications.

D. *Existing approaches for learning over transmitted data*

Before describing our proposed transmission scheme for learning over received data, we review existing approaches for this problem, and identify their limitations.

A first straightforward approach would consist of considering a conventional data transmission scheme targeting data reconstruction. In this case, it is shown in [32] that there is no need to completely reconstruct the data before applying the learning task. For instance, [33] proposed to train a Deep Neural Network (DNN) dedicated to classification directly in the JPEG transform domain. This solution is especially appropriate in the case where the data is already compressed and *e.g.*, stored in a dedicated server. However, in our case, it may be very sub-optimal in terms of coding rate. As a matter of fact, an information-theoretic analysis carried in [11] shows that the rate needed for DHT is much lower than the rate needed for data reconstruction. The same fact was also empirically observed in [23] for clustering over compressed data. In addition, and perhaps more surprisingly, this approach may also be sub-optimal in terms of learning performance. For instance, it is shown in [6] that there exists a trade-off in terms of coding rate between data reconstruction and identification. It is also shown in [34] that the classification performance after video compression and decompression is poor when low bitrates are considered. This shows the need to design a coding scheme fully dedicated to learning.

Alternatively, one could consider the use of full end-to-end Deep-Learning techniques, which were widely investigated in the telecommunication field recently, for data compression [8], [35], noisy channel transmission [9], [36], or joint source-channel coding [37]–[39]. Some of these solutions were extended to target learning problems such as image retrieval [40], image classification [41], or image recognition [42]. Most of these solutions can be seen as Variational Auto-Encoders (VAEs). VAEs are composed of one encoder, which produces a latent vector, and of one decoder, which may either perform data reconstruction [8], [9], [35]–[39], or apply some specific learning tasks [40]–[43] onto the latent vector. Usually, the VAE encoder and decoder are constructed from NNs. When no training set is available, the emitter may train its own NN [42], [44], given that it has enough computation and power resources. Otherwise, the training algorithm should be applied at the receiver, and the updated NN weights or the loss function for each training sample [9] should be transmitted back to the encoder via a reliable feedback channel. Therefore, the use of VAEs seems unrealistic in all the applications in which it is not possible to make such an intensive use of the feedback channel.

This is why in this work we do not consider the Deep-Learning based approach neither. Instead, we will develop a transmission scheme in which the transmitter scheme is fixed and does not need to be updated online, while the receiver can make use of a NN dedicated to the considered learning task. This NN will be trained during the first transmission phase, with no need to send back any training information to the encoder.

In what follows, we first introduce the proposed transmitter scheme (Section III), and then

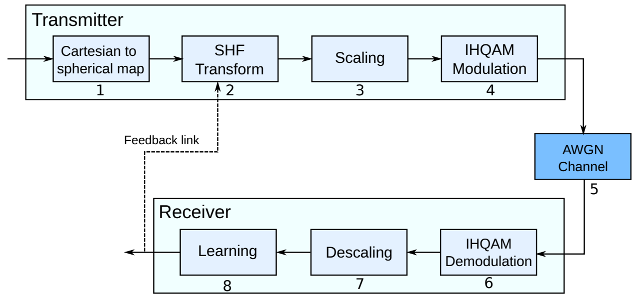


Fig. 1. Transmitter and Receiver scheme for Learning

describe the proposed receiver learning scheme (Section IV).

III. Data Transmission Scheme

In conventional data transmission schemes, the source-channel separation theorem states that designing the source coding scheme and the channel coding scheme independently from each other is optimal, at least in the asymptotic regime. However, this result most probably does not hold anymore when targeting learning. For instance, [10] proposes a joint source-channel code design for DHT which achieves better performance than the separate design. Intuitively, since learning algorithms are designed so as to handle the noise within the data, they should also show some robustness against channel noise. Therefore, it may be irrelevant to put some effort into completely correcting the channel noise before applying the learning algorithm. Following this idea, we build-up an unconventional data transmission scheme which avoids both standard lossless source coding (Huffman, Lempel Ziv, etc.) and standard channel coding (LDPC codes, Turbo codes, etc.). The proposed scheme is designed so as to preserve the data structure during channel transmission, so that the learning algorithm can directly handle the additional noise introduced by the channel. Note that in the context of lossy data reconstruction through a noisy channel, it was shown in [45] that transmitting uncoded data is optimal for some sets of source and channel pairs. However, the work of [45] was mostly theoretical and did not address learning.

Figure 1 shows the generic coding scheme (transmitter + receiver) we propose in this paper. In Figure 1, we see that the transmitter is composed of three main blocks: transform coding, scaling, and modulation, which we now describe.

A. SHF transform coding

As in most source coding approaches, our scheme first employs a transform coding operation. Here, we consider Spherical Harmonic Functions (SHF's) [24], which are known to be very good function approximators, mostly due to their polynomial forms. Transforms usually employed in source coding, like Discrete Cosine Transform (DCT), are real-valued for easier use. Here, on the opposite, we choose to employ SHF because it is a 2D complex transform. This will interface better with the modulation method considered in our scheme, since the constellation of this modulation is defined in the complex domain.

Consider a given matrix $\mathbf{X}$ of size $M \times N$ from the dataset $\{\mathbf{X}_s\}_{s \in [\![1,S]\!]}$. We use $X_{m,n}$ to denote the coefficient at position $(m, n) \in [\![0, M-1]\!] \times [\![0, N-1]\!]$ in the matrix $\mathbf{X}$. In order to apply SHF, Step $\mathbf{1}$ of Figure 1 converts the Cartesian coordinates $X_{m,n} = (m, n)$ into a spherical coordinate system. To do so, we set $\theta_m = \frac{\pi m}{M}$ and $\varphi_n = \frac{2\pi n}{N}$, where θ_m is the zenith (polar) angle such that $0 < \theta_m < \pi$, and φ_n is the azimuthal angle such that $0 < \varphi_n < 2\pi$.

Then, the SHF is defined from Legendre polynomials $P_\ell^{(k)} : [-1, 1] \to \mathbb{R}$ given as

$$P_\ell^{(k)}(u) = \frac{(-1)^k \sqrt{(1-u^2)^k}}{2^\ell \ell!} \left(\frac{d}{du}\right)^{\ell+k} \left(u^2 - 1\right)^\ell, \tag{2}$$

with the convention that $P_\ell^{(-k)} = (-1)^k \, P_\ell^{(k)}$, and where $\ell \in [\![0, +\infty]\!]$ and $k \in [\![0, \ell]\!]$. Legendre polynomials define an orthogonal basis. In addition, spherical harmonic functions $Y_\ell^{(k)}$ ($\ell \in [\![0, +\infty]\!]$, $k \in [\![0, \ell]\!]$) define an orthogonal-basis system that maps the spherical coordinates to scalar complex values as follow:

$$\begin{aligned} Y_\ell^{(k)}\left(\theta_m, \varphi_n\right) &= N_\ell^{(k)} P_\ell^{(k)}\left(\cos\theta_m\right) e^{ik\varphi_n} \\ &= N_\ell^{(k)} P_\ell^{(k)}\left(\cos\theta_m\right)\left(\cos\left(k\varphi_n\right) + i\sin\left(k\varphi_n\right)\right) \end{aligned} \tag{3}$$

where $N_\ell^{(k)}$ is a normalization constant defined as

$$N_\ell^{(k)} = \sqrt{\frac{(2\ell+1)}{4\pi} \frac{(\ell-k)!}{(\ell+k)!}}. \tag{4}$$

with the convention that $N_\ell^{(-k)} = N_\ell^{(k)}$. It can be shown that the functions $Y_\ell^{(k)} : \mathbb{R}^2 \to \mathbb{C}$ satisfy standard orthonormal conditions restated in [24]. All the expressions of the SHFs $Y_\ell^{(k)}$ can be found in [46].

Finally, in Step **2** of Figure 1, for all $\ell \in [\![0, +\infty]\!]$ and $k \in [\![0, \ell]\!]$, the transform coefficients $C_\ell^{(k)}$ are calculated from the functions $Y_\ell^{(k)}$ as

$$C_\ell^{(k)} = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X_{m,n} Y_\ell^{(k)}(\theta_m, \varphi_n). \tag{5}$$

In order to simplify notation in the following, we re-index the SHFs and transform coefficients as $Y_\ell^{(k)} := Y_p$ and $C_\ell^{(k)} := C_p$, and retain only P coefficients C_p. To do so, the functions $Y_\ell^{(k)}$ are ordered by taking first the function $Y_0^{(0)}$ for $\ell = 0$, then all functions $Y_1^{(k)}$ for $\ell = 1$, then all functions $Y_2^{(k)}$ for $\ell = 2$, and so on, until P functions have been taken. As a result, for all $(m, n) \in [\![0, M-1]\!] \times [\![0, N-1]\!]$, P-order approximations of the matrix coefficients $X_{m,n}$ are given by

$$X_{m,n} = \sum_{p=0}^{P-1} C_p Y_p(\theta_m, \varphi_n). \tag{6}$$

Equation (6) shows that the choice of the value of P is critical for the quality of the approximation. This choice will be discussed later on in the paper.

Since our scheme does not include channel coding, the P transform coefficients C_p are passed to the modulation step, after an intermediate scaling step.

B. Modulation

For modulation, we consider Irregular Hexagonal QAM (IHQAM) constellation [47, Section V], which is a very energy-efficient 2D signal constellation method. Here and for the sake of brevity, we only describe the 64-IHQAM constellation. In our simulations, we only considered the 64-IHQAM for its high reliability, but our scheme can be straightforwardly adapted to other constellation orders.

In a 2D signal constellation, the Symbol Error Rate (SER) is mainly affected by the minimum distance between two neighboring constellation points, and by the average symbol energy which depends upon the mean squared distance between constellation points and the origin. Therefore, in the optimum 2D hexagonal lattice based IHQAM constellation [47], constellation points are situated on concentric discs and the minimum distance separation of any two adjacent points is $2d$. Further, according to [47], real (resp. imaginary) coordinates of each constellation point are integer coefficients of d (resp. $\sqrt{3}d$).

TABLE I

EQUATIONS OF THE LINES THAT DEFINE DECISION BOUNDARIES IN 64-IHQAM. THE PROPOSED EQUATIONS HERE CORRECT SOME TYPOS IN TABLE V OF [47].

	R_9, R_8	R_{10}, R_7	R_{11}, R_6	R_{12}, R_5	R_{13}, R_4	R_{14}, R_3	R_{15}, R_2	R_{16}, R_1
Im(P_1)	$\frac{11d \mp x}{\sqrt{3}}$	$\frac{9d \pm x}{\sqrt{3}}$	$\frac{13d \mp x}{\sqrt{3}}$	$\frac{7d \pm x}{\sqrt{3}}$	$\frac{7d \pm x}{\sqrt{3}}$	–	–	–
Im(P_2)	$\frac{7d \pm x}{\sqrt{3}}$	$\frac{9d \mp x}{\sqrt{3}}$	$\frac{5d \pm x}{\sqrt{3}}$	$\frac{11d \mp x}{\sqrt{3}}$	$\frac{3d \pm x}{\sqrt{3}}$	$\frac{13d \mp x}{\sqrt{3}}$	$\frac{d \pm x}{\sqrt{3}}$	$\frac{d \pm x}{\sqrt{3}}$
Im(P_3)	$\frac{5d \mp x}{\sqrt{3}}$	$\frac{3d \pm x}{\sqrt{3}}$	$\frac{7d \mp x}{\sqrt{3}}$	$\frac{d \pm x}{\sqrt{3}}$	$\frac{9d \mp x}{\sqrt{3}}$	$\frac{-d \pm x}{\sqrt{3}}$	$\frac{11d \mp x}{\sqrt{3}}$	$\frac{-3d \pm x}{\sqrt{3}}$
Im(P_4)	$\frac{d \pm x}{\sqrt{3}}$	$\frac{3d \mp x}{\sqrt{3}}$	$\frac{-d \pm x}{\sqrt{3}}$	$\frac{5d \mp x}{\sqrt{3}}$	$\frac{-3d \pm x}{\sqrt{3}}$	$\frac{7d \mp x}{\sqrt{3}}$	$\frac{-5d \pm x}{\sqrt{3}}$	$\frac{9d \mp x}{\sqrt{3}}$
Im(P_5)	$-\left(\frac{d \pm x}{\sqrt{3}}\right)$	$-\left(\frac{3d \mp x}{\sqrt{3}}\right)$	$-\left(\frac{-d \pm x}{\sqrt{3}}\right)$	$-\left(\frac{5d \mp x}{\sqrt{3}}\right)$	$-\left(\frac{-3d \pm x}{\sqrt{3}}\right)$	$-\left(\frac{7d \mp x}{\sqrt{3}}\right)$	$-\left(\frac{-5d \pm x}{\sqrt{3}}\right)$	$-\left(\frac{9d \mp x}{\sqrt{3}}\right)$
Im(P_6)	$-\left(\frac{5d \mp x}{\sqrt{3}}\right)$	$-\left(\frac{3d \pm x}{\sqrt{3}}\right)$	$-\left(\frac{7d \mp x}{\sqrt{3}}\right)$	$-\left(\frac{d \pm x}{\sqrt{3}}\right)$	$-\left(\frac{9d \mp x}{\sqrt{3}}\right)$	$-\left(\frac{-d \pm x}{\sqrt{3}}\right)$	$-\left(\frac{11d \mp x}{\sqrt{3}}\right)$	$-\left(\frac{-3d \pm x}{\sqrt{3}}\right)$
Im(P_7)	$-\left(\frac{7d \pm x}{\sqrt{3}}\right)$	$-\left(\frac{9d \mp x}{\sqrt{3}}\right)$	$-\left(\frac{5d \pm x}{\sqrt{3}}\right)$	$-\left(\frac{11d \mp x}{\sqrt{3}}\right)$	$-\left(\frac{3d \pm x}{\sqrt{3}}\right)$	$-\left(\frac{13d \mp x}{\sqrt{3}}\right)$	$-\left(\frac{d \pm x}{\sqrt{3}}\right)$	$-\left(\frac{d \pm x}{\sqrt{3}}\right)$
Im(P_8)	$-\left(\frac{11d \mp x}{\sqrt{3}}\right)$	$-\left(\frac{9d \pm x}{\sqrt{3}}\right)$	$-\left(\frac{13d \mp x}{\sqrt{3}}\right)$	$-\left(\frac{7d \pm x}{\sqrt{3}}\right)$	$-\left(\frac{7d \pm x}{\sqrt{3}}\right)$	–	–	–

At the transmitter, each complex information signal is first mapped to the center of the nearest hexagon (Step 4 of Figure 1), considering the decision boundaries given in [47, Table V]. More into details, [47, Table V] provides linear equations that define the boundaries of the hexagons in the constellation represented in [47, Figure 16]. When implementing our scheme, we noticed that the table given in [47, Table V] contained some typos in the definitions of the boundaries. This is why we restated the correct boundaries in Table I of this paper for further clarity and future use. Note that regular HQAM constellation has comparatively simpler detection. On the other hand, the irregular HQAM provides improved power efficiency and optimum performance, at the cost of increased detection complexity [25]. When considering IHQAM, if the modulated signals are passed through an AWGN channel with a noise variance σ^2, then according to [47], the Signal-to-Noise Ratio (SNR) in dB is equal to

$$\text{SNR}_{dB} = 10\log_{10}\left(\frac{E_s}{N_0}\right) = 10\log_{10}\left(\frac{35.25d^2}{2\sigma^2}\right),$$

where E_s is the average energy of each signal, and $N_0 = 2\sigma^2$ is the spectral density of a two-sided Gaussian noise. Moreover, for a fixed SNR_{dB}, we can apply relations (7), (24), and (25) in [47] to calculate the Bit Error Probability (BEP) P_b of the 64-IHQAM over an AWGN channel.

Finally, we did not consider channel coding prior to the modulation scheme, since we do not target data reconstruction. Machine Learning algorithms considered at the receiver have the ability to handle the noise introduced by the channel.

C. Scaling

We now describe how the proposed scheme scales and maps the complex-valued transform coefficients C_p onto the IHQAM constellation. For a given $s \in [\![1, S]\!]$, we use $\mathbf{C}_s = (C_{s,1}, C_{s,2}, \cdots, C_{s,P})$ to denote the vector of transform coefficients of size P, where $C_{s,p} = y_{s,p} + jz_{s,p}$ is a continuous complex value. For all $p \in [\![1, P]\!]$, the means of the random variables $y_{s,p}$ and $z_{s,p}$ are given by $\mu_{y,p} = \mathbb{E}[y_{s,p}]$ and $\mu_{z,p} = \mathbb{E}[z_{s,p}]$, respectively, and their variances are given by $\sigma^2_{y,p} = \text{Var}[y_{s,p}]$ and $\sigma^2_{z,p} = \text{Var}[z_{s,p}]$, respectively. Given that the number S of samples is sufficiently large, we can estimate $\mu_{y,p}$, $\mu_{z,p}$, $\sigma^2_{y,p}$, $\sigma^2_{z,p}$, from the empirical means and variances of the data samples $y_{s,p}$ and $z_{s,p}$. These empirical means and variances can be calculated both at the transmitter from the observed data, and at the receiver from the first data transmission phase. The normalized versions of the components $y_{s,p}$ and $z_{s,p}$ are denoted by $\bar{y}_{s,p} = \frac{y_{s,p} - \mu_{y,p}}{\sqrt{\sigma^2_{y,p}}}$ and $\bar{z}_{s,p} = \frac{z_{s,p} - \mu_{z,p}}{\sqrt{\sigma^2_{z,p}}}$, respectively. Then, given a Gaussian random variable U with mean 0 and variance 1, we introduce the quantity u_α such that $\mathcal{P}(|U| \leq u_\alpha) \geq \alpha$. In other words, the confidence interval

$$\mathcal{I}_{y,\alpha} = \left[-u_\alpha \sqrt{\sigma^2_{y,p}} + \mu_{y,p} \leq y_{n,p} \leq u_\alpha \sqrt{\sigma^2_{y,p}} + \mu_{y,p}\right]$$

contains $\alpha\%$ of the real (resp. imaginary) values of the transform coefficients $C_{s,p}$[1]. The value of α has an impact on the quality of the signal at the receiver.

Now considering the 64-IHQAM constellation, Step **3** of Figure 1 scales the confidence interval of real (resp. imaginary) part of each transform coefficients to real (resp. quadrature) axis of 64-IHQAM constellation. To do so, and given that the real (resp. quadrature) axis of 64-IHQAM is limited to $[-8d, 8d]$ (resp. $\left[-4\sqrt{3}d, 4\sqrt{3}d\right]$) we apply the following transformations, respectively on the normalized real part $\bar{y}_{s,p}$ and imaginary part $\bar{z}_{s,p}$ of each of the P coefficients $C_{s,p}$:

$$\begin{aligned} &T_{\text{Re},p} : \mathcal{I}_{y,\alpha} \to [-8d\ ,\ 8d] \\ &T_{\text{Re},p}(t) = \tfrac{16d}{3.29\sqrt{\sigma^2_{y,p}}} \left(t - u_\alpha \sqrt{\sigma^2_{y,p}} - \mu_{y,p}\right) + 8d, \end{aligned} \tag{7}$$

$$\begin{aligned} &T_{\text{Im},i} : \mathcal{I}_{z,\alpha} \to \left[-4\sqrt{3}d\ ,\ 4\sqrt{3}d\right] \\ &T_{\text{Im},i}(t) = \tfrac{8\sqrt{3}d}{3.29\sqrt{\sigma^2_{z,p}}} \left(t - u_\alpha \sqrt{\sigma^2_{z,p}} - \mu_{z,p}\right) + 4\sqrt{3}d \end{aligned} \tag{8}$$

[1]In our case study, we observed that the random variables $y_{s,p}$ and $z_{s,p}$ have statistical distribution approximately like normal distribution, hence the choice of the confidence interval of a Gaussian random variable. However, in case of unknown statistical distributions, the confidence interval could be defined thanks to the Chebyshev's inequality [48].

Transforms (7) and (8) map at least $\alpha\%$ of the most probable coefficients to the 2-dimensional region of the 64-IHQAM constellation. Note that the values that might be out of the $\alpha\%$ confidence interval are mapped into the borders of this interval.

After the scaling step, Step **4** of Figure 1 uses the decision boundaries of the 64-IHQAM modulation to further quantize any continuous 2-dimensional $[T_{\mathrm{Re},p}(t), T_{\mathrm{Im},p}(t)]$ value to one of the 64-IHQAM constellation points. Since we do not use channel coding, the scaled values are directly quantized into constellation points, with no “mapping" strategy.

IV. Receiver learning scheme

The objective of the receiver is to perform either clustering or classification onto the received noisy data output by the AWGN channel. For both learning tasks, the receiver starts with the same two steps: demodulation and descaling.

Step **6** of Figure 1 corresponds to demapping the noisy signal which is distorted by Gaussian noise. In this phase, we use the decision boundaries of the 64-IHQAM to demap the noisy signal to the center point of the nearest hexagon. Then, Step **7** corresponds to descaling the p-th component of the demodulated/demapped vector of size P. This is done by applying the inverse of the linear transform $T_{\mathrm{Re},p}$ in (7) (resp. $T_{\mathrm{Im},p}$ in (8)) on real (resp. imaginary) part of the i-th component of the demodulated/demapped vector. Since these components are the center points of the hexagons in the 64-IHQAM, their corresponding descaled (inverted) points only take values among 64 discrete possibilities.

The next step of Figure 1 is specific to the considered learning task, although both tasks rely on a Complex-Valued Neural Network (CVNN) which we now describe.

A. Complex-Valued Neural Networks

In this section, we only provide the salient points of CVNN, and we refer the reader to [49], [50] for a full description. Given that a CVNN has complex input values, the activation functions and their derivatives have to be well-defined so that one can apply *e.g.*, a gradient descent optimization method over complex data. Specifically, [50] relies on Wirtinger derivation in order to calculate the gradient of the loss function of a CVNN.

In what follows, we consider a CVNN with an input layer of size $k_0 + 1$ nodes, V hidden layers, k_v nodes per layer, and a single-valued output layer. We further consider the Cartesian hyperbolic tangent

$$F(Z) = \tanh(\mathrm{Re}(Z)) + j\ \tanh(\mathrm{Im}(Z)) \tag{9}$$

as activation function for hidden layers of CVNN, where Z is a complex value, and we consider the sigmoid-based activation function

$$G(Z) = \text{sigmoid}(\mathrm{Re}(Z) + \mathrm{Im}(Z)) \tag{10}$$

for the output layer[2]. CVNN with the aforementioned materials have been implemented in Python using *Tensorflow* and *Keras*, see [26], [51]. For weight initialization, Glorot uniform (also known as Xavier uniform) [52] is used, and all biases start at zero as those are *Tensorflow*'s current (v2.1) default initialization methods for dense layers.

B. Clustering with CVNN

In our scheme, clustering over the received data is performed in two steps. The first step consists of reconstructing rough versions of the original matrices $\mathbf{X}_s$ using a CVNN shown in Figure 2. This reconstruction step will allow to efficiently invert the transform operation, while removing a part of the channel noise. It will also allow to use the standard Euclidean distance in the cost function (1) of K-means. Indeed, applying K-means without data reconstruction would require to identify a proper distance adapted to the internal geometric structure of the demodulated data.

In the CVNN, we use the activation functions described in Section IV-A. As loss function, we consider the Mean-Squared Error (MSE), since the aim of the clustering algorithm K-means is to minimize the MSE between each sample and its closest centroid. The CVNN is trained from the set of $\lceil \beta S \rceil$ samples that were transmitted at the first phase. In this first phase, we take advantage of Data Augmentation (DA), a technique that allows to increase the amount of training data by adding modified copies of already existing data [53]. This results in a training set of N_t samples. For a given coefficient $X_{m,n}$, the input layer of the NN is fed with the vector

$$\left(\bar{C}_0 Y_0(\theta_m, \varphi_n), \bar{C}_1 Y_1(\theta_m, \varphi_n), \cdots, \bar{C}_P Y_P(\theta_m, \varphi_n)\right)^T,$$

[2]Here, we selected a sigmoid activation function because the dataset we consider in our simulations is composed of gray-scaled images. But depending on the target, the sigmoid could be replaced by other activation functions, see [50].

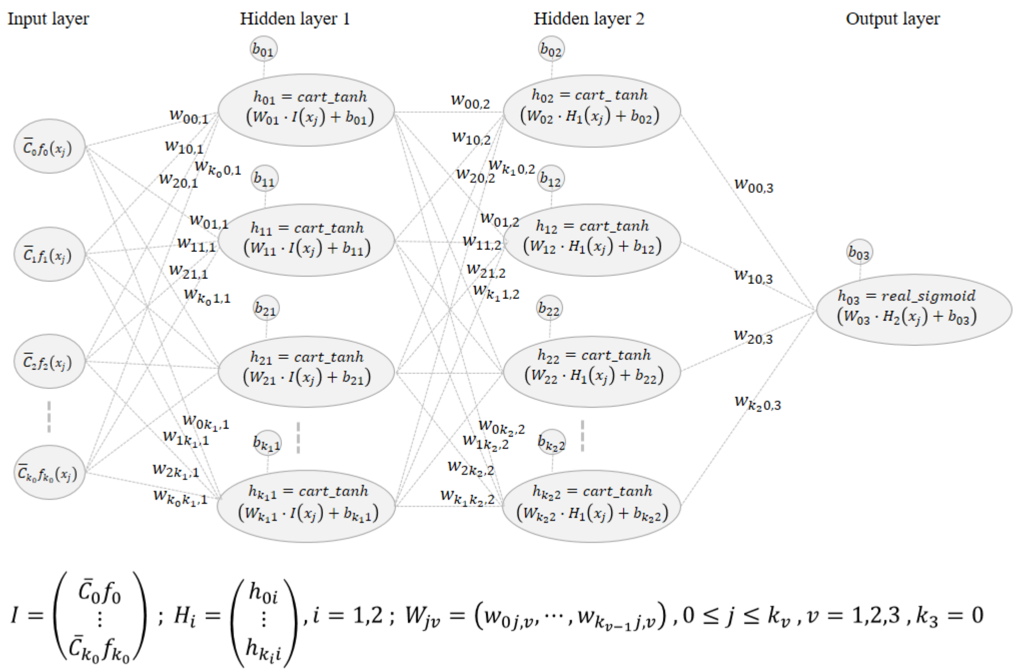


Fig. 2. Dense NN with input vector of size compressed vector, two hidden layers, and output vector of size one. In the figure, the function cart_tanh refers to the Cartesian hyperbolic tangent defined in (9), and the function real_sigmoid refers to the sigmoid-based function defined in (10).

see (6), where the coefficients $\bar{C}_p$ are the received transform coefficients after demodulation and descaling. The goal of the NN is to minimize the MSE between the NN output value $\hat{X}_{m,n}$ and the original value $X_{m,n}$, for all $(m, n) \in [\![0, M-1]\!] \times [\![0, N-1]\!]$. The NN is applied MN times so as to obtain MN components $\hat{X}_{m,n}$, which will allow to reconstruct a degraded version $\hat{\mathbf{X}}$ of each matrix $\mathbf{X}$.

Then, at the second stage of step **8**, the K-means algorithm is applied onto the set of reconstructed matrices $\{\hat{\mathbf{X}}_s\}_{s\in[\![1,S']\!]}$, where $S' \le S$ is the number of received data. In a practical system, for more efficiency, the K-means algorithm should be applied onto the set of matrices received from both the first and the second data transmission phase, so that a part of the matrices $\hat{\mathbf{X}}_s$ are in fact given by the original matrices $\mathbf{X}_s$ coming from the first step. However, in our simulations, we will only apply K-means onto the set of data received at the second phase, which will allow to evaluate with more fairness the performance of the proposed scheme (otherwise, the K-means algorithm could be positively biased by the first step).

C. Classification with CVNN

When considering classification instead of clustering, our proposed scheme remains entirely the same, except for Step 8 in Figure 1. Now, this step is composed of only one stage, since the CVNN will completely handle the classification. More into details, in the training phase, we assume that the receiver has access not only to a fraction β of the dataset, but also to the corresponding true labels. For instance, these true labels may be determined manually by a human operator, after a clustering step. This human operator should provide such effort for a small part of the dataset, hoping that the NN will provided labels for the remaining $(1-\beta)\%$ of the dataset. Manual labeling should be done anyway in many applications in which no prior dataset is available. As a result, while for clustering the NN is launched pixel-wise, here on the opposite the CVNN is fed with the P descaled transform coefficients $\bar{C}_p$. In addition, at the last layer, the number of NN outputs now equals the number of predefined classes, and the loss function is now the cross-entropy. However, the activation functions remain the same as for clustering.

D. Rate-adaption mechanism

Until now, we assumed in our description of the transmission scheme that the parameter P was fixed. This parameter indicates the number of SHF transform coefficients $C_\ell^{(k)}$ (5) which are retained at the transmitter, and it affects both the amount of data transmitted over the channel and the learning performance. This is why when training the CVNN after the first transmission phase, we will try different values of P and select the one that provides a sufficient level of learning performance. The performance criterion considered at this step will be defined in the next section for both clustering and classification. For classification, this step requires retaining a small part of the transmitted data into a validation set which will be used only for performance evaluation and will not be considered during the training. Finally, the retained value of P is sent back to the transmitter via the feedback link, and it will be used during the second transmission phase. As a result, the feedback channel should transmit $\lceil \log_2(P) \rceil$ bits of data (where $P \leq M \times N$), which is very low compared to the amount of data transmitted through the direct link, see the next section.

V. Rate and learning performance evaluation

Usually, the performance of conventional source and channel coding schemes is evaluated from metrics related to data reconstruction, such as the error probability for lossless source coding, or the distortion for lossy source coding. Alternatively, this section first identifies metrics of interest to evaluate the clustering and classification performance of a transmission scheme dedicated to learning. It then evaluates the source-channel coding rate of the proposed scheme.

A. Clustering performance

We consider two metrics of interest in order to evaluate the clustering performance of the proposed scheme. The first metric comes from the Confusion Matrix (CM), a square matrix such that the positive integer coefficient at position (i, j) gives the number of elements of actual class j which were predicted as belonging to class i. The metric

$$\text{cm} = \frac{\text{tr}(\text{CM})}{\sum_{i,j} \text{CM}(i, j)},$$

where $0 < \text{cm} \leq 1$, and $\text{tr}(.)$ is the trace of the matrix, is then calculated from the CM. The higher cm means the better clustering.

The second metric is the Silhouette score, denoted ss, and defined in [54]. The value of the Silhouette score varies between -1 and 1. A high ss means that the clusters are dense and well-separated from other clusters.

The first metric cm is calculated from the ground truth, that is the knowledge of the true clusters, while the second one ss only evaluates clusters homogeneity. We use these two very common metrics in our simulations, although many other ones exist (homogeneity score, completeness score, inertia, etc.), see [55] for an overview.

B. Classification performance

For classification, given the nature of the problem, performance criteria always require the ground truth. In our simulations, we will consider a very common one which is the accuracy [56]. Accuracy is simply calculated as the proportion of truly predicted labels among all tested samples.

C. Rate evaluation

We now evaluate the coding rate of the proposed transmission scheme. For the conventional coding scheme used at the first phase, we use R_{sc} and R_{cc} to denote respectively the source coding rate and the channel coding rate needed to transmit an image without or with loss, depending on what is needed at the first transmission phase. Then, since the number of input bits is BMN (assuming B bits per each of the $M \times N$ pixels of a gray-scaled image), and since the number of transmitted bits is $6P$ (P SHF coefficients are retained, and each one is mapped onto $2^6 = 64$ constellation points), the overall coding rate R_{learn} of our scheme is

$$R_{\text{learn}} = \beta \frac{R_{\text{sc}}}{R_{\text{cc}}} + (1-\beta)\frac{6P}{BMN}, \tag{11}$$

where β is the proportion of the dataset transmitted at the first phase, and the ratio $\frac{R_{\text{sc}}}{R_{\text{cc}}}$ is the joint source-channel coding rate [57]. In addition, while $R_{sc} \in [0,1]$ and $R_{cc} \in [0,1]$, the ratio $\frac{R_{\text{sc}}}{R_{\text{cc}}}$ does not necessarily belong to $[0,1]$.

In our simulations, we will assume that R_{sc} is the source coding rate after JPEG compression with a very small distortion, and that R_{cc} is the rate of an error-correction code aiming to correct most errors introduced by the AWGN channel. In (11), we see that the second transmission phase of our scheme highly benefits from the fact that no channel coding is employed in this phase. We also see that at this phase, the quantity to be optimized is the value of P that is the number of retained SHF's coefficients. Finally, note that if the learning was already done from a prior available dataset, we could set $\beta = 0$.

D. Coding rates of baseline schemes

In conventional data transmission setups, well-known Information-Theory results [58] state that the source coding R_{sc} should be greater than the source entropy (lossless coding) or than a certain rate-distortion function (lossy coding). These information-theoretic results allow to compare the performance of a given practical coding scheme with respect to the optimal coding rate. Unfortunately, no such result exists for the clustering and classification tasks considered in this paper, although some simpler problems such as DHT were addressed in the literature [10]–[12]. Determining the Information-Theoretic achievable performance for classification or clustering is out of our scope of this paper. Alternatively, we now provide the coding rates of two baseline schemes, which will serve as points of comparison with our approach.

As a first baseline, we consider a scheme in which the dataset is fully transmitted with conventional source and channel coding techniques, and completely reconstructed at the receiver, before applying the learning algorithm. This conventional approach has coding rate R_{conv} given by

$$R_{\text{conv}} = \frac{R_{\text{sc}}}{R_{\text{cc}}}, \tag{12}$$

where the terms $R_{\text{sc}}/R_{\text{cc}}$ were introduced in Section V-C. This corresponds to setting $\beta = 1$ in our proposed transmission scheme.

As a second baseline, we use the result of [59] which states that in theory, $K\log(K)$ coefficients are sufficient to retrieve correct cluster or class assignments, where K is the number of clusters or classes. This result holds after training, that is when the centroids (for clustering) or classes (for classification) are already known. Therefore, in this case, we consider the same first transmission phase as in our scheme, and we assume that $K\log(K)$ coefficients per data are transmitted in the second phase. We further consider that these coefficients are protected with a channel code of rate R_{cc}. The coding rate R_{ideal} of this scheme is given by

$$R_{\text{ideal}} = \beta\frac{R_{\text{sc}}}{R_{\text{cc}}} + (1-\beta)\,\frac{K\log K}{MNR_{\text{cc}}} \tag{13}$$

where K is the number of clusters or classes. This scheme is termed as “ideal" because it relies on the theoretical result of [59].

At the end, we expect the following rate ordering: $R_{\text{ideal}} \leq R_{\text{learn}} \leq R_{\text{conv}}$ to hold when considering the same level of learning performance among the three schemes. In our simulations, we will compare the rate-versus-learning performance of our scheme to these two baselines, and we will check whether the previous rate ordering is satisfied.

VI. SIMULATION RESULTS

In this section, we evaluate the performance of our proposed transmission scheme for clustering and classification. Our evaluation consists of two aspects: 1) the quality of clustering or classification, 2) the coding rate, given that the parameters of the scheme (number of SHF coefficients, etc.) are chosen so as to avoid any significant negative impact on the clustering or classification performance. The metrics considered in this evaluation are described in Section V. For both learning problems, we consider the MNIST dataset [60] that contains $70,000$ grayscale

TABLE II

CVNN PARAMETERS FOR CLUSTERING AND CLASSIFICATION, RESPECTIVELY, OVER MNIST

	CVNN Parameters	
task	clustering	classification
$\sharp$ nodes in input layer	$k_0 + 1 = 36, 49, 64, \cdots$	$k_0 + 1 = 16\text{MN}, 36\text{MN}, 64\text{MN}, \cdots$
Activation Function 1	Cartesian tangent hyperbolic	Cartesian tangent hyperbolic
$\sharp$ nodes in hidden layer 1	$k_1 + 1$	$k_1 + 1$
Activation Function 2	Cartesian tangent hyperbolic	Cartesian tangent hyperbolic
$\sharp$ nodes in hidden layer 2	$k_2 + 1$	$k_2 + 1$
Activation Function 3	Real-Sigmoid	Real-Softmax
$\sharp$ nodes in output layer	1	number of classes (10 for MNIST)
batchsize	200	200
epochs	10	10
loss function	mean squared error	categorical cross entropy
Optimizer method	Gradient descent Adam (rate 0.001)	Gradient descent Adam (rate 0.001)

images of size 28×28 of handwritten digits. The parameters of the considered CVNN are provided in Table II, both for clustering and classification.

A. Clustering

1) Simulation parameters: In order to evaluate the clustering performance of the proposed transmission scheme, we consider that $\beta = 1\%$ of MNIST samples are transmitted at the first phase, which represents 700 samples. The set of samples transmitted in the first phase is then expanded into $N_t = 2000$ samples by applying Data Augmentation (DA) techniques [61]. The considered DA technique includes a maximum of 20 degrees of rotation and up to 3 pixels left/right/up/down shifts. Next, the CVNN used at the receiver contains $V = 2$ layers, with $k_1 + 1 = 120$ nodes in its first hidden layer, and $k_2 + 1 = 784$ nodes at the second layer, which is equal to the number of pixels in each MNIST image. The CVNN is trained with 200 batch, 10 epochs, and a learning rate of 0.001. For the modulation scheme, we consider the 64-IHQAM constellation with parameter $d = 1$. In our simulations, we will evaluate the clustering performance for values $P \in \{36, 49, 64\}$. For the AWGN channel, we set a noise variance $\sigma^2 = 0.56$, which corresponds to a SNR around 15dB, and to a bit error probability $P_b = 0.062$ [47], [62].

TABLE III

PERFORMANCE EVALUATION OF THE K-MEANS CLUSTERING OVER ORIGINAL, BINARY, AND COMPRESSED DATA, WHERE COMPRESSED DATA ARE TRANSMITTED OVER AWGN CHANNEL WITH SNR $\approx$ 15dB AND $P_b = 0.062$.

metrics / K-means over:	cm	ss	Counter	σ^2_{Cout}	R (bits/pixel)
original MNIST	0.529	0.068	$\{0:242,\ 5:240,\ 8:226,\ 1:225,\ 9:194,$ $7:190,\ 3:189,\ 4:166,\ 6:165,\ 2:163\}$	963	$R_{\text{conv}} = 0.33$
binary MNIST (quantization: 1 bit per pixel)	0.506	0.067	$\{1:275,\ 3:240,\ 2:233,\ 6:224,\ 4:180,$ $9:179,\ 5:175,\ 8:168,\ 0:165,\ 7:161\}$	1569	$R_{\text{conv}} = 0.083$
NN outputs, with $P = 64$ AWGN ($\sigma^2 = 0.56$) & 64-IHQAM ($d = 1$)	0.526	0.081	$\{7:243,\ 0:232,\ 2:225,\ 8:213,\ 4:200,$ $9:198,\ 1:189,\ 6:184,\ 3:174,\ 5:142\}$	898	$R_{\text{learn}} = 0.064$
NN outputs, with $P = 49$ AWGN ($\sigma^2 = 0.56$) & 64-IHQAM ($d = 1$)	0.506	0.078	$\{5:256,\ 8:226,\ 6:219,\ 9:212,\ 4:204,$ $7:197,\ 2:195,\ 1:180,\ 3:178,\ 0:133\}$	1082	$R_{\text{learn}} = 0.050$
NN outputs, with $P = 36$ AWGN ($\sigma^2 = 0.56$) & 64-IHQAM ($d = 1$)	0.460	0.095	$\{7:258,\ 6:243,\ 2:241,\ 0:209,\ 5:197,$ $1:188,\ 4:186,\ 3:185,\ 8:173,\ 9:120\}$	1630	$R_{\text{learn}} = 0.037$

After completing the first transmission phase, and in order to evaluate the performance of the proposed transmission scheme, we consider the transmission of 2000 new samples in MNIST. In our scheme, for the clustering, we use the K-means algorithm initialized with K-means++, with 20 random initializations, a maximum of 300 iterations, and a tolerance value of 10^{-4}. The K-means function is imported from the Scikit.Learn Python library and for evaluation purpose, it is applied only on the samples transmitted at the second phase. In our simulations, we assume that the number $K = 10$ of clusters is known by the algorithm.

2) Clustering performance: In our simulations, we evaluate the clustering performance from the metrics ss and cm described in Section V-A. For further refinement, we also provide an array “Counter" which lists the number of samples assigned to each class. Since images in the MNIST dataset are uniformly distributed across clusters, a successful clustering should output almost an equal number of data points in each cluster. In this sense, for MNIST, a lower sample variance σ^2_{Count} for Counter means a better clustering quality.

We now present our simulation results for two sets of simulations. The first set of simulations aims to position our scheme with respect to K-means clustering on the original MNIST data, and on binary MNIST (one-bit quantization of each pixel). In other words, the K-means algorithm is applied directly on 2000 samples of these two datasets (original and binary), without considering our transmission scheme and without adding any channel noise. Table III shows the metrics cm, ss, and σ^2_{Count} obtained for original and binary versions of MNIST images, and also by applying the proposed scheme (referred to as “NN output images" in the table) with $P = 64,\ 49$, and 36. We observe from Table III that the aforementioned metrics are almost similar for clustering over original and binary versions of MNIST images, except for the variance σ^2_{Count} which is meaningfully smaller for original MNIST images. Then, when applying our scheme with $P = 64$ and $P = 49$, the metrics are all reasonably close to the ones for original MNIST images. Clustering over NN output images with $P = 36$ however leads to a performance degradation as it results in values of cm smaller and of σ^2_{Count} variance meaningfully larger than the ones for original MNIST. As a result, for MNIST and with the parameters considered in our simulations, setting P to any value greater than or equal to 49 seems sufficient to apply the K-means algorithm with a sufficient level of performance.

Finally, we notice that the Silhouette coefficient ss gives inconsistent results on our different setups. The metric ss does not rely on the ground truth, and instead evaluates clusters homogeneity. But here, the space in which the data evolves varies with the value of P, which may mislead the computation of ss and makes it improper to compare the schemes performance. Despite this, we provide the ss values, since the Silhouette criterion is widely considered in the literature.

Then, as the output of the second set of simulations, Table IV compares the clustering performance after two transforms: DCT and SHF. In this Table, we first show the clustering performance for a first scenario referred to as “reconstructed MNIST images" in which the considered two transforms are applied to the original data, but the transform coefficients pass neither through the transmission scheme nor through the noisy channel (no quantization, modulation, channel noise, etc., is applied to the data) and are reconstructed with the corresponding inverse transform. As a second scenario, we also restate the clustering performance of the proposed transmission scheme, referred to as “NN output images", for $P = 36$ and $P = 49$. From the metrics cm and σ^2_{Count} of Table IV, we see that clustering over direct DCT reconstruction is meaningfully better than with SHF. On the opposite, we observe that clustering after our transmission scheme and with SHF is far better than direct DCT and SHF reconstructions. This shows the efficiency of the CVNN in handling all the non-linear effects of the transmission scheme (quantization, channel noise, etc.). Finally, in Table IV, we notice that the methods which consider smaller values of P always have higher ss values. This confirms that the structure of the data has a strong influence

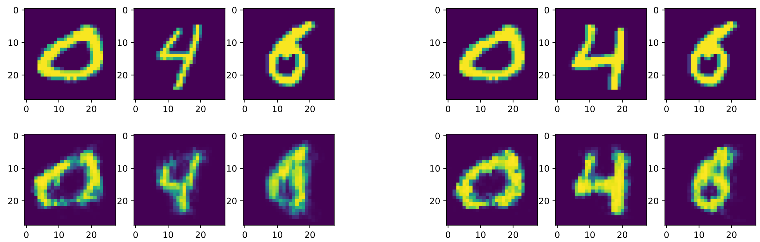
TABLE IV

COMPARISON OF THE K-MEANS CLUSTERING WITH DCT AND WITH SHF. K-MEANS IS APPLIED ON NN OUTPUT DATA, WHERE THEY ARE TRANSMITTED OVER AWGN CHANNEL WITH SNR $\approx$ 15DB AND $P_b = 0.062$.

metrics / K-means over:	cm	ss	Counter	σ^2_{Cout}
reconstructed MNIST images (using size= $(7,7)$ DCT sub-matrix)	0.465	0.112	$\{2:266,\ 5:266,\ 0:255,\ 4:237,\ 8:199,\ 7:176,\ 1:162,\ 6:157,\ 3:146,\ 9:136\}$	2665
reconstructed MNIST images (using size= $(6,6)$ DCT sub-matrix)	0.397	0.116	$\{1:320,\ 4:306,\ 6:282,\ 7:217,\ 2:217,\ 9:180,\ 3:152,\ 5:147,\ 0:103,\ 8:76\}$	7026
reconstructed MNIST images (using first 49 complex-valued SHF coefficients)	0.405	0.114	$\{8:283,\ 4:250,\ 2:246,\ 9:209,\ 6:208,\ 0:206,\ 7:204,\ 1:151,\ 5:125,\ 3:118\}$	2939
reconstructed MNIST images (using first 36 complex-valued SHF coefficients)	0.366	0.127	$\{6:271,\ 1:262,\ 0:231,\ 5:231,\ 4:230,\ 7:181,\ 2:161,\ 3:160,\ 9:155,\ 8:118\}$	2659
NN outputs, with $P = 49$ AWGN ($\sigma^2 = 0.56$) & 64-IHQAM ($d = 1$)	0.506	0.078	$\{5:256,\ 8:226,\ 6:219,\ 9:212,\ 4:204,\ 7:197,\ 2:195,\ 1:180,\ 3:178,\ 0:133\}$	1082
NN outputs, with $P = 36$ AWGN ($\sigma^2 = 0.56$) & 64-IHQAM ($d = 1$)	0.460	0.095	$\{7:258,\ 6:243,\ 2:241,\ 0:209,\ 5:197,\ 1:188,\ 4:186,\ 3:185,\ 8:173,\ 9:120\}$	1630

on the Silhouette values.

3) Rate evaluation: We now evaluate the rate of the proposed scheme, as well as the rates of the baseline schemes of Section V-C, for a given set of parameters. For MNIST, we have $M = N = 28$, and $K = 10$ clusters. As before, we consider a proportion $\beta = 1\%$ of data transmitted at the first phase. We consider a source coding rate $R_{sc} = 1/4$, since we observed that JPEG compression on MNIST with rate $R_{sc} = 1/4$ allows to reconstruct the original images almost without loss. And we consider a channel coding rate $R_{cc} = 3/4$, which is sufficient to correct errors with a bit error probability $P_b = 0.062$, as considered in our previous simulations. The coding rates R_{conv} and R_{learn} obtained with these parameters are shown in the last column of Table III, where the latter was evaluated for the three values $P = 36, 49, 64$. In the Table, we also indicated the coding rate R_{conv} for binary MNIST, and evaluate this coding rate by considering $R_{cc} = 3/4$ as before, $R_{sc} = 1/16$. This value of R_{sc} comes from the fact that one bit per pixel gives a compression ratio of $1/8$, and we observed that lossless Huffman coding allows to further divide this ratio by two. We observe that our scheme has a clear gain in coding rate compared to the considered two conventional coding schemes. Especially, the case $P = 49$ which was



(a) (b)

Fig. 3. Examples of MNIST images reconstructed by the CVNN before applying clustering. These images were encoded by the full transmitter scheme (including SHF and 64-IHQAM modulation), by considering a perfect channel: (a) Image reconstruction from 16 SHF coefficients, (b) Image reconstruction from 32 SHF coefficients.

identified as allowing for a sufficient clustering performance has a coding rate $R_{\text{learn}} = 0.05$ bit/symbol which is better than $R_{\text{conv}} = 0.083$ bit/symbol obtained for binary MNIST, while also allowing for a slightly improved clustering performance. In addition to the rates provided in Table III, we also get $R_{\text{ideal}} = 0.032$ bit/symbol for the ideal scheme described in Section V-C. We observe that the rate R_{learn} for $P = 49$ is not too far from the rate R_{ideal} of the ideal scheme, although there is still some space for a bit of improvement.

4) Data reconstruction: In the receiver designed for clustering, the CVNN first performs a reconstruction of the data, before applying the clustering algorithm. In Figure 3, we show some examples of MNIST images reconstructed by the CVNN, for 16 transmitted SHF coefficients (left figure) and for 32 transmitted SHF coefficients (right figure). These figures were obtained by considering the full emitter scheme with SHF and 64-IHQAM modulation, but without channel noise. The CVNN was trained over $\beta = 2.85\%$ of the dataset transmitted at the first phase. While data reconstruction is not the main purpose of our scheme, these figures also illustrate the generic aspect of our transmission scheme and the fact that it coud be employed in various applications.

B. Classification

1) Simulation parameters: We now evaluate the classification performance achieved by the proposed transmission scheme for various sets of parameters. The considered metric for classification performance evaluation is the accuracy. We consider the 64-IHQAM modulation with distance values $d \in \{1, 2\}$ in the constellation scheme, a number of SHF's coefficients $P \in \{16, 36\}$, and values $\beta \in \{2.85\%, 5.71\%\}$, for the proportion of MNIST samples transmitted at the first phase. We then evaluate the classification performance over a test set of 2000 MNIST images transmitted at the second phase of our scheme. We also consider several variance values for the channel noise, ranging from $\sigma^2 = 0$ to $\sigma^2 = 16$.

2) Classification performance: Table V shows the classification performance in terms of accuracy over the MNIST dataset using the proposed scheme. The same results are also represented graphically in Figure 4 for easier comparison between the different sets of parameters. As expected, we observe that the accuracy increases with P and d for fixed values of β and σ^2. Also, given P and d, the accuracy slightly increases with β. Finally, the parameters P and d seem to have a higher impact on the classification performance than β. Another deduction from Table V is that $P \geq 36$ is sufficient for our transmission scheme to achieve at least 90% accuracy when considering channel noise variance $0 \leq \sigma^2 \leq 9$. In addition, considering $P = 64$ instead of $P = 36$ does not improve much the accuracy. We also remark that we need a larger value β compared to when the transmission scheme was designed for clustering.

3) Rate evaluation: For conventional coding schemes, in order to obtain a fair rate comparison with our scheme, we considered classification of MNIST and binary MNIST from a standard Multilayer Perceptron (MLP) classifier, built from the same parameters as our CVNN (two layers, $k_1 + 1 = 120$, $k_2 + 2 = 784$). When trained on $\beta = 5.71\%$ of the original MNIST dataset, the MLP classifier returned an accuracy of 92% on a test set of 2000 MNIST samples. With the same value of β and training over binary MNIST, the MLP obtained an accuracy of 90%. In addition, for these two datasets (original and binary MNIST) with conventional coding schemes, the coding rates needed for classification are the same as the coding rates shown for clustering in Table III, that is $R_{\text{conv}} = 0.33$ bit/symbol for original MNIST, and $R_{\text{conv}} = 0.083$ bit/symbol for binary MNIST, given that we still consider $R_{cc} = 3/4$.

To obtain equivalent accuracy levels with our scheme, we can for instance consider $P = 36$ and $\beta = 5.71\%$, which gives accuracy larger than 90% for SNR values larger than 12.46dB. For these parameters, our scheme gives $R_{\text{learn}} = 0.051$ bit/symbol, which is still better than the coding rate of the conventional coding scheme for binary MNIST. In addition, considering a smaller value $\beta = 2.82\%$ with the same value $P = 36$ (at the price of a small accuracy degradation)

TABLE V

CLASSIFICATION PERFORMANCE OVER MNIST DATA (WITH $K = 10$ CLASSES) OF THE PROPOSED SCHEME, FOR AN AWGN CHANNEL. THE ACCURACY IS MEASURED OVER 2000 SAMPLES.

P	d	$\beta\%$	σ^2	SNR$_{dB}$	accuracy%	P	d	$\beta\%$	σ^2	SNR$_{dB}$	accuracy%	P	d	$\beta\%$	σ^2	SNR$_{dB}$	accuracy%
16	1	2.85	0	∞	85	36	1	2.85	0	∞	91	64	1	2.85	0	∞	93
16	1	2.85	1	12.46	80	36	1	2.85	1	12.46	90	64	1	2.85	1	12.46	91
16	1	2.85	4	6.44	73	36	1	2.85	4	6.44	83	64	1	2.85	4	6.44	88
16	1	2.85	6	4.68	70	36	1	2.85	6	4.68	76	64	1	2.85	6	4.68	86
16	1	2.85	9	2.91	67	36	1	2.85	9	2.91	75	64	1	2.85	9	2.91	84
16	1	2.85	12	1.67	62	36	1	2.85	12	1.67	69	64	1	2.85	12	1.67	74
16	1	2.85	16	0.42	55	36	1	2.85	16	0.42	64	64	1	2.85	16	0.42	72
16	1	5.71	0	∞	87	36	1	5.71	0	∞	92	64	1	5.71	0	∞	94
16	1	5.71	1	12.46	82	36	1	5.71	1	12.46	90	64	1	5.71	1	12.46	93
16	1	5.71	4	6.44	76	36	1	5.71	4	6.44	86	64	1	5.71	4	6.44	90
16	1	5.71	6	4.68	70	36	1	5.71	6	4.68	80	64	1	5.71	6	4.68	88
16	1	5.71	9	2.91	68	36	1	5.71	9	2.91	79	64	1	5.71	9	2.91	86
16	1	5.71	12	1.67	62	36	1	5.71	12	1.67	69	64	1	5.71	12	1.67	76
16	1	5.71	16	0.42	58	36	1	5.71	16	0.42	65	64	1	5.71	16	0.42	76
16	2	2.85	0	∞	85	36	2	2.85	0	∞	91	64	2	2.85	0	∞	92
16	2	2.85	1	18.48	83	36	2	2.85	1	18.48	91	64	2	2.85	1	18.48	92
16	2	2.85	4	12.46	80	36	2	2.85	4	12.46	88	64	2	2.85	4	12.46	91.5
16	2	2.85	6	10.7	80	36	2	2.85	6	10.7	84	64	2	2.85	6	10.7	85
16	2	2.85	9	8.93	76	36	2	2.85	9	8.93	83	64	2	2.85	9	8.93	84
16	2	2.85	12	7.7	75	36	2	2.85	12	7.7	82	64	2	2.85	12	7.7	83
16	2	2.85	16	6.44	73	36	2	2.85	16	6.44	82	64	2	2.85	16	6.44	82
16	2	5.71	0	∞	87	36	2	5.71	0	∞	93	64	2	5.71	0	∞	94
16	2	5.71	1	18.48	85	36	2	5.71	1	18.48	92	64	2	5.71	1	18.48	94
16	2	5.71	4	12.46	80	36	2	5.71	4	12.46	90	64	2	5.71	4	12.46	93
16	2	5.71	6	10.7	80	36	2	5.71	6	10.7	90	64	2	5.71	6	10.7	92
16	2	5.71	9	8.93	79	36	2	5.71	9	8.93	89	64	2	5.71	9	8.93	92
16	2	5.71	12	7.7	77	36	2	5.71	12	7.7	83	64	2	5.71	12	7.7	84
16	2	5.71	16	6.44	74	36	2	5.71	16	6.44	82	64	2	5.71	16	6.44	84

gives a coding rate $R_{\text{learn}} = 0.042$ bit/symbol, which is even closer to the rate of the ideal scheme $R_{\text{ideal}} = 0.038$ bit/symbol. This allows to conclude that the proposed transmission scheme permits to obtain a better coding rate than conventional coding.

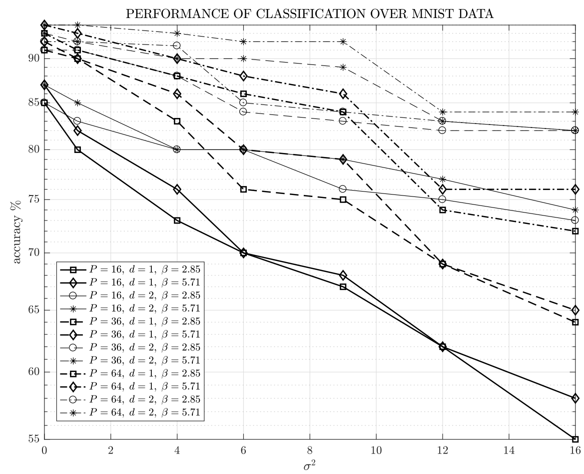


Fig. 4. Accuracy of our proposed classification scheme applied on compressed MNIST data transmitted over an AWGN channel with noise variance σ^2

VII. Conclusion

In this paper, we introduced a practical transmission scheme for efficient learning over received data at the output of an AWGN channel. The proposed scheme consists of a transmitter built from SHF transform and IHQAM modulation, and of a receiver that makes use of a CVNN to perform the considered learning task. We also provided the source/channel coding rate of this scheme, and evaluated its learning performance from numerical simulations. Numerical results showed a clear gain in terms of coding rate compared to conventional coding approaches, at the same learning performance level. These promising results were obtained given that no prior training dataset is needed by our scheme, and that only a small feedback was allowed between the receiver and the transmitter. Although generic, the proposed scheme was specified and evaluated for two standard learning tasks that are clustering with K-means and classification. Future works will be dedicated to specifying the proposed scheme to other learning tasks such as regression or clustering from other techniques. We will also investigate other channel models like the fading channel.

Acknowledgement

This work was supported by a French government support granted to the Cominlabs excellence laboratory under reference ANR-10-LABX-07-01, and by grant ANR-17-CE40-0020 (project EF-

FECtive) of the French National Research Agency ANR. The authors would like to thank Amin Zribi for his careful reading of the manuscript.

REFERENCES

[1] X. Lai, Q. Liu, X. Wei, W. Wang, G. Zhou, and G. Han, “A survey of body sensor networks,” *Sensors*, vol. 13, no. 5, pp. 5406–5447, 2013.

[2] S. C. Mukhopadhyay, “Wearable sensors for human activity monitoring: A review,” *IEEE sensors journal*, vol. 15, no. 3, pp. 1321–1330, 2014.

[3] I. F. Akyildiz, D. Pompili, and T. Melodia, “Underwater acoustic sensor networks: research challenges,” *Ad hoc networks*, vol. 3, no. 3, pp. 257–279, 2005.

[4] A. Miglani and N. Kumar, “Deep learning models for traffic flow prediction in autonomous vehicles: A review, solutions, and challenges,” *Vehicular Communications*, vol. 20, p. 100184, 2019.

[5] E. C. Strinati and S. Barbarossa, “6G networks: Beyond Shannon towards semantic and goal-oriented communications,” *Computer Networks*, vol. 190, p. 107930, 2021.

[6] E. Tuncel and D. Gündüz, “Identification and lossy reconstruction in noisy databases,” *IEEE transactions on information theory*, vol. 60, no. 2, pp. 822–831, 2013.

[7] Y. Blau and T. Michaeli, “Rethinking lossy compression: The rate-distortion-perception tradeoff,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 675–685.

[8] T. Dumas, A. Roumy, and C. Guillemot, “Autoencoder based image compression: can the learning be quantization independent?” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 1188–1192.

[9] F. A. Aoudia and J. Hoydis, “End-to-end learning of communications systems without a channel model,” in *52nd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2018, pp. 298–303.

[10] S. Sreekumar and D. Gündüz, “Distributed hypothesis testing over discrete memoryless channels,” *IEEE Transactions on Information Theory*, vol. 66, no. 4, pp. 2044–2066, 2019.

[11] G. Katz, P. Piantanida, R. Couillet, and M. Debbah, “On the necessity of binning for the distributed hypothesis testing problem,” in *IEEE International Symposium on Information Theory (ISIT)*, 2015, pp. 2797–2801.

[12] S. Salehkalaibar, M. Wigger, and L. Wang, “Hypothesis testing over the two-hop relay network,” *IEEE Transactions on Information Theory*, vol. 65, no. 7, pp. 4411–4433, 2019.

[13] S. Salehkalaibar and M. Wigger, “Distributed hypothesis testing based on unequal-error protection codes,” *IEEE Transactions on Information Theory*, vol. 66, no. 7, pp. 4150–4182, 2020.

[14] M. El Gamal and L. Lai, “Are slepian-wolf rates necessary for distributed parameter estimation?” in *53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2015, pp. 1249–1255.

[15] M. Raginsky, “Learning from compressed observations,” in *IEEE Information Theory Workshop*, 2007, pp. 420–425.

[16] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk, “Signal processing with compressive measurements,” *IEEE Journal of Selected topics in Signal processing*, vol. 4, no. 2, pp. 445–460, 2010.

[17] D. Pastor and F.-X. Socheleau, “Random distortion testing with linear measurements,” *Signal Processing*, vol. 145, pp. 116–126, 2018.

[18] E. Dupraz, D. Pastor, and F.-X. Socheleau, “A statistical signal processing approach to clustering over compressed data,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2791–2795.

[19] F. Pourkamali-Anaraki and S. Becker, “Preconditioned data sparsification for big data with applications to pca and k-means,” *IEEE Transactions on Information Theory*, vol. 63, no. 5, pp. 2954–2974, 2017.

[20] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[21] E. Dupraz, A. Roumy, and M. Kieffer, “Source coding with side information at the decoder and uncertain knowledge of the correlation,” *IEEE Transactions on Communications*, vol. 62, no. 1, pp. 269–279, 2013.

[22] V. Toto-Zarasoa, A. Roumy, and C. Guillemot, “Maximum likelihood bsc parameter estimation for the slepian-wolf problem,” *IEEE Communications Letters*, vol. 15, no. 2, pp. 232–234, 2010.

[23] E. Dupraz, “K-means Algorithm over Compressed Binary Data,” in *Data compression conference (DCC)*, 2018.

[24] A. Qamar, I. Din, and M. A. Khan, “Analysis of spherical harmonics and singular value decomposition as compression tools in image processing.” 2012.

[25] M. Abdelaziz and T. A. Gulliver, “Triangular constellations for adaptive modulation,” *IEEE Transactions on Communications*, vol. 66, no. 2, pp. 756–766, 2018.

[26] J. A. Barrachina, “Complex-valued neural networks (cvnn),” Jan. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.4452131

[27] A. K. Jain, “Data clustering: 50 years beyond k-means,” *Pattern recognition letters*, vol. 31, no. 8, pp. 651–666, 2010.

[28] G. Hamerly and C. Elkan, “Learning the k in k-means,” *Advances in Neural Information Processing Systems*, vol. 17, 03 2004.

[29] S. Hess and W. Duivesteijn, “k is the magic number—inferring the number of clusters through nonparametric concentration inequalities,” in *Machine Learning and Knowledge Discovery in Databases*, U. Brefeld, E. Fromont, A. Hotho, A. Knobbe, M. Maathuis, and C. Robardet, Eds. Cham: Springer International Publishing, 2020, pp. 257–273.

[30] G. P. Zhang, “Neural networks for classification: a survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 30, no. 4, pp. 451–462, 2000.

[31] D. M. Kline and V. L. Berardi, “Revisiting squared-error and cross-entropy functions for training neural network classifiers,” *Neural Computing & Applications*, vol. 14, no. 4, pp. 310–318, 2005.

[32] D. Edmundson and G. Schaefer, “An overview and evaluation of jpeg compressed domain retrieval techniques,” *Proceedings ELMAR-2012*, pp. 75–78, 2012.

[33] M. Ehrlich and L. S. Davis, “Deep residual learning in the jpeg transform domain,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3484–3493.

[34] S. Dodge and L. Karam, “Understanding how image quality affects deep neural networks,” in *Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, 2016, pp. 1–6.

[35] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, “Deep convolutional autoencoder-based lossy image compression,” in *Picture Coding Symposium (PCS)*. IEEE, 2018, pp. 253–257.

[36] T. J. O’Shea, K. Karra, and T. C. Clancy, “Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention,” in *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2016, pp. 223–228.

[37] E. Bourtsoulatze, D. B. Kurka, and D. Gündüz, “Deep joint source-channel coding for wireless image transmission,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, no. 3, pp. 567–579, 2019.

[38] K. Choi, K. Tatwawadi, A. Grover, T. Weissman, and S. Ermon, “Neural joint source-channel coding,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 1182–1192.

[39] Y. M. Saidutta, A. Abdi, and F. Fekri, "Joint source-channel coding over additive noise analog channels using mixture of variational autoencoders," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 7, pp. 2000–2013, 2021.

[40] M. Jankowski, D. Gündüz, and K. Mikolajczyk, "Wireless image retrieval at the edge," *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 89–100, 2020.

[41] Y. M. Saidutta, A. Abdi, and F. Fekri, "Analog joint source-channel coding for distributed functional compression using deep neural networks," in *IEEE International Symposium on Information Theory (ISIT)*, 2021, pp. 2429–2434.

[42] C.-H. Lee, J.-W. Lin, P.-H. Chen, and Y.-C. Chang, "Deep learning-constructed joint transmission-recognition for internet of things," *IEEE Access*, vol. 7, pp. 76 547–76 561, 2019.

[43] K.-L. Lim, X. Jiang, and C. Yi, "Deep clustering with variational autoencoder," *IEEE Signal Processing Letters*, vol. 27, pp. 231–235, 2020.

[44] S. Yun, J.-M. Kang, S. Choi, and I.-M. Kim, "Cooperative inference of DNNs over noisy wireless channels," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 8, pp. 8298–8303, 2021.

[45] M. Gastpar, B. Rimoldi, and M. Vetterli, "To code, or not to code: Lossy source-channel communication revisited," *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1147–1158, 2003.

[46] "Table of spherical harmonics." [Online]. Available: https://en.wikipedia.org/wiki/Table_of_spherical_harmonics#Real_spherical_harmonics

[47] P. K. Singya, P. Shaik, N. Kumar, V. Bhatia, and M.-S. Alouini, "A survey on higher-order QAM constellations: Technical challenges, recent advances, and future trends," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 617–655, 2021.

[48] "Chebyshev's inequality." [Online]. Available: https://en.wikipedia.org/wiki/Chebyshev%27s_inequality

[49] A. Hirose, *Complex-Valued Neural Networks*. Springer Berlin Heidelberg, 2012.

[50] M. F. Amin *et al.*, *Complex-valued neural networks: learning algorithms and applications*. LAP LAMBERT Academic Publishing, 2018.

[51] "Complex-valued neural network." [Online]. Available: https://complex-valued-neural-networks.readthedocs.io/en/latest/index.html

[52] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," *JMLR Workshop and Conference Proceedings*, pp. 249–256, 2010.

[53] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[54] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[55] "A demo of k-means clustering on the handwritten digits data." [Online]. Available: https://scikit-learn.org/stable/auto_examples/cluster/plot_kmeans_digits.html

[56] P. Branco, L. Torgo, and R. Ribeiro, "A survey of predictive modelling under imbalanced distributions," *arXiv preprint arXiv:1505.01658*, 2015.

[57] M. B. Abdessalem, A. Zribi, T. Matsumoto, E. Dupraz, and A. Bouallegue, "LDPC-based joint source channel coding and decoding strategies for single relay cooperative communications," *Physical Communication*, vol. 38, p. 100947, 2020.

[58] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[59] C. Boutsidis, A. Zouzias, M. W. Mahoney, and P. Drineas, "Randomized dimensionality reduction for k-means clustering," *IEEE Transactions on Information Theory*, vol. 61, no. 2, pp. 1045–1062, 2014.

[60] Y. LeCun, C. Cortes, and C. J. Burges, “The MNIST database of handwritten digits,” 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[61] C. Sammut and G. I. Webb, *Encyclopedia of machine learning and data mining*. Springer, 2017.

[62] F. Cogen and E. Aydin, “Performance analysis of hexagonal qam constellations on quadrature spatial modulation with perfect and imperfect channel estimation,” *Physical Communication*, p. 101379, 2021.