IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

End-to-End Learning-based Framework for Amplify-and-Forward Relay Networks

ANKIT GUPTA, (Student Member, IEEE) and MATHINI SELLATHURAI, (Senior Member, IEEE)

Ankit Gupta and Dr. Mathini Sellathurai are with the department of engineering and physical sciences (EPS), Heriot-Watt University, Edinburgh, UK (e-mail: ag104@hw.ac.uk, m.sellathurai@hw.ac.uk).

Corresponding author: Ankit Gupta (e-mail: ag104@hw.ac.uk).

This work was supported in part by the U.K. Engineering and Physical Sciences Research Council under Grant EP/P009670/1 and in part by the U.K.-India Education and Research Initiative Thematic Partnerships under Grant UGC -UKIERI 2016-17-058.

ABSTRACT We study end-to-end learning-based frameworks for amplify-and-forward (AF) relay networks, with and without the channel state information (CSI) knowledge. The designed framework resembles an autoencoder (AE) where all the components of the neural network (NN)-based source and destination nodes are optimized together in an end-to-end manner, and the signal transmission takes place with an AF relay node. Unlike the literature that employs an NN-based relay node with full CSI knowledge, we consider a conventional relay node that only amplifies the received signal using CSI gains. Without the CSI knowledge, we employ power normalization-based amplification that normalizes the transmission power of each block of symbols. We propose and compare symbol-wise and bit-wise AE frameworks by minimizing categorical and binary cross-entropy loss that maximizes the symbol-wise and bit-wise mutual information (MI), respectively. We determine the estimated MI and examine the convergence of both AE frameworks with signal-to-noise ratio (SNR). For both these AE frameworks, we design coded modulation and differential coded modulation, depending upon the availability of CSI at the destination node, that obtains symbols in 2n-dimensions, where n is the block length. To explain the properties of the 2n-dimensional designs, we utilize various metrics like minimum Euclidean distance, normalized second-order and fourth-order moments, and constellation figures of merit. We show that both these AE frameworks obtain similar spherical coded-modulation designs in 2n-dimensions, and bit-wise AE that inherently obtains the optimal bit-labeling outperforms symbol-wise AE (with faster convergence under low SNR) and the conventional AF relay network with a considerable SNR margin.

INDEX TERMS Amplify-and-forward, autoencoder, feed-forward neural networks, learning, and relay networks.

I. INTRODUCTION

End-to-end learning has appeared as a promising solution for jointly optimizing all the components of the point-to-point (P2P) communication network consisting of a neural network (NN)-based encoder and decoder at the transmitter and receiver by employing an autoencoder (AE) framework [1]. More complicated than the P2P networks is the relay networks, which include a relaying node assisting the transmission of the signals from the source to the destination node. We can broadly classify the relaying schemes as amplify-and-forward (AF) [2], [3] and decode-and-forward (DF) [4]. However, the AF scheme is employed practically because it provides us with low implementation complexity, and the signal is received, amplified, and re-transmitted all in the analog domain. Therefore in this work, we consider an end-to-end learning-based framework for AF relay network, wherein all the components at the source and destination nodes are optimized together in an end-to-end manner, similar to the AE-based P2P networks proposed in [1], but the signal transmission between the source and destination nodes takes place with the aid of an AF relay node.

A. LITERATURE REVIEW

In general, for any k bits transmitted using the n complex-baseband symbols, the AE frameworks can be broadly classified as symbol-wise AE (SWAE) and bit-wise AE (BWAE) frameworks. In a symbol-wise AE framework, one-hot vector representation of the 2^k possible symbols forms the input and output of the AE, while the AE is optimized by minimizing the categorical cross-entropy (CE) loss [1]. The symbol-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3085901, IEEE Access

wise AE for P2P communication networks has been widely investigated in [1], [5]– [11] and for the multi-user networks in [12]. Further, the authors in [13]– [16] studied symbol-wise AE framework for the AF and DF relay networks. However, as the symbol-wise AE's input and output are in the form of symbols, bit-labeling has to be done separately either by exhaustively searching through the $2^k!$ combinations or by heuristic search method, leading to the sub-optimal bit-labelings and bit-error-rate (BER) performance [17].

On the other hand, the bit-wise AE framework, takes k bits as input and output of the AE, while the AE is optimized by minimizing the binary CE loss. The bit-wise AE framework has been investigated for P2P networks in [17], and for AF and DF relay networks in [18] and [19], respectively. Although the bit-wise AEs seem like a trivial modification of the symbol-wise AEs, but by providing the bit-wise AE's input and output in the form of bits, we obtain automatic bit-labeling. Further, the authors in [17] analyzed the 2-dimensional (I and Q) modulation design comparing the bit-wise and symbol-wise AE frameworks for the P2P networks. The authors in [17] show that while the symbol-wise AE-based trained constellation improves the symbol-error-rate (SER), it degrades the BER in comparison to the bit-wise AE, in a P2P network. This is because symbol-wise AE aims to maximize the symbol-wise mutual information (MI), whereas bit-wise AE aims to maximize the bit-wise MI.

The symbol-wise AE-based coded-modulation design has been investigated for P2P and relay networks widely by replacing the channel-coding and modulation blocks with a NN at the encoder, and channel-decoding and demodulation blocks with a NN at the deoder [1], [5]– [16]. Recently, the bit-wise AE-based coded-modulation design has been investigated in [17] for the P2P networks. Until now no tool was known that can obtain block codes with automatic bit-labeling as a result of mathematical modeling of the communication system. In fact, the Shannon's coding theorem only states the existence of a good code without specificity, and only for infinite block lengths [20]. Thus, bit-wise AE has appeared as a novel research direction to obtain block codes for short block lengths in a P2P network. Although the authors in [18] focussed on bit-wise AE-based 2-dimensional modulation design with the achievable-sum-rate analysis for the AF relay networks, the bit-wise AE-based coded-modulation design has never been studied in the literature for AF relay networks, but also its BER analysis.

Furthermore, as the channel estimation still remains a challenging task, especially in a relay network, where two-hop channel state information (CSI) knowledge needs to be known accurately at the destination node to decode the signals correctly. The differential coded-modulation design is another important research topic, which has been investigated for the symbol-wise AE frameworks in [1], [5]– [16]. However, the bit-wise AE-based differential coded-modulation design has never been studied for the AF relay networks.

The analysis of the coded-modulation design comes up with new challenges, as these codewords are designed in the $2n$-dimensional space (n denotes the block length). In some of the previous works where the insights to the symbol-wise AE-based coded-modulation designs in $2n$-dimensional space is shown [1], [13], [16] it remains confined to the t-stochastic neighbour embedding (t-SNE) representation [21]. Although the t-SNE representation helps us to collapse the $2n$-dimensional space to 2 dimensions, but it does not reveal much information of designed coded modulation in $2n$-dimensional space, by only indicating the clusters of symbols in 2-dimensions. Moreover, the authors in [11] analyzed the packing density of symbol-wise AE frameworks in P2P networks. However, none of the works [1], [5]– [17], have analyzed and contrasted the bit-wise AE-based designed constellations over that of the symbol-wise AE by analyzing the coded-modulation designs either for the P2P or relay networks. Furthermore, comparison of bit-wise and symbol-wise AE from an information-theoretic perspective, such as by comparing their minimized CE loss functions and estimated MI, has never been investigated in the literature to determine the impact of AE framework on its convergence with the signal-to-noise ratio for the AF relay networks.

Thus, there exists a need to contrast the AE-based (differential) coded-modulation designs for the symbol-wise and bit-wise AE frameworks to understand the $2n$-dimensional codewords designed in both the frameworks and their potential BER performance gains. Further, to the best of the authors' knowledge, no such comparative study between bit-wise and symbol-wise AE frameworks exists for relay networks or coded-modulation designs even for P2P networks.

In particular, end-to-end learning-based relay networks using AE frameworks have been studied for AF relaying networks in [13], [18] and for DF relaying networks in [14]– [16], [19]. The authors in [18] studied a two-way AF relay network using a bit-wise AE performing only modulation design in 2-dimensions by employing NN-based multiple fully-connected (dense) layers at the AF relay node, while the authors in [13] studied a one-way AF relay network using a symbol-wise AE with NN-based multiple dense layers at the AF relay node. However, conventionally AF scheme is designed to have lower complexity at the relay node, with just the amplification operation to take place. Whereas, the NN-based processing by the use of dense layers at the relay node in [13], [18] contradicts the low implementation complexity intended for the AF scheme. Furthermore, the authors in [13], [18] considered full CSI knowledge at the AF relay node in their AE frameworks. However, conventionally the AF relay node only has the information about the channel gain knowledge, thereby providing an inherent advantage of utilizing the phase information at the AE-based framework over the conventional AF relay networks. Therefore, to the best of the authors' knowledge, none of the previous works have studied bit-wise AE-based coded-modulation design, and/or, have considered a minimal complexity AF relay node with the fair CSI requirements as the conventional networks.

Moreover, the removal of the CSI knowledge is studied in [13] for the AF relay network and in [14]– [16], [19] for

DF relay networks. The authors in [13] employed a single NN at relay node to decode and re-encode the signal together, whereas the authors in [14]– [16], [19] employed a greater decoding power by explicitly decoding and encoding signals at the relay with the help of separate NN-based decoders and encoders. However, to the best of the authors' knowledge, none of the previous works have removed the necessity of the CSI knowledge without utilizing NN-based processing at the relay node either for a bit-wise or symbol-wise AE framework, and/or, have studied bit-wise AE-based differential coded-modulation designs for an AF relaying networks.

Furthermore, for an end-to-end learning framework without the CSI knowledge, the radio transformer network (RTN) was first introduced in [1], and since then has been widely employed for decoding the signal in the absence of CSI knowledge at the decoder, such as in [14], [19]. However, to the best of the authors' knowledge, none of the previous works have proposed an RTN for AF relay networks for either the symbol-wise or bit-wise AE frameworks.

B. CONTRIBUTIONS

Now, we summarize the major contributions of this work as follows:

- We propose a novel end-to-end learning-based AF relay network using the AE frameworks. Specifically, we propose to employ NNs consisting of dense layers at the source and destination nodes that constitute the AE framework and employ a conventional AF relay node to minimize the implementation cost and maintain a fair CSI requirement between the proposed and conventional relay networks, compared to [13], [18].
- We design both the bit-wise and symbol-wise AE framework for the AF relay assisted network, and show that these frameworks are optimized by maximizing the bit-wise MI and symbol-wise MI of an AF relaying network, respectively, while minimizing the relative entropy between the posterior distributions at the encoder and decoder. Later, we formulate the AF relay assisted AE-based framework as a multilabel (multiclass) classification task for the bit-wise (symbol-wise) AEs.
- Both the proposed AE frameworks perform joint channel coding and modulation (coded-modulation) design. In particular, at the source node, the NN encoder of the bit-wise AE takes k bits as input and provides n complex-baseband symbols as output, while the NN encoder of the symbol-wise AE takes one-hot vector representation of the 2^k possible symbols as input to provide n complex-baseband symbols output. While at the destination node, the decoder utilizes a separate NN to decode k bits (or one-hot vector representation of the 2^k possible symbols) from the n symbols. Thus, the coded-modulation (demodulation) design takes place in $2n$-dimensional space. We train the encoder-decoder in an end-to-end manner by minimizing the binary CE and categorical CE loss functions in the bit-wise and symbol-wise AE, respectively. For greater insights, we compare the CE loss functions from an information-theoretic perspective and the estimated MI to analyze the convergence of the proposed AE frameworks.
- We remove the need for CSI knowledge and noise variances of the links for the proposed AE-based frameworks, even without the NN-based processing at the relay node, unlike [14]– [16], [19]. We show that training the encoder and decoder in an end-to-end manner leads to block-by-block differential coded-modulation (demodulation) design in $2n$-dimensional space, thereby enabling the AE framework to decode the signals without the CSI knowledge. Furthermore, we propose and investigate the impact of NN-based RTN on the AF relaying network.
- Further, without CSI knowledge, the amplification factor for the conventional AF relay node becomes a fixed value depending on the second-order statistics of the channel between the source to relay node and noise variances [22]– [28]. It is evident that fixed amplification factor is a sub-optimal approach for the AF scheme, thus, we also show that by utilizing a power normalization layer that normalizes the transmission power of n symbols to n at the AF relay node, we can improve the process of deciding the amplification factor, while keeping the signal transmission-reception in the analog domain and removing the requirement of second-order channel statistics and noise variances at the relay and CSI knowledge at the destination node.
- We focus on interpretability and analysis of the AE-based designed coded modulation in the $2n$-dimensional space, by utilizing various metrics, such as the minimum Euclidean distance, normalized second- and fourth-order moments, and constellation figure of merit. We compare the AE-based coded modulation designs for various scenarios, such as with and without CSI knowledge, between symbol-wise and bit-wise AEs, etc.

C. PAPER ORGANIZATION

The rest of the paper is organized as follows. We detail the system model for the considered AF relay network in Section II. In Section III we propose the bit-wise and symbol-wise AE and analyze the CE loss. In Section IV we detail the NN architecture, and process of training and testing. In Section V we show the performance evaluation for the proposed AE frameworks, and conclude this work with future directions in Section VI.

D. LIST OF ABBREVIATIONS AND NOTATIONS

To improve the readability of the paper, we have summarized the abbreviations in Table 1 and notations in Table 2, 3.

II. SYSTEM MODEL

In this section, we present the conventional AF relay networks, where the source node (S) wants to exchange its intended signal with the destination node (D) by employing an AF relay node (R) in two-phases, as shown in Fig. 1. Each

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3085901, IEEE Access

TABLE 1: Summary of abbreviations in the paper.

Abbreviation	Description
AE	Autoencoder
AF	Amplify-and-forward
AWGN	Additive white Gaussian noise
BER	Bit-error-rate
BWAE	Bit-wise Autoencoder
CE	Cross-entropy
CS	Channel statistics
CSI	Channel state information
D	Destination node
LLR	Log-likelihood ratio
MI	Mutual information
MLD	Maximum likelihood detector
NN	Neural network
P2P	Point-to-point
PN	Power normalization
R	Relay node
RBF	Rayleigh block fading
RTN	Radio transformer network
S	Source node
SNR	Signal-to-noise ratio
SWAE	Symbol-wise Autoencoder
TP	Transmit power
t-SNE	t-Stochastic neighbour embedding

TABLE 2: Summary of notations in the paper.

Notations	Description
2^k	Total possible symbols or codewords
$\mathbf{1}_s \in \{1, ..., 2^k\}$	One-hot input of SWAE
$\hat{\mathbf{1}}_s \in \mathbb{R}^{2^k}$	One-hot output of SWAE
$\boldsymbol{\theta}_s$	NN encoder parameters (weights and bias terms)
$\boldsymbol{\theta}_d$	NN decoder parameters (weights and bias terms)
δ_l	Number of neurons in l^{th} layer
$\sigma_l(\cdot)$	Activation function in l^{th} layer
$\boldsymbol{\pi}$	Parameters of AE
τ	Learning-rate
∇	Gradient operator
χ	Normalized fourth order-moment
α	Amplification factor
σ_r^2	Noise variance at relay node
σ_d^2	Noise variance at destination node
σ_{sr}^2	Second-order channel statistics
$\sigma(\mathbf{x})_i$	Softmax activation
$\sigma(x)$	Sigmoid activation

TABLE 3: Summary of notations in the paper.

Notations	Description
B	Batch-size
$\mathcal{C}$	All possible alphabets
CFM	Constellation figures of merit
D_{KL}	KL-divergence
$d_{\min}$	Minimum Euclidean distance
E_n	Normalized second-order moment
$\mathbf{e}_s$	Input of the AE
$f_{\boldsymbol{\theta}_s}(\mathbf{1}_s, x_s)$	NN encoder of SWAE at source node
$f_{\boldsymbol{\theta}_s}(\mathbf{u}_s, x_s)$	NN encoder of BWAE at source node
g_s	Modulation process
$g_{\boldsymbol{\theta}_d}(y_d, \hat{\mathbf{1}}_s)$	NN decoder of SWAE at destination node
$g_{\boldsymbol{\theta}_d}(y_d, \mathbf{l})$	NN decoder of BWAE at destination node
H	Entropy
h_{rd}	Channel between relay and destination node
h_{sr}	Channel between source and relay node
I	Mutual information
$\mathcal{I}^{\text{BWAE}}$	Estimated MI of BWAE
$\mathcal{I}^{\text{SWAE}}$	Estimated MI of SWAE
k	Total number of bits
L	Total number of dense layers in the AE
LLR	Log-likelihood ratio
$\mathcal{L}(f_{\boldsymbol{\theta}_s}, g_{\boldsymbol{\theta}_d})$	Cross-entropy loss
$\mathbf{l} \in \mathbb{R}^k$	k logits output at destination node in BWAE
M	Total constellation points (codewords)
n	Total number of channel-reuse (block length)
n_d	AWGN at destination node
n_r	AWGN at relay node
P_r	Transmission power of relay node
P_s	Transmission power of source node
$\mathbf{P_N}$	Power-normalization layer
$\tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s \vert y_d)$	Probabilities over 2^k output message in SWAE
$p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s \vert y_d)$	Probabilities over 2^k input message in SWAE
$p_{f_{\boldsymbol{\theta}_s}}(u_s^m \vert y_d)$	Probabilities over k input bits in BWAE
$\tilde{p}_{g_{\boldsymbol{\theta}_d}}(u_s^m \vert y_d)$	Probabilities over k output bits in BWAE
Q_{train}	Training dataset
Q_{test}	Testing dataset
$\mathbf{r}_l \in \mathbb{R}^{\delta_l}$	Bias terms in l^{th} layer
s	Symbol decoded by the decoder of SWAE
$\mathcal{S}$	Training dataset SNR values
t	Iteration number (epochs)
$u_s \in \{0,1\}^k$	Bits input at source node
$\hat{u}_s \in \{0,1\}^k$	Bits decoded at destination node
$\mathbf{W}_l \in \mathbb{R}^{\delta_{l-1} \times \delta_l}$	Weight matrix between $(l-1)^{\text{th}}$ and l^{th} layer
x_s	Source node signal
y_d	Signal received at destination node
y_r	Signal received at relay node
$\mathbf{Z}^{(l)}$	Output of l^{th} layer

of the nodes has a single antenna and the direct link between the S and D nodes is strongly attenuated because of severe path-loss and shadowing, and the communication can take place only via the AF relay node (R). We consider block-by-block encoding and decoding operation at the source and destination nodes, whereas the signal transmission in each phase takes place as symbol-by-symbol[1]. We consider independent and identically distributed (i.i.d.) Rayleigh block fading (RBF) channels $\sim \mathcal{CN}(0,1)$, such that it remains constant for the n transmissions (block length) in each phase,

[1]The encoder at the source node takes k bits as input and converts it to j bits via channel coding and then transmit the modulated n symbols to the destination node via $2n$ independent channel reuse in two-phases, then the destination node takes n symbols demodulates it to j bits and then perform channel decoding to get the k intended bits. Thus rate for the AF relay network becomes $R = k/2n$ [bits/channel use]. For the sake of clarity in explanation, we consider $n = 1$ in this subsection.

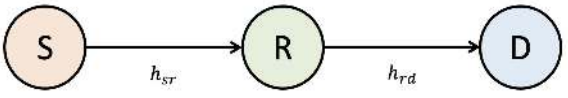


FIGURE 1: System model for AF relay networks.

while changes randomly in each phase and with time. In the first phase, the source node transmits $u_s \in \{0,1\}^k$ bits by mapping u_s to complex baseband symbol $x_s = g_s(u_s) \mapsto \mathbb{C}$, where $g_s(\cdot)$ denotes the modulation process, such that $\mathbb{E}\{|x_s|^2\} = 1$. The signal received by the AF relay node (R) can be given by

$$y_r = \sqrt{P_s} h_{sr} x_s + n_r \tag{1}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3085901, IEEE Access

where P_s represents the source transmission power, h_{sr} denotes the channel in the first phase transmission, and n_r is the additive white Gaussian noise (AWGN) at the AF relay node with $n_r \sim \mathcal{CN}(0, \sigma_r^2)$.

In the second phase, the relay node performs symbol-wise amplification with the amplification factor represented as

$$\alpha = \left(P_s|h_{sr}|^2 + \sigma_r^2\right)^{-1/2} \tag{2}$$

And re-transmit the amplified signal to the destination node (D), given by

$$\begin{aligned} y_d &= \sqrt{P_r} h_{rd} \alpha y_r + n_d \\ &= \underbrace{\sqrt{P_s P_r} h_{rd} h_{sr} \alpha x_s}_{\text{Intended Signal}} + \underbrace{\sqrt{P_s P_r} h_{rd} h_{sr} \alpha n_r + n_d}_{\text{Noise}} \end{aligned} \tag{3}$$

where P_r is the transmission power of the relay node, h_{rd} denotes the channel in the second phase transmission, and n_d is the AWGN with $n_d \sim \mathcal{CN}(0, \sigma_d^2)$. The destination node decodes the intended signal u_s by using the optimal maximum-likelihood detector (MLD) as

$$\hat{u}_s = \arg\min_{x \in \mathcal{C}} \left|\left| y_d - \sqrt{P_s P_r} h_{rd} h_{sr} \alpha x \right|\right|^2 \tag{4}$$

where $\mathcal{C}$ denotes all the possible alphabets, for example $\pm\sqrt{1/2} \pm \sqrt{1/2}i$ (for QPSK), etc.

In the differential scenario, i.e. without the CSI knowledge, we utilize traditional differential modulation and demodulation techniques at the source and destination node, such as differential QPSK (d-QPSK) and MLD decoding. The traditional differential schemes are near optimal because there is no selection combining cooperative diversity at the destination nodes [22]– [28]. However, the amplification factor designed in (2) utilizes the channel gain information and noise variances. In the case of absence of the CSI knowledge, there are two distinct ways proposed in literature to design the amplification factor, as detailed below:

- *Transmit power-based amplification factor* (TP-based α) – This approach decides the amplification factor on the basis of the transmission power of the source node [22], [23], as follows

$$\alpha = \left(P_s + 1\right)^{-1/2} \tag{5}$$

- *Channel statistics-based amplification factor* (CS-based α) – This approach utilizes the second-order statistics of the first hop channel between the source and relay node $\sigma_{sr}^2 = \mathbb{E}\{|h_{sr}|^2\}$ and noise variance at the relay node to determine the amplification factor [24]– [28], given by

$$\alpha = \left(P_s \sigma_{sr}^2 + \sigma_r^2\right)^{-1/2} \tag{6}$$

III. PROPOSED AUTOENCODERS FRAMEWORK FOR AF RELAY NETWORKS

In this section, we propose the end-to-end learning-based symbol-wise and bit-wise AE framework for AF relay network, as detailed in Fig. 2 and Fig. 3, respectively.

A. SYMBOL-WISE AE (SWAE) FRAMEWORK FOR AF RELAY NETWORKS

The source node's input message is a one-hot representation vector $\mathbf{1}_s \in \{1, ..., 2^k\}$ of the 2^k possible symbols of which only one of the element is 1 while the rest are zeros. The source node aims to map the one-hot encoded vector to a complex baseband symbol x_s, by a mapping function $f_{\boldsymbol{\theta}_s}(\mathbf{1}_s, x_s) : \mathbf{1}_s \mapsto x_s \in \mathbb{C}$, where $f_{\boldsymbol{\theta}_s}$ is the trainable parameters of the NN encoder with weights and bias terms. We impose a power normalization constraint on the output of the encoder, such that $||f_{\boldsymbol{\theta}_s}(\mathbf{1}_s, x_s)||_2^2 = 1$, and the signal received by the relay node can be given as $y_r = \sqrt{P_s} h_{sr} f_{\boldsymbol{\theta}_s}(\mathbf{1}_s, x_s) + n_r$. The relay node is a conventional AF relay node, thus the received signal is amplified as $x_r = \alpha y_r$. The amplified signal is re-transmitted over the second phase and the signal received by the destination node is given as $y_d = \sqrt{P_r} h_{rd} x_r + n_d$. The destination node implements the de-mapping $g_{\boldsymbol{\theta}_d}(y_d, \hat{\mathbf{1}}_s) : y_d \in \mathbb{C} \mapsto \hat{\mathbf{1}}_s \in \mathbb{R}^{2^k}$, where $g_{\boldsymbol{\theta}_d}$ denotes the trainable parameters of the NN decoder and the 2^k outputs represent the decoded one-hot vector representation, also referred as *logits* [29]. These outputs are then passed through a *softmax* function, $\sigma(\mathbf{x})_i = e^{x_i} / \sum_{i=1}^{2^k} e^{x_i}$ to obtain the probabilities over the 2^k output message denoted by $\tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s|y_d)$. Now, we utilize the categorical CE loss [20] to train the symbol-wise AE, averaged over y_d, as follows:

$$\mathcal{L}(f_{\boldsymbol{\theta}_s}, g_{\boldsymbol{\theta}_d}) := \mathbb{E}_{y_d}\left[H\left(p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s|y_d), \tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s|y_d)\right)\right] \tag{7a}$$

$$= -\mathbb{E}_{y_d}\left[\sum_{i=1}^{2^k} p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^i|y_d) \log \tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s^i|y_d)\right] \tag{7b}$$

$$= -\sum_{m=1}^{2^k} \int_{y_d} p(y_d) p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^i|y_d) \log \tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s^i|y_d)\, dy_d \tag{7c}$$

$$= \sum_{m=1}^{2^k} \int_{y_d} p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^i, y_d) \log \left[\frac{p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^i|y_d)}{p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^i|y_d) \tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s^i|y_d)}\right] dy_d \tag{7d}$$

$$= \sum_{m=1}^{2^k} \int_{y_d} p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^i, y_d) \log \left[\frac{p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^i|y_d)}{\tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s^i|y_d)}\right] dy_d - \sum_{m=1}^{2^k} \int_{y_d} p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^i, y_d) \log p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^i|y_d)\, dy_d \tag{7e}$$

$$= D_{KL}\left(p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s|y_d)||\tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s|y_d)\right) + H_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s|y_d) \tag{7f}$$

$$= D_{KL}\left(p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s|y_d)||\tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s|y_d)\right) + H_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s) - I_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s; Y_d) \tag{7g}$$

Where (7a) defines the categorical CE, (7b) comes from the definition of CE, (7c) opens the expectation along y_d, in (7d) we multiply and divide by $p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s^i|y_d)$, in (7e) we open the log function, (7f) comes from the definition of KL divergence and entropy, and (7g) utilizes the identity $H(a|b) = H(a) - I(a;b)$ [20]. By minimizing the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3085901, IEEE Access

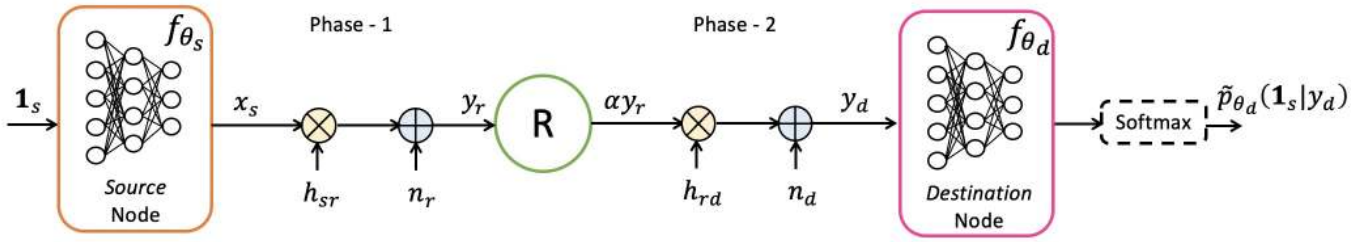


FIGURE 2: Symbol-wise autoencoder (SWAE) framework for amplify-and-forward relay networks.

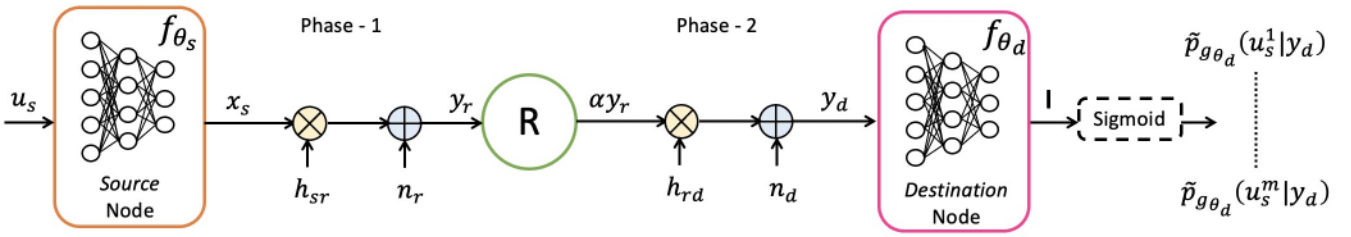


FIGURE 3: Bit-wise autoencoder (BWAE) framework for amplify-and-forward relay networks.

categorical CE loss in (7g), we are optimizing the SWAE framework to maximize the symbol-wise MI between input bits at the source node and received signal at the destination node $I_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s; Y_d)$, while minimizing the KL-divergence $D_{KL}\left(p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s|y_d)||\tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s|y_d)\right)$ between the posterior distributions learned at the encoder and the prior distribution learned at the decoder. Further, the entropy of the one-hot vector $H_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s)$ remains constant.

We can now obtain the estimated MI, which is defined as the MI subtracted by the relative entropy between the learnt distributions at the encoder and decoder. For the AF relay network, the estimated MI will be divided by 2 because the transmission takes place in two time-slots, given as $\mathcal{I}^{\text{SWAE}} = \tilde{I}^{\text{SWAE}}_{f_{\boldsymbol{\theta}_i}}(\mathbf{1}_s; Y_d)/2$. By rearranging the terms in (7g) we have

$$\begin{aligned}\tilde{I}^{\text{SWAE}}_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s; Y_d) &\triangleq I_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s; Y_d) - D_{KL}\left(p_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s; y_d)|| \right. \\ &\qquad \left. \tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{1}_s; y_d)\right) \\ &= H_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s) - \mathcal{L}^{\text{SWAE}}(f_{\boldsymbol{\theta}_s}, g_{\boldsymbol{\theta}_d}) \qquad (8)\end{aligned}$$

Since the first term in (8) remains a constant, we can see that the estimated MI of a symbol-wise AE depends on the training loss only.

Once the symbol-wise AE is trained with the input-output of the network as a one-hot vector $\mathbf{1}_s$ representing the 2^k possible symbol for the k bits. We can obtain the symbol with highest probability as the decoded symbol at the destination node. However, we need to perform bit-labelling separately on the AE-based designed constellation to map the $\hat{\mathbf{1}}_s$ vector to $\hat{u}_s$ bit vector. But, bit-labelling remains a challenging task, especially as the modulation order increases, or while designing AE-based coded modulation in $2n$-dimensional space. Because in such scenarios, the AE-based designed modulation might not form grids as conventional QAM, leading to $2^k!$ possible combinatorial problem to be solved [17].

B. BIT-WISE AE (BWAE) FRAMEWORK FOR AF RELAY NETWORKS

The source node takes bits as input, given by $\mathbf{u}_s \in \{0,1\}^k$, and maps it to a symbol x_s, by mapping function $f_{\boldsymbol{\theta}_s}(\mathbf{u}_s, x_s) : \mathbf{u}_s \mapsto x_s \in \mathbb{C}$. We impose a power normalization constraint on the output of the encoder, such that $||f_{\boldsymbol{\theta}_s}(\mathbf{u}_s, x_s)||_2^2 = 1$, and the signal received by the relay node can be given as $y_r = \sqrt{P_s} h_{sr} f_{\boldsymbol{\theta}_s}(\mathbf{u}_s, x_s) + n_r$. We consider a conventional relay node, thus the received signal is amplified as $x_r = \alpha y_r$. The amplified signal is re-transmitted over the second phase channel and the signal received by the destination node is given as $y_d = \sqrt{P_r} h_{rd} x_r + n_d$. The destination node implements the demapping $g_{\boldsymbol{\theta}_d}(y_d, \mathbf{l}) : y_d \in \mathbb{C} \mapsto \mathbf{l} \in \mathbb{R}^k$. The destination node outputs k logits (one per bit) given by $\mathbf{l} \in \mathbb{R}^k$. Then we apply a *sigmoid* activation function $\sigma(x) = \frac{1}{1+e^{-x}}$ [29] on each of the k logits, to obtain the probabilities $\tilde{p}_{g_{\boldsymbol{\theta}_d}}(u_s^m|y_d)$, $m = 1, ..., k$. Furthermore, in [17] it is shown that the logits correspond to the log-likelihood ratios (LLRs) as:

$$LLR(m) := \log \frac{1 - \tilde{p}_{g_{\boldsymbol{\theta}_d}}(u_s^m = 0|y_d)}{\tilde{p}_{g_{\boldsymbol{\theta}_d}}(u_s^m = 0|y_d)} = l_s^m, \quad \forall m \qquad (9)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3085901, IEEE Access

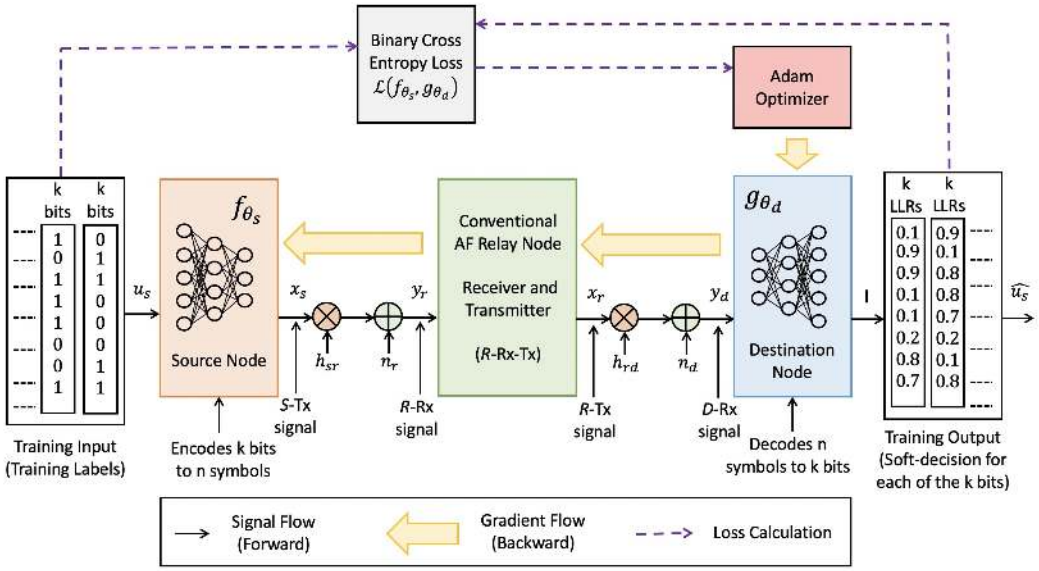


FIGURE 4: Block diagram of training for the proposed bit-wise AE-based end-to-end learning system.

where, $\tilde{p}_{g_{\boldsymbol{\theta}_d}}(u_s^m = 1|y_j) = \sigma(l_s^m)$. We train the bit-wise AE by minimizing the binary CE loss, averaged over y_d, as:

$$\begin{aligned}\mathcal{L}(f_{\boldsymbol{\theta}_s}, g_{\boldsymbol{\theta}_d}) &:= \sum_{m=1}^{k} \mathbb{E}_{y_d}\left[H\left(p_{f_{\boldsymbol{\theta}_s}}(u_s^m|y_d), \tilde{p}_{g_{\boldsymbol{\theta}_d}}(u_s^m|y_d)\right)\right] \\ &\triangleq D_{KL}\left(p_{f_{\boldsymbol{\theta}_s}}(\mathbf{u}_s|y_d)||\tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{u}_s|y_d)\right) \\ &\quad + H_{f_{\boldsymbol{\theta}_s}}(U_s) - I_{f_{\boldsymbol{\theta}_s}}(U_s; Y_d) \qquad (10)\end{aligned}$$

where, (10) is obtained following the similar steps as (7a)–(7g). In contrast to (7g) we can see that we are maximizing the bit-wise MI $I_{f_{\boldsymbol{\theta}_s}}(U_s; Y_d)$ in (10) and minimizing the KL divergence between the posterior distributions learned at the encoder and the prior distribution learned at the decoder $D_{KL}\left(p_{f_{\boldsymbol{\theta}_s}}(\mathbf{u}_s|y_d)||\tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{u}_s|y_d)\right)$. Further, entropy of bits $H_{f_{\boldsymbol{\theta}_s}}(U_s)$ remains constant and equal to the $H_{f_{\boldsymbol{\theta}_s}}(\mathbf{1}_s)$ in (7g).

Now, the estimated MI for the bit-wise MI is given as $\mathcal{I}^{\text{BWAE}} = \tilde{I}^{\text{BWAE}}_{f_{\boldsymbol{\theta}_i}}(U_s; Y_d)/2$ and by using (10), we have

$$\begin{aligned}\tilde{I}^{\text{BWAE}}_{f_{\boldsymbol{\theta}_s}}(U_s; Y_d) &\triangleq I_{f_{\boldsymbol{\theta}_s}}(U_s; Y_d) - D_{KL}\left(p_{f_{\boldsymbol{\theta}_s}}(\mathbf{u}_s|y_d)|| \right. \\ &\qquad \left. \tilde{p}_{g_{\boldsymbol{\theta}_d}}(\mathbf{u}_s|y_d)\right) \\ &= H_{f_{\boldsymbol{\theta}_s}}(U_s) - \mathcal{L}^{\text{BWAE}}(f_{\boldsymbol{\theta}_s}, g_{\boldsymbol{\theta}_d}) \qquad (11)\end{aligned}$$

Since the first term in (11) remains a constant, we can see that the estimated MI of a bit-wise AE depends on the training loss only.

Once the bit-wise AE is trained with the bits as its input-output, we can directly obtain the bit-labelling for designed complex baseband symbols x_s. We will later show that the bit-wise AE produces gray coded bit-labelling automatically. Thus, bit-wise AE removes the $2^k!$ complexity of bit-labelling required in the symbol-wise AE.

In contrast to the previous bit-wise AE works on P2P networks in [17] where the SNR information was required to train an AE, and two-way AF relay networks in [18] where a separate NN was created for each SNR, we also remove the necessity to acquire the SNR information for correctely encoding or decoding the signals. In fact, we create a single bit-wise AE framework that can be tested on a range of SNRs reducing the floating parameters in the NN by 95% [13].

IV. IMPLEMENTATION OF AUTOENCODER FRAMEWORK – PARAMETERS AND TRAINING

As the output of the bit-wise AE are bits and that of symbol-wise AE are symbols. We can formulate the proposed AF relay assisted AE-based framework as a multilabel (multiclass) classification task for the bit-wise (symbol-wise) AEs. We can optimize the bit-wise and symbol-wise AE framework by maximizing the minimum Euclidean distance between the constellation points, but it is not an optimal metric for designing channel-coding [30], [31]. More specifically, optimizing the binary CE and categorical CE loss obtained in (10) and (7g) is a better metric, as it directly optimizes the AE frameworks by maximizing the bit-wise MI and symbol-wise MI, respectively [30], [31]. In fact, the minimization of the binary CE loss directly leads to the maximization of the generalized MI. Now, we detail the NN architecture for the encoder and decoder and process of training the AE framework for the CE losses below.

For the sake of brevity, in Fig. 4 we show the block diagram for the training of the proposed bit-wise AE framework for the AF relay networks, and a similar representation can be done for the symbol-wise AE framework. In this work, we utilize dense layers in the neural network. Let there be L layers in a NN with $L-1$ hidden layers performing the encoding (or decoding) operations, wherein the number of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3085901, IEEE Access

TABLE 4: NN architecture of encoder.

Layer No. (l)	Nodes (δ_l^e)	Remarks
$l=0$	δ_0	*Input* (e_i)
$l=1$	64	$\sigma_1 = Tanh$
$l=2$	32	$\sigma_2 = Tanh$
$l=3$	16	$\sigma_3 = Tanh$
$l=4$	$2n$	$\sigma_4 = Linear$
$l=5$	$2n$	*Power normalization* ($\mathbf{P_N}$)

nodes in the $l^{\text{th}} \in \{1, ..., L\}$ layer is given by δ_l. Then, the l^{th} dense layer [29] can be represented as

$$\mathbf{Z}^{(l)} = \sigma_l \left(\mathbf{W}_l \mathbf{Z}^{(l-1)} + \mathbf{r}_l \right) \tag{12}$$

where $\sigma_l(\cdot)$ denotes the activation function, $\mathbf{W}_l \in \mathbb{R}^{\delta_{l-1} \times \delta_l}$ is the weight matrix and $\mathbf{r}_l \in \mathbb{R}^{\delta_l}$ represents the bias terms. Now, we detail the AE frameworks for AF relay networks below.

A. DESIGNING NEURAL NETWORK-BASED ENCODER

The source node is equipped with its own NN-based encoder that performs block-by-block encoding, wherein k bits (or 2^k symbols) are modulated to n symbols. Since we perform joint channel coding and modulation design, this can be referred as *NN-based coded modulation design*. The encoder consists of $M+1$ layers with M dense layers and a power normalization (PN) layer, given as

$$f_{\boldsymbol{\theta}_s}(e_s, x_s) = \mathbf{P_N}\left(\sigma_M\left(\mathbf{W}_M \sigma_{M-1}\left(\mathbf{W}_{M-1}\sigma_{M-2}\left(... \right.\right.\right. \\ \left.\left.\left. \sigma_1\left(\mathbf{W}_1 \mathbf{e}_s + \mathbf{r}_1\right)...\right) + \mathbf{r}_{M-1}\right) + \mathbf{r}_M\right)\right) \tag{13}$$

where e_s represents the input to the encoder and $\mathbf{P_N}$ denotes the power normalization layer, with no trainable parameters, mandating that the transmission power is $||f_{\boldsymbol{\theta}_s}(e_s, x_s)||_2^2 = n$, given as

$$\mathbf{P_N}(\mathbf{X}) : ||\mathbf{X}||_2^2 = n \tag{14}$$

For the case of symbol-wise AE the input to the encoder becomes symbols, whereas the input to the encoder is bits for the bit-wise AE. Thus, the input to the encoder with the number of input nodes can be given as follows

$$\{e_s, \delta_0\} = \begin{cases} \{\mathbf{1}_s, 2^k\}, & \text{for Symbol-wise AE} \\ \{u_s, k\}, & \text{for Bit-wise AE} \end{cases} \tag{15}$$

For the sake of fair comparison between symbol-wise and bit-wise AE frameworks, we keep the encoder's NN architecture same for all the scenarios, as summarized in Table 4[2].

Note that the output of the power normalization layer is the output of the source node's encoder, i.e. $\mathbf{x}_s \in \mathbb{R}^{2n}$. As NN can support only real values, thus we have $2n$ outputs, where $\{1, ..., n\}$ denotes the real part and $\{n+1, ..., 2n\}$ denotes the imaginary part of the n complex baseband symbols.

[2]In particular, we consider $M = 4$ dense layers for designing the encoder, with $\delta_1^e = 64$, $\delta_2^e = 32$, $\delta_3^e = 16$, $\delta_4^e = 2n$. Also, we keep $\sigma_{\{1,2,3\}}$ as *Tanh* activation function, whereas σ_4 as *Linear* activation function. The *Tanh* activation function can be given as $\sigma(x) = tanh(x) = \frac{2}{1+e^{-2x}} - 1$, and the *Linear* activation function can be given as $\sigma(x) = linear(x) = x$ [29].

Also, note that the signal transmission takes place symbol-by-symbol.

B. DESIGNING AF RELAY NODE

The relay node is deigned on the basis of the presence or absence of the CSI knowledge, as follows:

- *With the CSI Knowledge* – In this case, we use a conventional AF relay node that amplifies the received signal using the amplification factor in (2) and re-transmits the signal to the destination node. The process of signal transmission–reception remains the same as conventional scenario for both the bit-wise and symbol-wise AEs, as shown in Table 5. Where, in the first time-slot the source node transmits the signal to the relay node while the relay node remains silent, and in the second time-slot the relay transmits the signal to the destination node while the source node remains silent.
- *Without the CSI Knowledge* – In this case, we propose two approaches as detailed below:
  - *Using Conventional AF Relay* – In this case, we use a conventional AF relay node that amplifies the received signal using the amplification factor in (5) or (6) depending if the amplification factor is determined using TP-based α or CS-based α method. Further, the process of signal transmission and reception remains the same as conventional scenario for both the bit-wise and symbol-wise AEs, as shown in Table 5.
  - *Using Power Normalized AF Relay* (PN at relay) – In this case, we modify the signal transmission and reception process, as detailed in Table 6. Herein, the source node transmits the complete block of data comprising n symbols to the relay node in the first n time-slots, while the relay node remains silent. Then the relay node performs amplification by normalizing the power of n symbols using the power normalization layer $\mathbf{P_N}$ given in (14), with no trainable parameters, mandating that the transmission power is $||\mathbf{x}_r||_2^2 = n$. Then in the next n time slots, the relay node performs symbol-by-symbol transmission, while the source node remains silent.

C. DESIGNING NEURAL NETWORK-BASED DECODER

The symbols re-transmitted by the AF relay node are decoded by the destination node. The decoder performs block-by-block decoding, where n symbols are demodulated to k bits. Since we perform joint channel decoding and demodulation it can also be referred as *NN-based coded demodulation design*. The decoder at the destination node consists of N dense layers, given as

$$g_{\boldsymbol{\theta}_d}(y_d, \hat{e}_s) = \sigma_N\left(\mathbf{W}_N \sigma_{N-1}\left(\mathbf{W}_{N-1}\sigma_{N-2}\left(...\sigma_1\left(\right.\right.\right. \\ \left.\left.\left. \mathbf{W}_1 \mathbf{L_L}\left(\mathbf{y}_d\right) + \mathbf{r}_1\right)...\right) + \mathbf{r}_{N-1}\right) + \mathbf{r}_N\right) \tag{16}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3085901, IEEE Access

TABLE 5: Process of signal transmission–reception of a block of data (block length n) for conventional AF relay network, and AE-based scenarios – with CSI knowledge and without CSI knowledge using conventional AF relay.

Time-instants $[\kappa]$	$[1]$	$[2]$	$[3]$	$[4]$	$\cdots\cdots$	$[2n-1]$	$[2n]$
Symbol Tx by S	$x_s[1]$	–	$x_s[2]$	–	$\cdots\cdots$	$x_s[n]$	–
Symbol Rx by R	$y_r[1]$	–	$y_r[2]$	–	$\cdots\cdots$	$y_r[n]$	–
Symbol Tx by R	–	$x_r[1]$	–	$x_r[2]$	$\cdots\cdots$	–	$x_r[n]$
Symbol Rx by D	–	$y_d[1]$	–	$y_d[2]$	$\cdots\cdots$	–	$y_d[n]$

TABLE 6: Process of signal transmission–reception of a block of data (block length n) for AE-based scenarios without CSI knowledge using power normalized AF relay.

Time-instants $[\kappa]$	$[1]$	$[2]$	$\cdots\cdots$	$[n]$	$[n+1]$	$[n+2]$	$\cdots\cdots$	$[2n-1]$	$[2n]$
Symbol Tx by S	$x_s[1]$	$x_s[2]$	$\cdots\cdots$	$x_s[n]$	–	–	$\cdots\cdots$	–	–
Symbol Rx by R	$y_r[1]$	$y_r[2]$	$\cdots\cdots$	$y_r[n]$	–	–	$\cdots\cdots$	–	–
Symbol Tx by R	–	–	$\cdots\cdots$	–	$x_r[1]$	$x_r[2]$	$\cdots\cdots$	$x_r[n-1]$	$x_r[n]$
Symbol Rx by D	–	–	$\cdots\cdots$	–	$y_d[1]$	$y_d[2]$	$\cdots\cdots$	$y_d[n-1]$	$y_d[n]$

TABLE 7: NN architecture of decoder.

Layer No. (l)	**Nodes** (δ_l)	**Remarks**
$l=0$	$2n$	*Input (Output of* $\mathbf{L_L}$*)*
$l=1$	256	$\sigma_1 =$ *Tanh*
$l=2$	128	$\sigma_2 =$ *Tanh*
$l=3$	64	$\sigma_3 =$ *Tanh*
$l=4$	32	$\sigma_4 =$ *Tanh*
$l=5$	δ_0	$\sigma_5 =$ *Softmax (for SWAE) or* $\sigma_5 =$ *Sigmoid (for BWAE)*

where $\hat{e}_s$ is the output at the decoder corresponding to input e_s at the encoder, and $\mathbf{L_L}$ denotes the L Lambda layers with no trainable parameters. Please note that the first lambda layer takes received symbols as input and the output of the last lambda layer forms the input to the NN decoder. The Lambda layers changes depending on the scenarios, thus detailed separately in Section V. For the sake of fair comparison between symbol-wise and bit-wise AE frameworks, we keep the decoder's NN architecture same for all the scenarios, as summarized in Table 7[3].

D. MODEL TRAINING AND UPDATES

Similar to the previous works [13]– [18], we propose to employ NNs consisting of dense layers at the source and destination nodes that constitute the AE framework, both of these NNs are trained jointly while minimizing the CE loss, also referred as end-to-end learning. In particular, the input to the NN encoder at the source node is bits (for BWAE) or symbols (for SWAE), whereas the input to the NN decoder at the destination node is the amplified signal by the AF relay node, distorted in two-phase relay transmission. We train this AE framework, such that it becomes unaffected by testing SNR and can handle the two-hop fading channels effectively. Besides, we are using a conventional AF relay node, thus we do not need to perform explicit training at the AF relay node, unlike the works in [13]– [18]. Once trained, we deploy the NN weights at the source and destination nodes for future predictions. Later in this AE-framework, the NN-based source node communicates with the NN-based destination node via an AF relay node.

The expected loss for AE-based AF relay network is given by $\mathcal{L}(f_{\boldsymbol{\theta}_s}, g_{\boldsymbol{\theta}_d})$. Depending on the constructed symbol-wise or bit-wise AE the loss can be categorical or binary CE loss as shown in (7g) and (10), respectively. Let there be $\{Q_{\text{train}}, Q_{\text{test}}\}$ training and testing samples, then expected loss can be estimated via sampling [29] as

$$\mathcal{L}(f_{\boldsymbol{\theta}_s}, g_{\boldsymbol{\theta}_d}) = \frac{1}{B}\sum_{q=1}^{Q} \mathcal{L}_{f_{\boldsymbol{\theta}_s}, g_{\boldsymbol{\theta}_d}}(e_q, \hat{e}_q) \tag{17}$$

where B denotes the batch size, and since we train the designed NN for all the Q_{train} training samples, thus $q = \{1, ..., Q\}$, where $Q = \frac{Q_{\text{train}}}{B}$ and $\{e_q, \hat{e}_q\}$ denotes the training input-output of the AE. Thus, we can write the objective function as follows

$$\min_{f_{\boldsymbol{\theta}_s}, g_{\boldsymbol{\theta}_d}} \mathcal{L}(f_{\boldsymbol{\theta}_s}, g_{\boldsymbol{\theta}_d}) \tag{18}$$

The most widely employed method to solve the optimization problem is stochastic gradient descent (SGD) method, wherein for the NNs, the gradient is obtained by back propagation method [29]. Herein, we update the parameter set $\boldsymbol{\pi} = \{f_{\boldsymbol{\theta}_s}, g_{\boldsymbol{\theta}_d}\}$ iteratively in the SGD, as follows

$$\boldsymbol{\pi}^{(t)} = \boldsymbol{\pi}^{(t-1)} - \tau \nabla \mathcal{L}(\boldsymbol{\pi}^{(t-1)}) \tag{19}$$

where $\tau > 0, t, \nabla$ represent the learning rate, iteration index and gradient operator, respectively.

E. PREDICTIONS

The process of predictions vary depending on the designed bit-wise or symbol-wise AE frameworks as:

- *For Bit-Wise Autoencoders (BWAE)* – We have *Sigmoid* activation at the last layer of the decoder, giving us soft probabilistic outputs $\tilde{p}_{g_{\boldsymbol{\theta}_d}}(u_s^m|y_d), \forall\, m = \{1, ..., k\}$. Then for each $p^{\text{th}} = \{1, ..., Q_{\text{Test}}\}$ testing sample,

[3]In particular, we consider $N = 5$ dense layers for designing the encoder, with $\delta_1^d = 256$, $\delta_2^d = 128$, $\delta_3^d = 64$, $\delta_4^d = 32$, $\delta_5^d = \delta_0$. Also, we keep $\sigma_{\{1,2,3,4\}}$ as *Tanh* activation function, whereas σ_5 as *Softmax* activation function for symbol-wise AE and σ_5 as *Sigmoid* activation function for bit-wise AE.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3085901, IEEE Access

the bits can be predicted $\hat{u}_s^{(m,p)}$ by keeping a simple threshold such as 0.5 on the $\tilde{p}_{g_{\boldsymbol{\theta}_d}}(u_s^{(m,p)}|y_d)$, as below

$$\hat{u}_s^{(m,p)} = \begin{cases} 0, & \text{if } \tilde{p}_{g_{\boldsymbol{\theta}_d}}(u_s^{(m,p)}|y_d) \leq 0.5, \\ 1, & \text{if } \tilde{p}_{g_{\boldsymbol{\theta}_d}}(u_s^{(m,p)}|y_d) > 0.5, \end{cases} \quad \forall\, m, p. \tag{20}$$

- *For Symbol-Wise Autoencoders (SWAE)* – We have *Softmax* activation at the last layer of the decoder, giving us output $\mathbf{1}_s := 1_s^n \in [0,1], \forall\, n = 1, ..., 2^k$, such that $\sum_{n=1}^{2^k} 1_s^n = 1$, where each of the element denotes a possible symbol. Then for each $q^{\text{th}} = \{1, ..., Q_{\text{Train}}\}$ training sample, we determine the symbol s^q with the largest probability, as follows

$$s^q = \arg \max_{n=1,...,2^k} \left(1_s^{(n,q)}\right), \quad \forall\, q. \tag{21}$$

Now, we have to perform bit-labelling, for the 2^k symbols. As detailed earlier bit-labelling can become a $2^k!$ combinatorial problem. Thus, we employ a heuristic approach to label the symbols, wherein we map the symbols to bits $s^q \mapsto \hat{u}_s^q$ according to the Gray codes. For example, for QPSK modulation, i.e. $(n.k) = (1, 2)$, we have 4 symbols as output, we label the symbols as $s_1 \mapsto \{0, 0\}$, $s_2 \mapsto \{0, 1\}$, $s_3 \mapsto \{1, 1\}$, and $s_4 \mapsto \{1, 0\}$, respectively. Then for each p^{th} testing sample we find the symbol s^p with the largest probability, as

$$s^p = \arg \max_{n=1,...,2^k} \left(1_s^{(n,p)}\right), \quad \forall\, p. \tag{22}$$

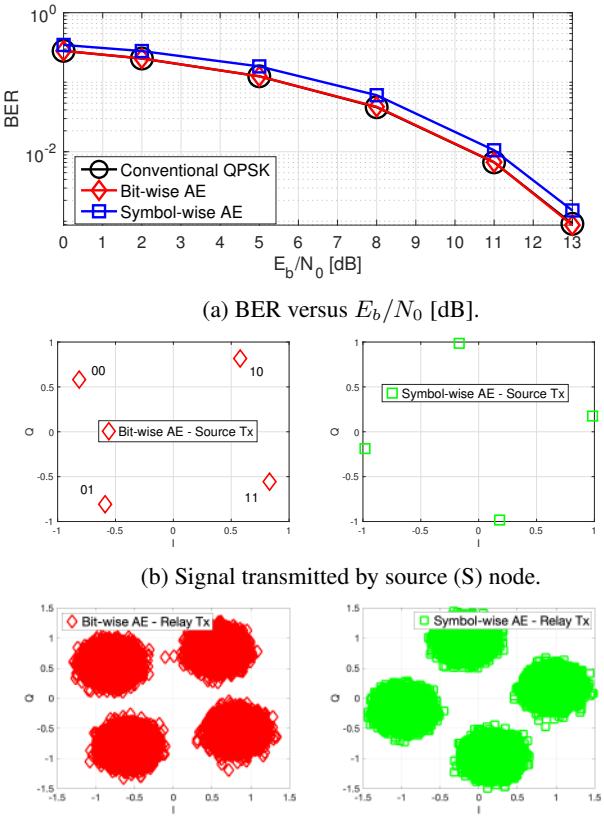
And utilize the bit-labelling done for the training dataset to map $\hat{u}_s^p$ from s^p.

Then, we calculate the bit-error-rate (BER) between the true bits intended to be transmitted (u_s^p) and predicted bits decoded at the receiver $(\hat{u}_s^p)$.

V. SIMULATION RESULTS

In this section, we evaluate the proposed bit-wise and symbol-wise AE frameworks for the AF relay networks with practical SNR values. We utilize QPSK modulation similar to [14]. To train the proposed architectures we utilize SGD with Adam optimizer [32], where the weights of the dense layers are initialized with the Glorot initializer [33]. We keep the learning rate $\tau = 0.00125$, batch size $B = 6000$, number of training epochs as 50, and transmission power of each node $P_s = P_r = 1$. We implement the proposed AE framework using Keras [34] with Tensorflow [35] as back-end. We show the performance for AWGN and RBF channels, where the channel remains constant during a transmission block of n symbols and then changes randomly. For the conventional scenarios, we utilize the optimal MLD in (4) at the destination node.

We have created the bit-wise and symbol-wise AE for the AF relay networks such that it remains *unaffected of testing SNR values*. In other words, we create a single AE model that can be deployed for any testing SNR. Thus, unlike the previous works in [17], [18], our proposed symbol-wise and bit-wise AE frameworks do not need the SNR value for prediction. For this, we create a training dataset of 10^5 samples for each of the SNR values in $\mathcal{S}$, and we test the designed models on the unseen testing dataset of 10^5 samples. For AWGN channels, we keep the $\mathcal{S} = \{4\}$ dB and for the RBF channels, we keep the $\mathcal{S} = \{3, 10, 23, 28, 38, 42\}$ dB. We note that for AWGN channels, only one SNR point brings the best AE performance, in particular, a low SNR point, intuitively this is because in the presence of large AWGN the AE learns to map the constellation points as far away as possible. For the RBF channels, we need multiple SNR points to obtain the best AE performance, ranging from low SNR to high SNR points, intuitively this is because the AE needs to learn to map constellation in the presence of both RBF channels in two-hop transmission and AWGN at the relay and destination node, thus when training only a low SNR point the AE observes only noise and thereby is not able to learn to map the constellation optimally leading to the stagnation in BER curves, especially in differential scenarios.



(a) BER versus E_b/N_0 [dB].

(b) Signal transmitted by source (S) node.

(c) Signal transmitted by relay (R) node.

FIGURE 5: Performance evaluation for Uncoded Scenarios $(n, k) = (1, 2)$ for the AF relay networks.

A. AE-BASED MODULATION DESIGN OF THE AF RELAY NETWORK UNDER AWGN CHANNELS

In this subsection, we evaluate the proposed *AE-based modulation design*, i.e. we keep the number of channels reused $n = 1$. Particularly we keep $(n, k) = (1, 2)$ and for sake of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3085901, IEEE Access

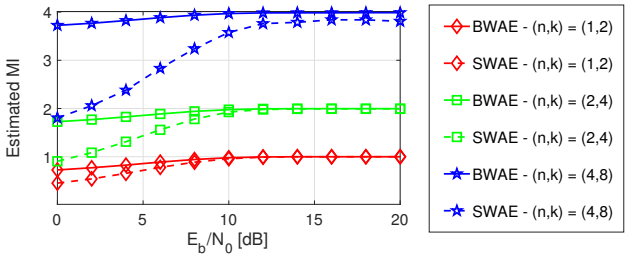


FIGURE 6: Estimated MI versus transmit SNR E_b/N_0.

clarity, we utilize AWGN channels by considering $h_{(\cdot)} = 1$, this assumption holds as there is no direct link between the source and destination link. Also, there are *no* Lambda layer at the decoders.

We show the transmit SNR (E_b/N_0) versus the BER performance in Fig. 5a. As the SNR increases the BER reduces. As we know that MLD is optimal for AWGN channels, we can see that the proposed bit-wise AE achieves performance similar to the optimal MLD of the conventional AF relay networks. Whereas the proposed symbol-wise AE performs ≈ 1 dB worse than the optimal MLD. This can be understood by the constellation learnt by the AEs in Fig. 5b. The constellation learned by the encoder of the source node in both the proposed bit-wise and symbol-wise AEs are four symbols for four possible combinations of bits $(k = 2)$. Thus, the NN-based encoder forms 2^k symbols for k input bits in both the bit-wise and symbol-wise AE. Further, bit-wise AE is leading to an automatic bit-labeling in Gray coding format, whereas bit-labeling for symbol-wise AE is done heuristically as detailed in Section IV-E. Furthermore, bit-wise AE leads to optimal rotation and translation leading to the performance gains, compared to the symbol-wise AE.

B. ESTIMATED MUTUAL INFORMATION OF THE AF RELAY NETWORK

We now compare the estimated MI for the symbol-wise and bit-wise AE obtained in (8) and (11) for the AF relay networks in Fig. 6. We keep $(n, k) = \{(1, 2), (2, 4), (4, 8)\}$ under AWGN channels. As the transmit SNR increases the estimated MI increases, until it reaches the upper bound $k/2$. This suggests the KL-divergence loss approaches 0 as E_b/N_0 increases indicating that the proposed AE frameworks can well approximate at a moderate E_b/N_0, but bit-wise AE shows $\approx$ 6 dB improvement in estimated MI at $E_b/N_0 = 0$ dB for AE-based modulation design $(n = 1)$. The improvement of bit-wise AE further increased with AE-based coded modulation design $(n > 1)$ where we see that bit-wise AE shows ≈ 10 dB improvement in estimated MI at $E_b/N_0 = 0$ dB. As the entropy term in (8) and (11) is equal, the estimated MI depends only on the classification errors (or the CE losses) calculated across the 2^k input-output for the SWAE compared to k input-output in the BWAE, and the low SNR regime leads to more classification errors in AE training with the additional $(2^k - k)$ number of classes in a symbol-wise AE.

C. AE-BASED DIFFERENTIAL CODED-MODULATION DESIGN UNDER RBF CHANNELS

In this subsection, we evaluate the performance of the proposed bit-wise and symbol-wise AE-based coded modulation design, i.e. we keep the number of channels reused $n > 1$, in particular, we keep $(n, k) = (7, 8)$, under RBF channels. Herein we consider a differential scenario, thus none of the (S, R, D) nodes has the CSI knowledge or noise variance information for any links. For the conventional scenarios, we consider $(7, 4)$ Hamming coding, along with differential QPSK and MLD decoding. We utilize TP-based α in (5) and CS-based α in (6) for the conventional and the proposed AE frameworks. Note that we do not have any Lambda layers at the decoder.

The *t-SNE* [21] is a widely adopted metric in the ML wireless community [1], [13], [16] for insights into the AE-based designed constellations in higher dimensional space, given as follows:

- *t-Stochastic Neighbour Embedding (t-SNE)* – In essence the t-SNE helps us to visualize the $2n$-dimensional data in 2 dimensions. This happens by transforming the similarities among data points to joint probabilities to decrease the KL divergence within the joint probabilities of the 2-dimensional embedding and the $2n$-dimensional constellation design.

For greater insights, we propose to evaluate the following metrics generally used for sphere packing[4] to characterize the optimality of designed coded-modulation in $2n$-dimensions [36]:

- *Minimum Euclidean distance* – We determine the minimum Euclidean distance between any two points as

$$d_{min} = \min_{u \neq v} \quad ||\mathbf{x}_u - \mathbf{x}_v||, \quad \forall \{u, v\} \in M \tag{23}$$

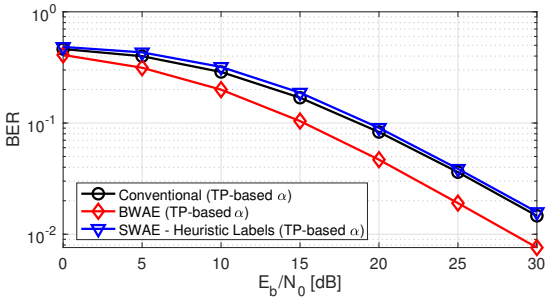
  where $\mathbf{x}_,...,\mathbf{x}_M$ denotes the M constellation points mapped in the $2n$-dimensional space.
- *Normalized second-order moment* – We can define this metric as the average squared Euclidean distance between a point in the packing and the origin of the coordinate system, normalized by the square of the minimum Euclidean distance, given as

$$E_n = (M d_{min}^2)^{-1} \sum_{u=1}^{M} ||x_u||^2 \tag{24}$$

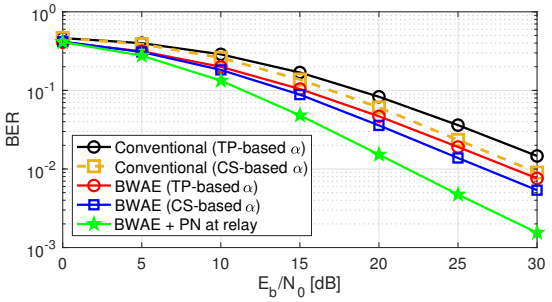
  This metric remains indifferent to scaling thus pivotal to differentiate the packing densities.

[4]We can define the problem of sphere packing as the packing of M points in a $2n$-dimensional space, with each point considered as the center of a $2n$-dimensional hypersphere of a given diameter and the aim remains to pack each of these spheres as densely as possible without overlapping, such that the Euclidean distance between any two points is above a defined value. Generally speaking, the problem of sphere packing increases with increase in n dimensions and becomes infeasible for large values of n. Whereas, the AE-based methods provide us an easy solution to learn the mappings in higher dimensional space.

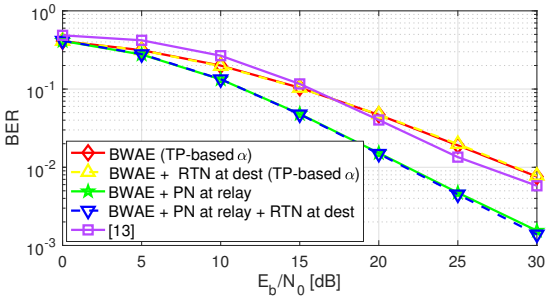
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3085901, IEEE Access



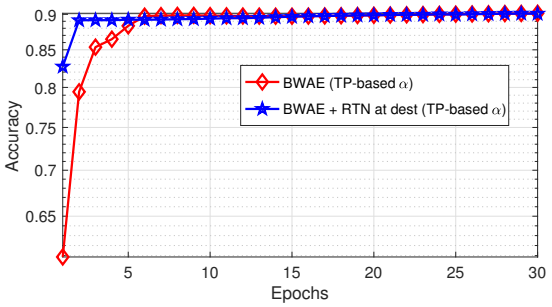
(a) Conventional versus Proposed AE Frameworks.



(b) Bit-wise AE with power normalization (PN) at relay node.

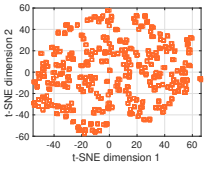


(c) RTN network in decoders and comparison with [13].

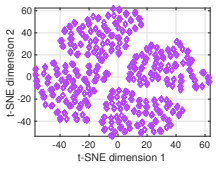


(d) Convergence of BWAE in presence of RTN in decoders.

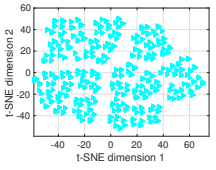
FIGURE 7: Performance evaluation for the differential coded-modulation AF relay networks.



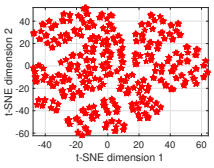
(a) SWAE



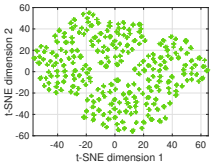
(b) BWAE



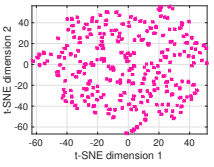
(c) BWAE + PN



(d) BWAE + PN + RTN



(e) BWAE + RTN.



(f) [13].

FIGURE 8: t-SNE representation in 2 dimensions.

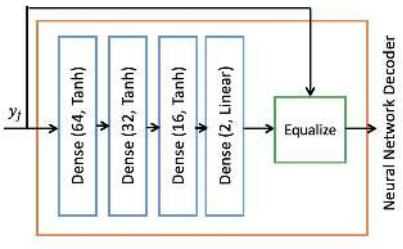


FIGURE 9: Radio transformer network (RTN)

- *Normalized fourth-order moment* or *kurtosis* – It measures the variation of the squared Euclidean norm among the constellation points, defined as

$$\chi = (E_n^2 d_{min}^4 M)^{-1} \sum_{u=1}^{M} ||x_u||^4 \qquad (25)$$

The $\chi = 1$ denotes that a spherical code is created with equal norm for all constellation points which is an optimal sphere packing if the number of points per dimension is small enough.

- *Constellation figure of merit* – This metric is the most suitable energy metric as the modulation constructions are being analyzed at the identical bandwidth, given as

$$\text{CFM} = n/E_n \qquad (26)$$

From Fig. 8, we can see that $256 = 2^8$ clusters are formed in the 2-dimensional space for all the proposed AE models, indicating that 2^k constellation points are formed while designing coded modulation for k bits in bit-wise AE and 2^k symbols in symbol-wise AE frameworks. Apart from this, we can not obtain any further intuition. Thus, we will focus on the other metrics hereafter.

1) Conventional versus Proposed AE Frameworks

We compare the performance of conventional (differential QPSK + $(7, 4)$ Hamming coded) and proposed AE frame-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3085901, IEEE Access

TABLE 8: Insights using other metrics for differential coded-modulation design

Model	**Minimum Euclidean Distance** d_{min} (↑ better)	**Normalized second moment** E_n (↓ better)	**Normalized fourth moment** χ	**Constellation Figure of Merit** CFM (↑ better)
Conventional Diff. QPSK + Hamming Code	0.76	12.1	1	0.6
State-of-the-art [13]	0.9	8.6	1	0.8
Symbol-wise AE	1.30	4.1	1	1.7
Bit-wise AE	**1.32**	**3.9**	**1**	**1.8**
Bit-wise AE + PN at relay	1.30	4.1	1	1.7
Bit-wise AE + RTN at dest.	1.27	4.3	1	1.6
Bit-wise AE + PN at relay + RTN at dest.	1.31	4.1	1	1.7

works using TP-based α in (5), in Fig. 7a[5]. From Table 8, we can see that the minimum Euclidean distance for the designed symbol-wise AE is 1.30 compared to only 0.76 in the conventional scenario, still the symbol-wise AE performs ≈ 0.8 dB worse than the conventional scenario. This can be explained by *Remark 1* below.

Remark 1: From Table 8 we have $\chi = 1$ indicating that a symbol-wise AE formulates spherical codes in $2n = 14$-dimensional space. Which is not of the form of a grid, thereby leading to a $2^8!$ combinatorial-problem for bit-labeling the designed symbols in the constellation map. Although we utilize a heuristic method to label the symbols as bits in the gray-coding format, it is still sub-optimal leading to the performance degradation of the symbol-wise AE.

In Fig. 7a, we can see that the bit-wise AE performs ≈ 3.5 dB better than the conventional scenario. This can be explained by the following *Remark 2*.

Remark 2: Firstly, we utilize the minimization of the binary CE loss which is proven to be equivalent to maximizing the generalized mutual information (GMI) of the relay network [18]. Secondly, we have shown in (10) that by minimizing the binary CE loss we maximize the bit-wise MI, which is closely related to achievable rate by the bit-metric decoding (BMD) [17]. Thirdly, we are designing joint channel coding and mdoulation design, thus the constellation design is taking place in $2n$-dimensional space, which from the modulation perspective, is leading to the maximization of the minimum Euclidean distance and minimization of packing density of the points to 1.32 and 3.9 in bit-wise compared to the 0.76 and 12.1 in conventional scenarios, as seen in Table 8; and from the coding perspective, is leading to the maximization of the minimum Hamming distance between the codewords. This is because we have already seen in AWGN channels that bit-wise AE learns bit-labeling in Gray coded format, thus the maximum error between two adjacent points is 1 bit. In contrast, the conventional differential-QPSK is taking place in only 2-dimensions with the addition of parity check bits using Hamming codes. Thus, end-to-end learning models can learn the coded-modulation design in n times higher dimensional space, such that no CSI is required at the decoder in the destination node to decode the signal.

[5]Please note that we see similar performance gains by using CS-based α in (6), thus for the sake of brevity, we show the performance with only TP-based α in Fig. 7a.

We can compare the bit-wise and symbol-wise AE performance with the following *Remark 3*.

Remark 3: In Table 8, we can see that the minimum Euclidean distance, packing density, and constellation figure of merit of the points designed by the symbol-wise AE and bit-wise AE becomes $\{1.30, 4.1, 1.7\}$ and $\{1.32, 3.9, 1.8\}$, respectively, both of which are very close to each other. Thus, minimizing the binary CE in bit-wise AE and categorical CE in symbol-wise AE almost forms similar coded-modulation design as a spherical code (since $\chi = 1$ in both the cases). The major difference in the BER performance ($\approx$ 5 dB as seen in Fig. 7a) comes from the fact that automatic bit-labeling is done in a gray coded format in bit-wise AE.

2) Using Power Normalized AF Relay Node

We know that as the transmit SNR increases the noise power at the relay node decreases, thus the amplification factor defined as (2) will increase. But as we do not have CSI knowledge or noise variance of the links in the TP-based α in (5), we obtain the constant amplification factor of $\alpha = 0.707$, which remains sub-optimal in low SNR regimes but remains not accurate for high SNR regimes. However, CS-based α in (6) utilizes the second-order channel statistics and noise variance to determine the amplification factor. As a result, in Fig. 7b we can see that the knowledge of channel statistics and noise variance at the relay node helps in designing more accurate amplification factor, leading to $\approx$ 2 dB gain with CS-based α in BER performance compared to the TP-based α. As now, we have understood the importance of determining the accurate amplification factor, we wanted to create an amplification factor that satisfies the following constraints – (1) does not include any deep learning layers at the relay node, and (2) does not require noise variance or CSI knowledge. Both of these conditions will be satisfied if we utilize a power normalization layer. Hence, we replace the process of constant amplification with a $\mathbf{P_N}$ layer as discussed in Section IV-B. In Fig. 7b, we can see that inclusion of the $\mathbf{P_N}$ layer brings performance improvement of $\approx$ 8.5 dB over the conventional scenario with TP-based α. This is because the $\mathbf{P_N}$ layer is helping in normalizing the n symbols' power to n, proving extremely beneficial especially for higher SNR regimes. Interestingly, in Table 8 we can see that bit-wise AE with power normalization layer at relay forms spherical codes with $\chi = 1$ and slightly worsen

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3085901, IEEE Access

the AE-based coded modulation design by reducing d_{min} and increasing E_n compared to conventional relay-based bit-wise AE. This indicates that the performance improvement by adding $\mathbf{P_N}$ layer is only because the PN layer is creating a better amplification factor at the relay node than designed with the CS or TP-based α.

3) Including an Additional RTN in Lambda Layers at the Destination Node

We propose a RTN as shown in Fig. 9, and evaluate the impact of including an RTN with the NN decoder in a bit-wise AE for the AF relaying (having cascaded channels) in Fig. 7c. In particular, we include RTN in Lambda layers in the NN decoder in Table 7. Including an RTN in the decoder of the proposed bit-wise AE gives the same BER performance as without an RTN in the decoder. Intuitively, this might be because we have cascaded channels from two-hops in AF relaying that needs to be decoded together, but also because even without RTN the decoder was able to decode the signals with higher accuracy so including an RTN is not helpful to improve the performance in AF relaying networks. In fact, from Table 8, we can see that including an RTN at the destination node slightly worsen the AE-based coded modulation design.

We check the convergence of the training accuracy of the bit-wise AE with and without an RTN in the decoder in Fig. 7d. The RTN in decoder starts with higher accuracy (or lower loss), and starts converging in 2 epochs, whereas if we do not have an RTN in decoder it starts with lower accuracy and needs approximately 7 epochs for convergence. We can see that including RTN will give the same accuracy as without RTN once 15 epochs are reached. Thus, RTN in destination node for the AF relay networks can be helpful in scenarios where re-training time plays an important role in deciding the deployment of the AE network in real-world scenarios, but at the expense of slightly worse AE-based coded modulation design.

4) Comparison of Proposed BWAE with [13]

We compare the proposed bit-wise AE with the state-of-the-art symbol-wise AE-based AF relay network in [13] in Fig. 7c. The authors in [13] utilized a NN-based relay node. In the proposed bit-wise AE, we utilize a conventional relay node with TP-based α in (5). Still, the proposed bit-wise AE with constant amplification factor performs better than [13] for up to $E_b/N_0 \leq 17$ dB and BER performance remains close thereafter. Further, if we utilize a power-normalization layer at the relay node, the proposed bit-wise AE always outperforms [13]. This is because in Table 8 we can see that the minimum Euclidean distance, second order moment and CFM of the proposed bit-wise and symbol-wise AE frameworks is $\{(1.32, 3.9, 1.8), (1.30, 4.1, 1.7)\}$ which is much better compared to the symbol-wise AE framework designed in [13] $\{0.9, 8.6, 0.8\}$. Thus, utilizing a conventional relay node with a power normalization layer is better than utilizing NN-based relay node, but also utilizing conventional AF relay node with TP-based α gives better performance compared to NN-based relay node up to moderate SNR, without utilizing any channel statistics or noise variance at the relay node. However, if we utilize CS-based α then a conventional AF relay node always outperforms the NN-based relay node.

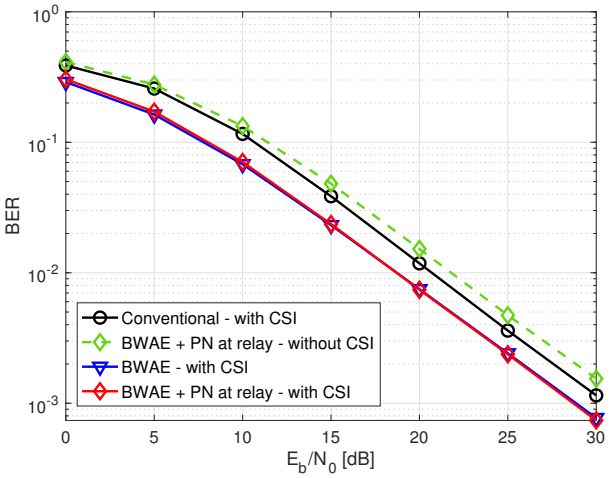


FIGURE 10: AF relay networks with CSI knowledge.

TABLE 9: Insights for coded-modulation design with CSI knowledge.

Metrics	*Conventional QPSK + Hamming Code*	*Bit-wise AE*	*Bit-wise AE + PN at relay*
Minimum Euclidean Distance d_{min} (↑ better)	1.4	**1.8**	1.7
Normalized second moment E_n (↓ better)	3.5	**2.3**	2.5
Normalized fourth moment χ	1	**1**	1
Constellation Figure of Merit CFM (↑ better)	2	**3.1**	2.9

D. AE-BASED CODED MODULATION DESIGN UNDER RBF CHANNELS

A major concern with previous works in [13], [18] was that the AF relay node had the unfair advantage of knowing the full CSI knowledge (thus additional phase information) and presence of deep learning layers (thus additional processing-power) at the relay node. In this subsection, we consider that the relay node only knows channel gains and noise variance information, and has no deep learning layers. Also, destination node has full CSI knowledge. We keep $(n, k) = (7, 8)$. For the conventional scenario, we consider $(7, 4)$ Hamming coding, along with QPSK modulation-demodulation and MLD decoding. We first detail the configuration of two Lambda layers for the decoder (in destination node) of the bit-wise AE with CSI knowledge – (1) in the first lambda

TABLE 10: Total parameters in the NN-based encoder, decoder and RTN.

NN Node	Total Parameters	
	BWAE	*SWAE*
NN-based Encoder	$3,422$	$19,294$
NN-based Decoder	$47,336$	$55,520$
NN-based RTN	$3,488$	$3,488$

layer, we perform channel equalization for the first-hop channel h_{sr} on received signal y_d, (2) in the second lambda layer, we perform channel equalization for the second-hop channel h_{rd} on the output of first lambda layer. Now the output after the second step is processed output of Lambda layers $\mathbf{L_L}$, given to the decoder in the destination node to predict the output e_s.

In Fig. 10, we see that as E_b/N_0 increases the BER reduces. Again, similar to the reasons mentioned in Remark 2, bit-wise AE (with CSI knowledge) performs better than the conventional scenario (with CSI knowledge) by $\approx$ 3 dB. We also evaluated the performance of the proposed bit-wise AE with a power-normalization layer, instead of the amplification factor, at the relay node. Unlike, the differential scenario in Fig. 7c, we see that BER performance remains the same with the PN relay, this is because we have varying amplification factor as detailed in (2) for conventional AF relays. Furthermore, in Table 9 we can see that spherical codes are formed ($\chi = 1$). Also, the bit-wise AE with conventional relay node obtains the superior coded-modulation design compared to the relay node with a power normalization layer, similar to the differential scenario in Table 8. Lastly, for sake of comparison to the differential scenario shown in Fig. 7c, we can see that even without CSI knowledge the proposed bit-wise AE with a power-normalization layer performs only 1 dB worse than conventional with CSI knowledge. Also, comparing the AE-based coded modulation design for the without CSI knowledge (differential) scenario and with CSI knowledge scenario using the Tables 8, 9, we can say that by using the CSI knowledge AE can design the coded-modulation by reducing the packing density to 2.3 compared to 3.9 in the scenario when no CSI knowledge is present, whereas the AE-based coded modulation designed for differential scenario is only short of 0.4 packing density compared to the conventional scenario with CSI knowledge.

E. COMPUTATIONAL COMPLEXITY AND TIME-COST ANALYSIS

In this subsection, we detail the computational complexity and time-cost analysis for the relay with/without CSI knowledge for all the proposed AE-based end-to-end learning frameworks below

- *Total Number of Parameters* – We consider dense layers in this work, which have associated weights $\mathbf{W}_l \in \mathbb{R}^{\delta_{l-1}\times\delta_l}$ and bias $\mathbf{r}_l \in \mathbb{R}^{\delta_l}$ terms as optimization parameters for each l^{th} dense layers. The total number of parameters in each of the proposed NN-based encoder, decoder and RTN (for both the BWAE and SWAE) can be given as

$$P = \sum_{l=0}^{J} \delta_l \times \delta_{l+1} + \sum_{l=1}^{J} \delta_l \tag{27}$$

  where for the NN-based encoder we have $J = 4$, $\delta_0 = k$ in BWAE, $\delta_0 = 2^k$ in SWAE; for the NN-based decoder we have $J = 5$, $\delta_5 = k$ in BWAE, $\delta_5 = 2^k$ in SWAE; and for the NN-based RTN we have $J = 4$, $\delta_0 = 2n$, respectively. Furthermore, the power normalization layer and Lambda layer in the proposed AE-based frameworks does not have any optimization parameters. For sake of summary, we provide the total parameters in the NN-based encoder, decoder and RTN in Table 10.

  Directly, the total number of optimization parameters in BWAE and SWAE can be calculated as the sum of individual optimization parameters in Table 10, which is much less, especially for a BWAE, than the AE-based AF relay network proposed in [13] where the total number of parameters are $114,286$.
- *Memory Space* – The memory space of the proposed AE-based frameworks directly depends on the total optimization parameters in the NN (detailed above). However in this work, we propose to utilize either a conventional or a PN-based AF relay node. As detailed in Table 5, the conventional AF relay does not require a memory buffer since it amplifies and re-transmit the received signal at each time-instant. However as detailed in Table 6, the PN-based AF relay requires a memory buffer to store n symbols to perform power normalization. Thus, utilizing a PN-based AF relay has a higher memory cost compared to the conventional AF relay node.
- *Training and Testing Cost* – In this work, we utilize an Intel Core i7-6700 CPU, with a GeForce RTX 2080 Ti GPU of 11 GB RAM on an Ubuntu 18.04.4 LTS OS to run our simulations both during the training and testing phase. We detail the training and testing time-cost for the with CSI and without CSI scenarios in Table 11, wherein we utilize $\approx$ 41% and $\approx$ 36% of the GPU during training and testing, respectively. In Table 11, we can make the following observations:

TABLE 11: Time-cost analysis.

CSI knowledge	AE-based Model	Training Cost (in sec.)	Testing Cost (in $\times 10^{-5}$ sec.)
Without CSI	*BWAE (TP-based α)*	33.2	3.7
Without CSI	*BWAE (CS-based α)*	33.2	3.7
Without CSI	*BWAE + PN at relay*	33.2	3.7
Without CSI	*BWAE + RTN at dest*	39.4	4.2
Without CSI	*BWAE + PN at relay + RTN at dest*	39.7	4.2
Without CSI	*SWAE (TP-based α)*	49.6	4.7
Without CSI	[13]	251.4	6.1
With CSI	*BWAE*	137.3	4.8
With CSI	*BWAE + PN at relay*	137.2	4.8

- The training and testing cost without CSI AEs is lesser compared to with CSI AEs, this is because in the scenario with CSI knowledge the proposed AE-based frameworks additionally include Lambda layers at the NN decoder that performs the channel equalization.
- BWAE takes lesser training time in comparison to the SWAE (even though they both have the same NN architectures) this is because BWAE has k input-output compared to 2^k input-output in SWAE leading to a larger number of parameters (as detailed in Table 10) in SWAE.
- BWAE takes lesser testing time in comparison to the SWAE this is because we have an additional step of performing heuristic-bit labeling in the SWAE.
- Including a PN layer at the relay node does not impact on training-testing time cost of the AE-based frameworks.
- In the case of without CSI knowledge, the RTN at the destination node increases the NN parameters in the AE (as detailed in Table 10) thereby increasing the training and testing time cost.
- The proposed AE frameworks take almost $6.6\times$ lesser training time and 39% lesser testing time in comparison to the AE-based AF relay networks in [13].

VI. CONCLUSION AND FUTURE WORKS

In this work, we propose end-to-end learning-based coded-modulation and differential coded-modulation designs in $2n$-dimensional space using the bit-wise and symbol-wise AE frameworks for the AF relaying network. Further, we propose to employ a conventional AF relay node instead of an NN-based relay node to minimize the implementation cost. We create a single AE model trained on multiple values of SNRs, that can be deployed for any testing SNR, without the need of the SNR value for prediction. We show that the NN-based encoder forms 2^k constellation points as a spherical code for both symbol-wise or bit-wise AE frameworks. Also, we show that minimizing the binary CE loss in bit-wise AE and categorical CE loss in symbol-wise AE almost forms a similar coded-modulation design as a spherical code. The major difference in the BER performance ≈ 5 dB comes from the fact that automatic bit-labeling is done in a gray-coded format in bit-wise AE, whereas we need to perform the bit-labeling in symbol-wise AE by solving a $2^k!$ combinatorial problem. We show that minimizing the binary CE loss for the bit-wise AE instead of the categorical CE loss for the symbol-wise AE leads to significant gains in estimated MI in low SNR regimes, while both the AEs converge to the upper bound of estimated MI at a similar moderate SNR. Furthermore, we show that the bit-wise AE takes lesser training and testing time in comparison to the symbol-wise AE because of automatic bit-labeling and reduced input-output dimensions.

The proposed AE frameworks are capable of decoding the signal without the CSI knowledge and noise variances of any links. Also, the traditional AF relay network is outperformed by the proposed AE by 3 dB. Moreover, including a power normalization layer at the relay node, that normalizes the n transmit symbols' power to n helps us to improve our performance by additional 5 dB. Further, including an RTN in the decoder of the proposed bit-wise AE gives the same BER performance as without an RTN in the decoder, but can be helpful in scenarios where re-training time plays an important role in deciding the deployment of the AE network. Furthermore, utilizing a conventional relay node with a power normalization layer is better than utilizing deep learning layers (or NN) at the relay node, but also utilizing a conventional AF relay node gives similar performance as the relay node with a NN. Lastly, we show that by using the CSI knowledge AE can design the coded-modulation by increasing the packing density by 1.5 compared to the differential scenario. Furthermore, the proposed bit-wise AE frameworks take almost $6.6\times$ lesser training time and 39% lesser testing time in comparison to the AE-based relay networks in [13].

As an end-to-end learning-based relay network using AE optimization is new paradigm research, we provide some insights of some future necessary research below:

- *Multi-User and Multi-Relay Networks* – We considered a simple single-user single-relay network in this work. Moreover, multi-user single-relay networks presents challenges of interference and noise amplification [2]– [4]. The authors in [12] considered a multi-user network, wherein firstly the interference strength is determined using DL-based approaches and then subsequently utilized to update the AE's decoders. Inspired by [12], we can consider a multi-user AF relay networks [2]– [4], wherein the interference due to multi-user from both the transmission hops can be estimated and employed to update the AE's decoders. Moreover, a single-user multi-relay network can be a direct extension of this work because we considered a conventional AF relay node and in the case of multiple relays, the proposed AE framework can be investigated. We can also extend this work by combining multi-user multi-relay AE frameworks by extending works in [2].
- *I/Q Imbalance and Additional Hardware Impairments* – In practice, relaying systems are compromised by the hardware impairments, e.g., in-phase (I) and quadrature-phase (Q) imbalance (IQI), power amplifier non-linearities, and phase noise [37], [38]. The IQI can be described as a phase and/or amplitude mismatch between the I and Q arms at the transmitter (Tx) and/or receiver (Rx) sides, leading to undesirable effects such as added image signal, frequency interference, etc, deteriorating the network performance [39]. Apart from the IQI, tackling the additional hardware impairments (AHI) is of notable importance as discussed by the authors in [37]. Since there exists no works considering

IQI and AHI in the AE-based frameworks literature, it will be an interesting research topic to analyze the performance of coded-modulation design using bit-wise AE frameworks in the presence of IQI and AHI.

- *Two-Way AF Relay Networks* – In this work, we considered a one-way AF relay network, wherein a terminal node communicates with another terminal node using a relay node in two-time slots and if both the terminal nodes want to communicate with each other, then we require four time-slots. However, in two-way relaying two terminal nodes communicate messages to each other at the same time using a relay node in two-time slots. In a two-way AF (TWAF) relay network, both the terminal nodes transmit their data simultaneously to the relay node, which then amplifies and re-transmit the amplified signal to the terminal nodes [40]. Although each terminal node can perform self-interference cancellation (SIC) to remove its signal, the major challenge of a TWAF relay network comes in the management of the interference of simultaneously received signals at the TWAF relay node and noise amplification with the amplification of the received signals [40]. The TWAF relay network was designed using a symbol-wise AE with BER metric in [13] and using a bit-wise AE with achievable-sum-rate (ASR) metric in [18]. However, both of these works [13], [18] employ NN-based re-encoding of the received signal at the TWAF relay node and utilize symbol-wise AE frameworks. Thus it will be interesting to research bit-wise AE-based coded-modulation designs for the TWAF relay networks with a conventional TWAF relay node.
- *Error Correction* – In this work, we consider Hamming codes as a baseline error correction scheme. Recently, the authors in [17] considered a bit-wise AE-based modulation design for P2P communication networks and designed an iterative LDPC decoding algorithm at the destination node. Inspired by [17], we can utilize the LLRs obtained in (9) in the proposed bit-wise AE frameworks and design more powerful iterative decoding based AE frameworks.

REFERENCES

[1] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer," in *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 4, pp. 563-575, Dec. 2017.

[2] A. Gupta, K. Singh and M. Sellathurai, "Time-Switching EH-Based Joint Relay Selection and Resource Allocation Algorithms for Multi-User Multi-Carrier AF Relay Networks," in *IEEE Transactions on Green Communications and Networking*, vol. 3, no. 2, pp. 505-522, June 2019.

[3] K. Singh, A. Gupta and T. Ratnarajah, "QoS-Driven Energy-Efficient Resource Allocation in Multiuser Amplify-and-Forward Relay Networks," in *IEEE Transactions on Signal and Information Processing over Networks*, vol. 3, no. 4, pp. 771-786, Dec. 2017.

[4] K. Singh, A. Gupta, T. Ratnarajah and M. Ku, "A General Approach Toward Green Resource Allocation in Relay-Assisted Multiuser Communication Networks," in *IEEE Transactions on Wireless Communications*, vol. 17, no. 2, pp. 848-862, Feb. 2018.

[5] S. Dörner, S. Cammerer, J. Hoydis and S. t. Brink, "Deep Learning Based Communication Over the Air," in *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 132-143, Feb. 2018.

[6] F. A. Aoudia and J. Hoydis, "Model-Free Training of End-to-End Communication Systems," in *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 11, pp. 2503-2516, Nov. 2019.

[7] N. Wu, X. Wang, B. Lin and K. Zhang, "A CNN-Based End-to-End Learning Framework Toward Intelligent Communication Systems," in *IEEE Access*, vol. 7, pp. 110197-110204, 2019.

[8] M. Sadeghi and E. G. Larsson, "Physical Adversarial Attacks Against End-to-End Autoencoder Communication Systems," in *IEEE Communications Letters*, vol. 23, no. 5, pp. 847-850, May 2019.

[9] X. Chen, J. Cheng, Z. Zhang, L. Wu, J. Dang and J. Wang, "Data-Rate Driven Transmission Strategies for Deep Learning-Based Communication Systems," in *IEEE Transactions on Communications*, vol. 68, no. 4, pp. 2129-2142, April 2020.

[10] M. E. Morocho-Cayamcela and W. Lim, "Accelerating wireless channel autoencoders for short coherence-time communications," in *Journal of Communications and Networks*, vol. 22, no. 3, pp. 215-222, June 2020.

[11] V. Raj and S. Kalyani, "Design of Communication Systems Using Deep Learning: A Variational Inference Perspective," in *IEEE Trans. on Cognitive Commun. and Networking*, vol. 6, no. 4, pp. 1320-1334, Dec. 2020.

[12] D. Wu, M. Nekovee and Y. Wang, "Deep Learning-Based Autoencoder for m-User Wireless Interference Channel Physical Layer Design," in *IEEE Access*, vol. 8, pp. 174679-174691, 2020.

[13] A. Gupta and M. Sellathurai, "End-to-End Learning-based Amplify-and-Forward Relay Networks using Autoencoders," *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, Dublin, Ireland, 2020, pp. 1-6.

[14] Y. Lu, P. Cheng, Z. Chen, Y. Li, W. H. Mow and B. Vucetic, "Deep Autoencoder Learning for Relay-Assisted Cooperative Communication Systems," in *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5471-5488, Sept. 2020.

[15] Y. Lu, P. Cheng, Z. Chen, W. H. Mow and Y. Li, "A Learning Approach to Cooperative Communication System Design," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 5240-5244.

[16] A. Gupta and M. Sellathurai, "A Stacked-Autoencoder Based End-to-End Learning Framework for Decode-and-Forward Relay Networks," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Barcelona, Spain, 2020, pp. 5245-5249.

[17] S. Cammerer, F. A. Aoudia, S. Dörner, M. Stark, J. Hoydis and S. ten Brink, "Trainable Communication Systems: Concepts and Prototype," in *IEEE Transactions on Communications*, vol. 68, no. 9, pp. 5489-5503, Sept. 2020.

[18] T. Matsumine, T. Koike-Akino and Y. Wang, "Deep Learning-Based Constellation Optimization for Physical Network Coding in Two-Way Relay Networks," *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, Shanghai, China, 2019, pp. 1-6.

[19] Y. Lu, P. Cheng, Z. Chen, W. H. Mow, Y. Li and B. Vucetic, "Deep Multi-Task Learning for Cooperative NOMA: System Design and Principles," in *IEEE Journal on Selected Areas in Communications*, vol. 39, no. 1, pp. 61-78, Jan. 2021.

[20] T. M. Cover and J. A. Thomas, *Elements of information theory.*, John Wiley & Sons, Nov. 2012.

[21] L. V. der Maaten, and G. E. Hinton,"Visualizing data using t-SNE," in *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008. [Online]. Available: https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf.

[22] M. M. Harb and M. F. Al-Mistarihi, "Dual Hop Differential Amplify-and-Forward relaying with selection combining cooperative diversity over Nakagami-m fading channels," *2016 8th IEEE International Conference on Communication Software and Networks (ICCSN)*, Beijing, China, 2016, pp. 225-228.

[23] M. R. Avendi and H. H. Nguyen, "Selection Combining for Differential Amplify-and-Forward Relaying Over Rayleigh-Fading Channels," in *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 277-280, Mar. 2013.

[24] Z. Fang, L. Zheng, L. Wang and L. Jin, "A frequency domain differential modulation scheme for asynchronous amplify-and-forward relay networks," *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, Chengdu, China, 2015, pp. 977-981.

[25] T. Himsoon, Weifeng Su and K. J. R. Liu, "Differential transmission for amplify-and-forward Cooperative communications," in *IEEE Signal Processing Letters*, vol. 12, no. 9, pp. 597-600, Sept. 2005.

[26] M. R. Avendi and H. H. Nguyen, "Performance of Selection Combining for Differential Amplify-and-Forward Relaying Over Time-Varying Chan-

nels," in *IEEE Transactions on Wireless Communications*, vol. 13, no. 8, pp. 4156-4166, Aug. 2014.

[27] Y. Lou, Y. Ma, Q. Yu, H. Zhao and W. Xiang, "A Differential ML Combiner for Differential Amplify-and-Forward System in Time-Selective Fading Channels," in *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 10157-10163, Dec. 2016.

[28] W. Cho, R. Cao and L. Yang, "Optimum Resource Allocation for Amplify-and-Forward Relay Networks With Differential Modulation," in *IEEE Transactions on Signal Processing*, vol. 56, no. 11, pp. 5680-5691, Nov. 2008.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, The MIT Press, 2016.

[30] T. Koike-Akino, P. Popovski and V. Tarokh, "Optimized constellations for two-way wireless relaying with physical network coding," in *IEEE Journal on Selected Areas in Commun.*, vol. 27, no. 5, pp. 773-787, June 2009.

[31] T. T. Nguyen and L. Lampe, "Bit-Interleaved Coded Modulation with Mismatched Decoding Metrics," in *IEEE Transactions on Communications*, vol. 59, no. 2, pp. 437-447, Feb. 2011.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980.

[33] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feed-forward neural networks," in *Proceedings International Conference AI Statistics*, vol. 9, pp. 249-256, May 2010.

[34] N. Ketkar, "Introduction to keras," *Deep Learning with Python, Springer*, pp. 97–111, 2017.

[35] Martìn Abadi et. al., "TensorFlow: Large-scale machine learning on heterogeneous systems," *Technical Report, Goggle Brain, arXiv*, 2015. [Online]:https://arxiv.org/abs/1605.08695.

[36] E. Agrell, "Database of sphere packings," [Online]: http://codes.se/packings, 2014, accessed Mar. 1, 2019.

[37] T. Schenk, *RF Imperfections in High-Rate Wireless Systems: Impact and Digital Compensation*, 1st ed. The Netherlands: Springer, 2008.

[38] A. E. Canbilen, et. al., "Impact of I/Q Imbalance on Amplify-and-Forward Relaying: Optimal Detector Design and Error Performance," *in IEEE Transactions on Communications*, vol. 67, no. 5, pp. 3154-3166, May 2019.

[39] J. Li, M. Matthaiou and T. Svensson, "I/Q Imbalance in AF Dual-Hop Relaying: Performance Analysis in Nakagami-m Fading," in *IEEE Transactions on Communications*, vol. 62, no. 3, pp. 836-847, March 2014.

[40] K. Singh, A. Gupta and T. Ratnarajah, "QoS-Driven Resource Allocation and EE-Balancing for Multiuser Two-Way Amplify-and-Forward Relay Networks," in *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, pp. 3189-3204, May 2017.

ANKIT GUPTA (Student Member, IEEE) received the B.Tech. degree in electronics and communication engineering from Guru Gobind Singh Indraprastha University, New Delhi, India, in 2015. He was with Aricent Technologies, Ltd., (Holdings), Gurugram, India, until 2017. He is currently pursuing the Ph.D. degree in signal processing and communication engineering with Heriot-Watt University. His current research interests include cooperative communications, non-orthogonal multiple access techniques, deep learning, autoencoders, end-to-end learning, coded-modulation designs, reinforcement learning and optimization methods in signal processing and communications.

MATHINI SELLATHURAI (Senior Member, IEEE) is currently a Full Professor in signal processing and intelligent systems with Heriot-Watt University, Edinburgh, U.K. In her 20-year research on signal processing for communications, she has made seminal contributions on MIMO wireless systems. She has authored or coauthored 200 IEEE entries. She has written a book and several book chapters in topics related to this project. She is a Fellow of HEA. She was a member for the IEEE SPCOM Technical Strategy Committee, from 2014 to 2018. She received the IEEE Communication Society Fred W. Ellersick Best Paper Award in 2005, the Industry Canada Public Service Award for contributions in science and technology in 2005, and the Best Ph.D. thesis medal from NSERC Canada, in 2002. She was also the General Co-Chair of the IEEE SPAWC2016 in Edinburgh. She was an Editor of the IEEE TSP from 2009 to 2014 and from 2015 to 2018. She has given invited talks.