Received 15 April 2021; revised 20 May 2021; accepted 31 May 2021. Date of publication 9 June 2021; date of current version 18 June 2021.

Digital Object Identifier 10.1109/OJCOMS.2021.3087815

Capacity-Driven Autoencoders for Communications

NUNZIO A. LETIZIA (Graduate Student Member, IEEE), AND ANDREA M. TONELLO (Senior Member, IEEE)

Institute of Networked and Embedded Systems, Chair of Embedded Communication Systems, University of Klagenfurt, 9020 Klagenfurt, Austria

CORRESPONDING AUTHOR: N. A. LETIZIA (e-mail: nunzio.letizia@aau.at)

ABSTRACT The autoencoder concept has fostered the reinterpretation and the design of modern communication systems. It consists of an encoder, a channel and a decoder block that modify their internal neural structure in an end-to-end learning fashion. However, the current approach to train an autoencoder relies on the use of the cross-entropy loss function. This approach can be prone to overfitting issues and often fails to learn an optimal system and signal representation (code). In addition, less is known about the autoencoder ability to design channel capacity-approaching codes, i.e., codes that maximize the input-output mutual information under a certain power constraint. The task being even more formidable for an unknown channel for which the capacity is unknown and therefore it has to be learnt. In this paper, we address the challenge of designing capacity-approaching codes by incorporating the presence of the communication channel into a novel loss function for the autoencoder training. In particular, we exploit the mutual information between the transmitted and received signals as a regularization term in the cross-entropy loss function, with the aim of controlling the amount of information stored. By jointly maximizing the mutual information and minimizing the cross-entropy, we propose a theoretical approach that a) computes an estimate of the channel capacity and b) constructs an optimal coded signal approaching it. Theoretical considerations are made on the choice of the cost function and the ability of the proposed architecture to mitigate the overfitting problem. Simulation results offer an initial evidence of the potentiality of the proposed method.

INDEX TERMS Digital communications, physical layer, statistical learning, autoencoders, coding theory, mutual information, channel capacity, explainable machine learning.

I. INTRODUCTION

COMMUNICATION systems have reached a high degree of performance, meeting demanding requirements in numerous application fields due to the ability to cope with real-world effects exploiting various accomplished physical system models. Reliable transmission in a communication medium has been investigated in the milestone work of Shannon [1] who suggested to represent the communication system as a chain of fundamental blocks, i.d., the transmitter, the channel and the receiver. Each block is mathematically modelled in a bottom-up fashion, so that the full system results mathematically tractable.

On the contrary, machine learning (ML) algorithms take advantage of the ability to work with and develop top-down models. In particular, deep learning (DL) has recently experienced a strong growth thanks to a larger availability of labeled data and increased computational power of processing units. Several fields have borrowed techniques from ML resulting in significant contributions and research progress. However, only in recent years, researchers have started adopting ML tools in the communication domain with promising results [2]–[4]. In particular, the communication chain has been reinterpreted as an autoencoder-based system [2], a deep neural network (NN) which takes as input a sequence of bits s, produces a coded signal, feeds it into a channel layer and tries to reconstruct the initial sequence from the channel output samples. The intermediate channel layer implicitly separates a prior neural block, the encoder, from the posterior one, the decoder. The encoder maps bits into symbols to be transmitted, ideally placing them into a coded signal that changes during the training process in order to mitigate the effects of the channel and noise. The decoder,

instead, performs a classification task and predicts the input message $\hat{s}$ from the received signal samples. The autoencoder can be trained end-to-end such that the block-error rate (BLER) of the full system is minimized. This idea pioneered a number of related works aimed at showing the potentiality of deep learning methods applied to wireless communications [4]–[7]. In [4], communication over-the-air has been proved possible without the need of any conventional signal processing block, achieving competitive bit error rates w.r.t. to classical approaches. Turbo autoencoders [5] reached state-of-the-art performance with capacity-approaching codes at a low signal to noise ratio (SNR). These methods rely on the a-priori channel knowledge (most of the time a Gaussian noise intermediate layer is assumed) and they fail to scale when the channel is unknown. To model the intermediate layers representing the channel, one approach is to use generative adversarial networks (GANs) [8]. GANs are a pair of networks in competition with each other: a generator model G that captures the data distribution and a discriminator model D that distinguishes if a sample is a true sample coming from real channel samples rather than a fake one coming from samples generated by G. In this way, the generator implicitly learns the channel distribution $p_Y(y|x)$, resulting in a differentiable network which can be jointly trained in the autoencoder model [9]–[11]. A recent work in [6] considered an AWGN channel with additive radar interference and demonstrated the autoencoders ability to produce optimal constellations in regions where no optimal solutions were available, outperforming standard configurations.

None of the aforementioned methods explicitly considered the information rate in the cost function. In this direction, the work in [12] included the information rate in the formulation and leveraged autoencoders to jointly perform geometric and probabilistic signal constellation shaping. Labeling schemes for the learned constellations have been discussed in [13], where the authors introduced the bit-wise autoencoder. If the channel model is not available, the encoder can be independently trained to learn and maximize the mutual information between the input and output channel samples, as presented in [14]. But therein the decoder is independently designed from the encoder and it does not necessarily grant error-free decoding. Indeed, the decoding stage may not have enough capacity to learn the demapping scheme nor converge during training, especially for large networks. Therefore, the encoder and decoder learning process shall be done *jointly*. In addition, the cost function used to train the autoencoder shall be appropriately chosen. With this goal in mind, let us firstly look into the historical developments and progresses made in the ML field, strictly related to the challenge considered in this paper.

The autoencoder was firstly introduced as a non-linear principle component analysis method, exploiting neural networks [15]. Indeed, the original network contained an internal bottleneck layer which forced the autoencoder to develop a compact and efficient representation of the input data, in an unsupervised manner. Several extensions have been further investigated, such as the denoising autoencoder (DAE) [16], trained to reconstruct corrupted input data, the contractive autoencoder (CAE) [17], which attempts to find a simple encoding and decoding function, and the k-sparse autoencoder [18] in which only k intermediate nodes are kept active. Autoencoders find application also in generative models as described in [19]. However, all of them are particular forms of *regularized* autoencoders. Regularization is often introduced as a penalty term in the cost function and it discourages complex and extremely detailed models that would poorly generalize on unseen data. In this context, the information bottleneck Lagrangian [20] was used as a regularization term to study the sufficiency (fidelity) and minimality (complexity) of the internal representation [21], [22]. So far, in the context of communication systems design, regularization in the loss function has not been introduced yet. In addition, the decoding task is usually performed as a classification task. In ML applications, classification is usually carried out by exploiting a final softmax layer together with the categorical cross-entropy loss function. The softmax layer provides a probabilistic interpretation of each possible bits sequence so that the cross-entropy measures the dissimilarity between the reference and the predicted sequence of bits distribution, $p(s)$ and $q(\hat{s})$, respectively. Nevertheless, training a classifier via cross-entropy suffers from the following problems: firstly, it does not guarantee any optimal latent representation. Secondly, it is prone to overfitting issues, especially in the case of large networks, thus, long codes design. Lastly, in the particular case of autoencoders for communications, the fundamental trade-off between the rate of transmission and reliability, namely, the channel capacity C, is not explicitly considered in the learning phase.

These observations made us rethinking the problem by formulating the two following questions.

a) Given a power constraint, is it possible to design capacity-approaching codes exploiting the principle of autoencoders?
b) Given a power constraint, is it possible to estimate channel capacity with the use of an autoencoder?

The two questions are inter-related and the answer of the first one provides an answer to the second one in a constructive way, since if such a code is obtained, then the distribution of the input signal that maximizes the mutual information is also determined, and consequentially the channel capacity can also be obtained.

Inspired by the information bottleneck method [20] and by the notion of channel capacity, a novel loss function for autoencoders in communications is proposed in this paper. The amount of information stored in the latent representation is controlled by a regularization term estimated using the recently introduced mutual information neural estimator (MINE) [23], enabling the theoretical design of nearly optimal codes. To the best of our knowledge, it is the first time that the influence of the channel appears in the *end-to-end learning* phase in terms of mutual information.

More specifically, the contributions of the paper are the following.

- A new loss function is proposed. It enables a new signal constellation shaping method.
- Channel coding is obtained by *jointly* minimizing the cross-entropy between the input and decoded message, and maximizing the mutual information between the transmitted and received signals.
- A regularization term β controls the amount of information stored in the symbols for a fixed message alphabet dimension M and a fixed rate $R < C$, playing as a trade-off parameter between error-free decoding ability and maximal information transfer via coding. The NN architecture is referred to as *rate-driven autoencoder*.
- In addition, the label smoothing regularization technique is used during the autoencoder learning process. An entropy description of the predicted messages is discussed and illustrated.
- By including the mutual information, we propose a new theoretical iterative scheme to built capacity-approaching codes of length n and rate R and consequently estimate channel capacity. This yields a scheme referred to as *capacity-driven autoencoder*.
- With the notion of explainable ML in mind, the rationale for the proposed metric and methodology is discussed in more fundamental terms following a) the concept of confidence penalty, b) the information bottleneck method [20] and c) by discussing the cross-entropy decomposition.

The remainder of the paper is organized as follows: In Section II, we briefly review the autoencoder principle and starting from the channel capacity concept, we intuitively motivate the presence of the mutual information in a new loss function. Section III discusses the mathematical foundation behind the new regularization term. In Section IV, we exploit the mutual information block previously introduced in a theoretical framework that iteratively designs capacity-approaching codes and learns the channel capacity. Section V presents an initial validation of the proposed methodology through numerical results. Finally, Section VI reports the conclusions.

Notation: X denotes a multivariate random variable of dimension d, while $x \in \mathcal{X}$ denotes its realization. Vectors $\mathbf{y}$ and matrices $\mathbf{Y}$ are represented using lower case bold and upper case bold letters, respectively. $p_Y(y|x)$ and $p_{XY}(x, y)$ represent the conditional and joint probability density functions, while $p_X(x)p_Y(y)$ is the product of the two marginals. $H(X)$ denotes the entropy of the random variable X, while $I(X; Y)$ denotes the mutual information between the random variables X and Y. Lastly, $D_{\mathrm{KL}}(p||q)$ is the Kullback-Leibler divergence of p from q.

II. RATE-DRIVEN AUTOENCODER

In this section, we introduce an autoencoder architecture to design a coding scheme that reaches a certain rate under a certain power constraint and code length. Then, we present the motivations behind the need of a new design metric that accounts for the mutual information in the classical cross-entropy loss function.

A. END-TO-END AUTOENCODER-BASED COMMUNICATIONS

The communication chain can be divided into three fundamental blocks: the transmitter, the channel, and the receiver. The transmitter attempts to communicate a message $s \in \mathcal{M} = \{1, 2, \ldots, M\}$. To do so, it transmits n complex baseband symbols $\mathbf{x} \in \mathbb{C}^n$ at a rate $R = (\log_2 M)/n$ (bits per channel use) over the channel, under a power constraint. In general, the channel modifies $\mathbf{x}$ into a distorted and noisy version $\mathbf{y}$. The receiver takes as input $\mathbf{y}$ and produces an estimate $\hat{s}$ of the original message s. From an analytic point of view, the transmitter applies a transformation $f : \mathcal{M} \to \mathbb{C}^n$, $\mathbf{x} = f(s)$ where f is referred to as the *encoder*. The channel is described in probabilistic terms by the conditional transition probability density function $p_Y(y|x)$. The receiver, instead, applies an inverse transformation $g : \mathbb{C}^n \to \mathcal{M}$, $\hat{s} = g(\mathbf{y})$ where g is referred to as the *decoder*. Such communication scheme can be interpreted as an autoencoder which learns internal robust representations $\mathbf{x}$ of the messages s in order to reconstruct s from the perturbed channel output samples $\mathbf{y}$ [2].

The autoencoder is a deep NN trained end-to-end using stochastic gradient descent (SGD). The encoder block $f(s; \theta_E)$ maps s into $\mathbf{x}$ and consists of an embedding layer followed by a feedforward NN with parameters θ_E and a normalization layer to fulfill a given power constraint. The channel is identified with a set of layers; a canonical example is the AWGN channel, a Gaussian noise layer which generates $y_i = x_i + w_i$ with $w_i \sim \mathcal{CN}(0, \sigma^2)$, $i = 1, \ldots, n$. The decoder block $g(\mathbf{y}; \theta_D)$ maps the received channel samples $\mathbf{y}$ into the estimate $\hat{s}$ by building the empirical probability mass function $p_{\hat{S}|Y}(\hat{s}|y; \theta_D)$. It consists of a feedforward NN, with parameters θ_D, followed by a softmax layer which outputs a probability vector of dimension M that assigns a probability to each of the possible M messages. The encoder and decoder parameters (θ_E, θ_D) are jointly optimized during the training process with the objective to minimize the categorical cross-entropy loss function

$$\mathcal{L}(\theta_E, \theta_D) = \mathbb{E}_{(x,y)\sim p_{XY}(x,y)}\Big[-\log\Big(p_{\hat{S}|Y}\big(\hat{s}|y; \theta_D\big)\Big)\Big], \quad (1)$$

where $\mathbf{y}$ explicitly depends on the encoding block $\mathbf{x} = f(s; \theta_E)$, and thus, on the parameters θ_E, while the decoder $g(\mathbf{y}; \theta_D)$ calculates the probability mass function $p_{\hat{S}|Y}(\hat{s}|y; \theta_D)$. The performance of the autoencoder-based system is typically measured in terms of bit error rate (BER) or block error rate (BLER)

$$P_e = P\big[\hat{s} \neq s\big]. \quad (2)$$

The autoencoder ability to learn joint coding and modulation schemes [2], [4] for any type of channel (even for

those without a known model) and for any type of non-linear effects (e.g., from amplifiers and clipping) [7] demonstrates the potentiality and flexibility of the approach.

However, the cross-entropy loss function does not guarantee any optimality in the code design and it is often prone to overfitting issues [22], [24]. In addition and most importantly, optimal system performance is measured in terms of achievable rates, thus, in terms of mutual information $I(X;Y)$ between the transmitted $\mathbf{x}$ and the received signals $\mathbf{y}$, defined as

$$I(X;Y) = \mathbb{E}_{(x,y)\sim p_{XY}(x,y)}\left[\log \frac{p_{XY}(x,y)}{p_X(x)p_Y(y)}\right], \tag{3}$$

or alternatively in terms of Kullback-Leibler divergence

$$I(X;Y) = D_{\mathrm{KL}}(p_{XY}||p_X \cdot p_Y). \tag{4}$$

In communications, the trade-off between the rate of transmission and reliability is expressed in terms of channel capacity. For a memory-less channel, the capacity is defined as

$$C = \max_{p_X(x)} I(X;Y), \tag{5}$$

where $p_X(x)$ is the input signal probability density function. Finding the channel capacity C is at least as complicated as evaluating the mutual information. As a direct consequence, building capacity-approaching codes is a formidable task.

Given a certain power constraint and rate R, the autoencoder-based system, that is trained to minimize the cross-entropy loss function, is able, if large enough, to automatically build nearly zero-error codes. Nevertheless, there exists a code at the same rate that exhibits better performance and may even exist a higher rate error-free code (asymptotically). Therefore, the autoencoder does not provide a capacity-achieving code. In other words, conventional autoencoding approaches, through cross-entropy minimization, allow to obtain excellent decoding schemes. Nevertheless, no guarantee to find an optimal encoding scheme is given, especially in deep NNs where problems such as vanishing and exploding gradients occur [25]. Hence, the starting point to design capacity-approaching codes is to redefine the loss function used by the autoencoder. In detail, we propose to include the mutual information quantity as a regularization term. The proposed loss function reads as follows

$$\hat{\mathcal{L}}(\theta_E,\theta_D) = \mathbb{E}_{(x,y)\sim p_{XY}(x,y)}\left[-\log\left(p_{\hat{S}|Y}(\hat{s}|y;\theta_D)\right)\right] - \beta I(X;Y). \tag{6}$$

The loss function in (6) forces the autoencoder to jointly modify the network parameters (θ_E,θ_D). The decoder reconstructs the original message s with lowest possible error probability P_e, while the encoder finds the optimal input signal distribution $p_X(x)$ which maximizes $I(X;Y)$, for a given rate R and code length n and for a certain power constraint. We will denote such type of trained autoencoder as *rate-driven* autoencoder. It should be noted that such a NN architecture does not necessarily provide an optimal code capacity-wise, since we set a target rate which does not correspond to channel capacity. To solve this second objective, in Section IV we will describe a theoretical methodology leading to a new scheme that we name *capacity-driven* autoencoder.

To compute the mutual information $I(X;Y)$, we can exploit recent results such as MINE [23], as discussed below.

B. MUTUAL INFORMATION ESTIMATION

The mutual information between two random variables, X and Y, is a fundamental quantity in statistics and information theory. It measures the amount of information obtained about X by observing Y. The difficulty in computing $I(X;Y)$ resides in its dependence on the joint probability density function $p_{XY}(x,y)$, which is usually unknown. Common approaches to estimate the mutual information rely on binning, density and kernel estimation [26], k-nearest neighbours [27], f-divergence functionals [28], and variational lower bounds [29].

Recently, the MINE estimator [23] proposed a NN-based method to estimate the mutual information (see (4)). MINE is based on the Donsker-Varadhan dual representation [30] of the Kullback-Leibler divergence, in particular

$$D_{\mathrm{KL}}(p||q) = \sup_{T:\Omega\to\mathbb{R}} \mathbb{E}_{x\sim p(x)}[T(x)] - \log\left(\mathbb{E}_{x\sim q(x)}\left[e^{T(x)}\right]\right), \tag{7}$$

where the supremum is taken over all functions T such that the expectations are finite. Indeed, by parameterizing a family of functions $T_\theta : \mathcal{X}\times\mathcal{Y}\to\mathbb{R}$ with a deep NN with parameters $\theta\in\Theta$, the following bound [23] holds

$$I(X;Y) \geq I_\theta(X;Y), \tag{8}$$

where $I_\theta(X;Y)$ is the neural information measure [23] defined as

$$I_\theta(X;Y) = \sup_{\theta\in\Theta} \mathbb{E}_{(x,y)\sim p_{XY}(x,y)}\left[T_\theta(x,y)\right] - \log\left(\mathbb{E}_{(x,y)\sim p_X(x)p_Y(y)}\left[e^{T_\theta(x,y)}\right]\right). \tag{9}$$

The neural information $I_\theta(X;Y)$ in (9) can be maximized using back-propagation and gradient ascent, leading to a tighter bound in (8). To avoid biased gradients, the authors in [23] suggested to replace the expectation in the denominator (coming from the derivative of the logarithm) with an exponential moving average. The consistency property of MINE guarantees the convergence of the estimator to the true mutual information value.

Estimating the mutual information $I(X;Y)$ is not enough to build capacity-approaching codes for a generic channel. A maximization over all possible input distribution $p_X(x)$ is also required. Therefore, to learn an optimal scheme, at each iteration the encoder needs both the cross-entropy gradient for the decoding phase and the mutual information gradient, from MINE, for the optimal input signal distribution. The

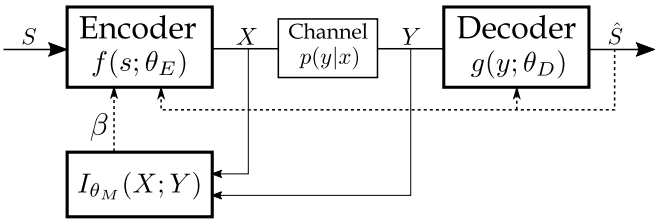


FIGURE 1. Rate-driven autoencoder with the mutual information estimator block. The channel samples and the decoded message are used during the training process. The former allows to built an optimal encoding scheme exploiting the mutual information block, the latter, instead, allows to measure the decoding error through the cross-entropy loss function.

proposed loss function in (6) (see also Fig. 1) shows such double role. In this way, the autoencoder trained with the new loss function intrinsically designs codes for which the mutual information $I(X;Y)$ is known and maximal by construction, under the aforementioned constraints of power, rate R and code-length n.

The rationale behind the proposed method is formally discussed in the next section.

III. MUTUAL INFORMATION REGULARIZATION

The autoencoder is a classifier network since the decoding block performs a classification task while attempting to recover the sequence of transmitted bits. The training of large autoencoder networks may suffer from known NN degradation issues such as overfitting and vanishing gradients. In the following, we discuss in detail both problems motivating why the addition of the mutual information regularization term to the cross-entropy loss function, as in (6), and the adoption of the label smoothing technique, offer better performance. Indeed, the cross-entropy can be minimized even for random labels as shown in [31], leading to several overfitting issues. To combat overconfident predictions, the mutual information term plays the role of both an entropy penalty and an information bottleneck regularizer. Furthermore, its gradient directly influences the encoder parameter, mitigating the vanishing gradient problem.

A. OVERFITTING MITIGATION

The objective of the autoencoder is twofold: the encoder searches for a latent representation of the transmitted signal that is robust to channel and noise distortions while the decoder learns how to successfully recover the distorted transmitted signal. The latter, in particular, attempts to find a signal representation that provides robust performance in a more general domain than the one given by the observed/training dataset. This is rendered possible by the stochastic description of the channel. Nevertheless, fitting the empirical distribution often leads to overfitting, an undesired effect that appears when the network places most of the probability mass on a subset of the possible output classes. In other words, the decoder block produces a peaky leptokurtic conditional output distribution $p_{\hat{S}|Y}(\hat{s}|y;\theta_D)$ which may improve the training loss at the expenses of the generalization ability of the network. To avoid leptokurtic distributions, a strategy relies on entropy regularization, which penalizes confident output distributions by encouraging more entropic ones. In [32], the authors introduced the confidence penalty regularization term as an entropy term in the supervised learning setting:

$$H\Big(p_{\hat{S}|Y}(\hat{s}|y;\theta_D)\Big) = \sum_i p_{\hat{S}|Y}(\hat{s}_i|y;\theta_D)\log\Big(p_{\hat{S}|Y}(\hat{s}_i|y;\theta_D)\Big), \tag{10}$$

where y and $\hat{s}$ are the input and output of the classifier, respectively. However, in the autoencoder set up, y is the output of the channel layer and is dependent itself to the training process. Indeed, the overfitting issue may occur while producing a peaky latent space distribution that does not maximize the entropy of X and consequently, taking in consideration the channel layers, of Y. To combat such leptokurtic behavior in the encoding distribution while considering the channel dependence in the output, we propose to penalize it by exploiting the mutual information between the transmitted and received signals:

$$I(X;Y) = H(X) + H(Y) - H(X,Y), \tag{11}$$

where $H(X,Y)$ is the joint entropy measure. The objective of this penalty regularization term is to promote more entropic signal input-output distributions while reducing the joint entropy, thus, placing the joint probability mass in peaky clusters.

The mutual information penalty regularizer directly influences the encoder network parameters during the learning phase. However, as mentioned before while discussing the confidence penalty in (10), peakiness may appear also in the target output distribution $p_{\hat{S}|Y}(\hat{s}|y;\theta_D)$. To further improve the decoder generalization, we propose also to adopt the label smoothing technique [33], a type of entropy regularizer particularly effective when dealing with hard target distributions such as the one-hot vectors of the autoencoder scheme. The idea of label smoothing is to replace the target output distribution (also referred to as label distribution) $p_{\hat{S}|Y}(\hat{s}|y) = \delta_{\hat{S},S}$ with a mixture of the original ground-truth and the chosen distribution $u(\hat{s})$:

$$\hat{p}_{\hat{S}|Y}(\hat{s}|y) = (1-\epsilon)\delta_{\hat{S},S} + \epsilon u(\hat{s}), \tag{12}$$

where in this paper $u(\hat{s})$ is chosen to be as the uniform distribution $u(\hat{s}) = 1/M$, and ϵ is a positive number. Label smoothing forces more entropic output messages $\hat{S}$ distributions, improving the decoder generalization ability. The cross-entropy loss function using label smoothing reads as follows

$$\hat{\mathcal{L}}(\theta_E,\theta_D) = \mathbb{E}_{(s,y)\sim\hat{p}_{\hat{S}|Y}(\hat{s}|y)\cdot p_Y(y|x)}\Big[-\log\Big(p_{\hat{S}|Y}(\hat{s}|y;\theta_D)\Big)\Big]. \tag{13}$$

In Section V-B, we demonstrate how the combination of both the mutual information regularizer and the label

smoothing technique leads to more entropic predicted output distributions.

Another important benefit of the proposed mutual information penalty term consists in the mitigation of the vanishing gradient problem.

B. VANISHING GRADIENT MITIGATION

It is well known that adding layers with certain activation functions to the NN may result in small gradients of the loss function, inhibiting an effective update of the parameters of the first layers. In the case of large autoencoder networks, the encoder block may suffer from such vanishing gradient problem, especially if the decoder block is itself a deep NN and the channel block is modeled by a multi-layer channel generator obtained via a GAN based training scheme [9]. Indeed, the generator in a GAN framework is typically a deep network that implicitly estimates the conditional channel distribution $p_Y(y|x)$. For this reason, the back-propagated gradient responsible for the update of the encoder parameters θ_E may be negligible when only cross-entropy is used as training cost function. Hence, a regularizer whose gradient influences directly the parameters of the encoder block can better guide the network in identifying the latent space. In particular, if the regularization term is the mutual information, then the encoder also attempts to find the optimal channel input distribution, tackling the achievement of channel capacity, which is exactly the aim of optimal communication schemes.

In detail, given the loss function in (6)

$$\hat{\mathcal{L}}(\theta_E, \theta_D) = \mathcal{L}(\theta_E, \theta_D) - \beta I_{\theta_E}(X; Y), \tag{14}$$

and given a probabilistic channel generative model $y = h(x; \theta_h)$, at each training iteration the gradient back-propagated from the decoder network $g(y; \theta_D)$ to the encoder network $f(s; \theta_E)$ can be computed as

$$\nabla_{\theta_E}\hat{\mathcal{L}}(\theta_E, \theta_D) = \nabla_{\theta_E}\mathcal{L}(\theta_E, \theta_D) - \beta\nabla_{\theta_E} I_{\theta_E}(X; Y) \tag{15}$$

and using the chain rule

$$\begin{aligned}\nabla_{\theta_E}\hat{\mathcal{L}}(\theta_E, \theta_D) = &\frac{\partial \mathcal{L}}{\partial g}\cdot\frac{\partial g}{\partial h}\cdot\frac{\partial h}{\partial f}\cdot\nabla_{\theta_E} f(s;\theta_E)\\ &- \beta\frac{\partial I}{\partial h}\cdot\frac{\partial h}{\partial f}\cdot\nabla_{\theta_E} f(s;\theta_E)\\ &- \beta\frac{\partial I}{\partial f}\cdot\nabla_{\theta_E} f(s;\theta_E).\end{aligned} \tag{16}$$

From the relations in (16), it is clear that the regularization term with strength β can in principle generate a more energetic gradient. Indeed, the first term in the RHS comes from the cross-entropy gradient and it depends on the decoder $(\partial g/\partial h)$, while the second and third terms in the RHS contain the gradient of the mutual information regularizer and do not depend on the decoder.

C. INFORMATION BOTTLENECK METHOD

In [22], the authors proved how a deep NN can just memorize the dataset (in its weights) to minimize the cross-entropy, yielding to poor generalization. Hence, the authors proposed an information bottleneck (IB) regularization term to prevent overfitting, similarly to the IB Lagrangian, originally presented in [20]. Indeed, the IB method optimally compresses the input random variable by eliminating the irrelevant features which do not contribute to the prediction of the output random variable.

From an autoencoder-based communication systems point of view, let $S \to X \to Y$ be a prediction Markov chain, where S represents the message to be sent, X the compressed symbols and Y the received symbols. The IB method solves

$$\mathcal{L}(p(x|s)) = I(S; X) - \beta I(X; Y), \tag{17}$$

where the positive Lagrange multiplier β plays as a trade-off parameter between the complexity of the encoding scheme (rate) and the amount of relevant information preserved in it.

The communication chain adds an extra Markov chain constraint $Y \to \hat{S}$, where $\hat{S}$ represents the decoded message. Therefore, in order to deal with the full autoencoder chain, we decide to substitute the first term of the RHS in (17) with the cross-entropy loss function, as presented in (6). However, the Lagrange multiplier (or regularization parameter in ML terms) operates now as a trade-off parameter between the complexity to reconstruct the original message and the amount of information preserved in its compressed version.

D. CROSS-ENTROPY DECOMPOSITION

To further motivate the choice for the new loss function with the mutual information as the regularization term, let us consider the following decomposition of the cross-entropy loss function.

Lemma 1 (See [12]): Let $s \in \mathcal{M}$ be the transmitted message and let (x, y) be samples drawn from the joint distribution $p_{XY}(x, y)$. If $x = f(s; \theta_E)$ is an invertible function representing the encoder and if $p_{\hat{S}|Y}(\hat{s}|y; \theta_D) = g(y; \theta_D)$ is the decoder block, then the cross-entropy function $\mathcal{L}(\theta_E, \theta_D)$ admits the following decomposition

$$\begin{aligned}\mathcal{L}(\theta_E, \theta_D) = H(S) - I_{\theta_E}(X; Y) +\\ + \mathbb{E}_{y\sim p_Y(y)}\Big[D_{\mathrm{KL}}\Big(p_{X|Y}(x|y)||p_{\hat{S}|Y}(x|y;\theta_D)\Big)\Big].\end{aligned} \tag{18}$$

A complete proof of Lemma 1, slightly different from the one in [12], is reported in the Appendix.

The cross-entropy decomposition in Lemma 1 can be read in the following way: the first two terms are responsible for the conditional entropy of the received symbols. In the particular case of a uniform source, only the mutual information between the transmitted and received symbols is controlled by the encoding function during the training process. On the contrary, the last term measures the error in computing the divergence between the true posterior distribution and the

Algorithm 1 Capacity Learning With Capacity-Driven Autoencoders

Inputs:
L SNR increasing values, ϵ threshold.
Initialize:
$R_0 = k_0/n_0$ initial rate, $i = 0, j = 0$.
for $l = 1$ to L **do**
  Train $AE^{(0)}(k_0, n_0)$;
  Compute $I^{(0)}_{\theta_M}(X;Y)$;
  while $\Delta > \epsilon$ **do**
    $k_{i+1} = (R_i \cdot n_j) + 1$;
    $R_{i+1} = k_{i+1}/n_j$;
    Train $AE^{(i+1)}(k_{i+1}, n_j)$;
    if R_{i+1} is not achievable **then**
      $n_{j+1} = n_j + 1$;
      $j = j + 1$;
    else
      Compute $I^{(i+1)}_{\theta_M}(X;Y)$;
      Evaluate $\Delta = \left| \frac{I^{(i+1)}_{\theta_M}(X;Y) - I^{(i)}_{\theta_M}(X;Y)}{I^{(i)}_{\theta_M}(X;Y)} \right|$;
    $i = i + 1$;
  $C_l = I^{(i)}_{\theta_M}(X;Y)$ estimated capacity.

decoder-approximated one. As discussed before, the network could minimize the cross-entropy just by minimizing the KL-divergence, concentrating itself only on the label information (decoding) rather than on the symbol distribution (coding). To avoid this, we propose (6) where the parameter β helps in balancing the two different contributions as follows

$$\mathcal{L}(\theta_E, \theta_M, \theta_D) = H(S) - I_{\theta_E}(X;Y) - \beta I_{\theta_E,\theta_M}(X;Y) + \\ + \mathbb{E}_{y\sim p_Y(y)}\Big[D_{\mathrm{KL}}\Big(p_{X|Y}(x|y)||p_{\hat{S}|Y}(x|y;\theta_D)\Big)\Big]. \tag{19}$$

Moreover, if the mutual information estimator is consistent, (19) is equal to

$$\mathcal{L}(\theta_E, \theta_D) = H(S) - (\beta+1) I_{\theta_E}(X;Y) + \\ + \mathbb{E}_{y\sim p_Y(y)}\Big[D_{\mathrm{KL}}\Big(p_{X|Y}(x|y)||p_{\hat{S}|Y}(x|y;\theta_D)\Big)\Big]. \tag{20}$$

It is immediate to notice that for $\beta < -1$, the network gets in conflict since it would try to minimize both the mutual information and the KL-divergence. Therefore, optimal values for β lie on the semi-line $\beta > -1$.

IV. CAPACITY-DRIVEN AUTOENCODER

Interestingly, the mutual information block can be exploited to obtain an estimate of the channel capacity. The autoencoder-based system is subject to a power constraint coming from the transmitter hardware and it generally works at a fixed rate R and channel uses n. For R and n fixed, the scheme discussed in Fig. 1 optimally designs the coded signal distribution and provides an estimate $I_{\theta_M}(X;Y)$ of the mutual information $I(X;Y)$ which approaches R. However, a question remains still open: *is the achieved rate with the designed code actually channel capacity*?

To find the channel capacity C and determine the optimal signal distribution $p_X(x)$, a broader search on the coding rate needs to be conducted, relaxing both the constraints on R and n. The flexibility on R and n requires to use different autoencoders. In the following, we denote with $AE(k,n)$ a rate-driven autoencoder-based system that transmits n complex symbols at a rate $R = k/n$, where $k = \log_2(M)$ and M is the number of possible messages. The proposed methodology can be segmented in two phases:

1) Training of a rate-driven autoencoder $AE(k,n)$ for a fixed coding rate R and channel uses n, enabled via the loss function in (6);
2) Adaptation of the coding rate R to build capacity approaching codes $\mathbf{x} \sim p_X(x)$ and consequently find the channel capacity C.

We remark that the capacity C is the maximum data rate R that can be conveyed through the channel at an arbitrarily small error probability. Therefore, the proposed algorithm makes an initial guess rate R_0 and smoothly increases it by playing on both k and n.

The basic idea is to iteratively train at the i-th iteration a pair of rate-driven autoencoders $AE^{(i)}(k_i, n_j), AE^{(i+1)}(k_{i+1}, n_j)$ and evaluate both the mutual informations $I^{i}_{\theta_M}(X;Y), I^{(i+1)}_{\theta_M}(X;Y)$, at a fixed power constraint. The first autoencoder works at a rate $R_i = k_i/n_j$, while the second one at $R_{i+1} = k_{i+1}/n_j$, with $R_{i+1} > R_i$. If the ratio

$$\left| \frac{I^{(i+1)}_{\theta_M}(X;Y) - I^{(i)}_{\theta_M}(X;Y)}{I^{(i)}_{\theta_M}(X;Y)} \right| < \epsilon, \tag{21}$$

where ϵ is an input positive parameter, the code is reaching the capacity limit for the fixed power. If the rate R_{i+1} is not achievable (it does not exist a nearly error-free decoding scheme), a longer code n_{j+1} is required. The algorithm in Tab. 1 describes the pseudocode that implements the channel capacity estimation and capacity-approaching code using as a building block the rate-driven autoencoder.

A. IMPORTANT REMARKS

The proposed capacity-driven autoencoder offers a constructive learning methodology to design a coding scheme that approaches capacity and to know what such a capacity is, even for channels that are unknown or for which a closed form expression for capacity does not exist. Indeed, training involves numerical procedures which may introduce some challenges. Firstly, the autoencoder is a NN and it is well known that its performance depends on the training procedure, architecture design and hyper-parameters tuning. Secondly, the MINE block converges to the true mutual information mostly for a low number of samples. In practice, when n is larger than 4, the estimation often produces

unreliable results, therefore, a further investigation on stable numerical estimators via NNs needs to be conducted. Lastly, the autoencoder fails to scale with high code dimension. Indeed, for large values of n, the network could get stuck in local minima or, in the worst scenario, could not provide nearly zero-error codes due to limited resources or not large enough networks. The proposed approach transcends such limitations, although they have to be taken into account in the implementation phase. In addition, the work follows the direction of explainable machine learning, in which the learning process is motivated by an information-theoretic approach. Possible improvements are in defining an even tighter bound in (8) and in adopting different network structures (convolutional or recurrent NNs).

It should be noted that the approach works also for non linear channels where optimal codes have to be designed under an average power constraint and not for a given operating SNR which is appropriate for linear channels with additive noise.

V. NUMERICAL RESULTS

In this section, we present results obtained with the rate-driven autoencoders. They demonstrate an improvement in the decoding schemes (measured in terms of BLER) and show the achieved rates with respect to capacity in channels for which a closed form capacity formulation is known, such as the AWGN channel, and unknown, such as additive uniform noise and Rayleigh fading ones.

The following schemes consider an average power constraint at the transmitter side $\mathbb{E}[|\mathbf{x}|^2] = 1$, implemented through a batch-normalization layer. Training of the end-to-end autoencoder is performed w.r.t. to the loss function in (6), implemented via a double minimization process since also the MINE block needs to be trained:

$$\min_{\theta_E,\theta_D}\min_{\theta_M}\mathbb{E}_{(x,y)\sim p_{XY}(x,y)}\Big[-\log\Big(p_{\hat{S}|Y}\big(\hat{s}|y;\theta_D\big)\Big)\Big]-\beta I_{\theta_M}(X;Y). \tag{22}$$

Furthermore, the training procedure was conducted with the same number of iterations for different values of the regularization parameter β, at a fixed value of $E_b/N_0 = 7$ dB. Unless otherwise specified, we included label smoothing with $\epsilon = 0.2$ during the autoencoders training process. We used Keras with TensorFlow [34] as backend to implement the proposed rate-driven autoencoder. The code has been tested on a Windows-based operating system provided with Python 3.6, TensorFlow 1.13.1, Intel core i7-3820 CPU. To allow reproducible results and for clarity, the code is rendered publicly available.[1]

A. CODING-DECODING CAPABILITY

As first experiment, we consider a rate-driven autoencoder AE(4, 2) with rate $R = 2$. The advantage of using a mutual information estimator block during the end-to-end training phase is expected to be more pronounced from $n > 1$. To demonstrate the effective influence on the performance of the mutual information term controlled by β in (22), we investigate 4 different representative values of the regularization parameter for both cases with and without label smoothing. Fig. 2 illustrates the obtained BLER after the same number of training iterations when the regularization term is added in the cost function. We notice that the lowest BLER is achieved for $\beta = 0.2$, therefore as expected, the mutual information contributes in finding better encoding schemes. Despite the small gain, the result highlights that better BLER can be obtained using the same number of iterations. As shown in (20), negative values of β tend to force the network to just memorize the dataset, while large positive values create an unbalance. We remark that $\beta = 0$ and $\epsilon = 0$ corresponds to the classic autoencoder approach proposed in [2]. Fig. 2 also illustrates a slightly worse BLER when trained with label smoothing. However, we highlight the fact that label smoothing renders the network more robust to overfitting which does not necessarily reflect into higher accuracy of the model [35]. To identify optimal values of β, a possible approach can try to find the value of β for which the two gradients (cross-entropy and mutual information) are equal in magnitude. In the following, we assume $\beta = 0.2$.

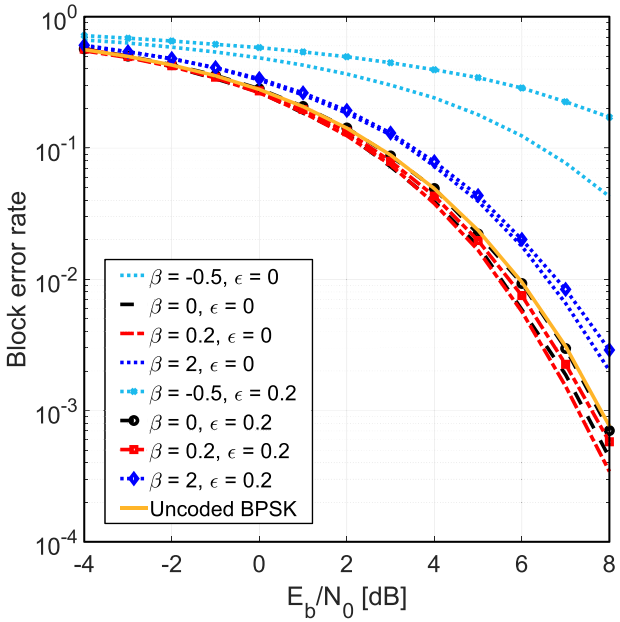


FIGURE 2. BLER of the rate-driven autoencoder AE(4, 2) for different values of the regularization β and label smoothing parameter ϵ, for an average power constraint, $k = 4$ and $n = 2$.

To assess the methodology even with higher dimension M of the input alphabet, we illustrate the optimal constellation schemes when the number of possible messages is $M = 16$ and $M = 32$. Moreover, two cases are studied, when we transmit one complex symbol ($n = 1$) and two dependent complex symbols ($n = 2$) over the channel. Fig. 3a and Fig. 3b show the learned hexagonal spiral grid constellations when only one symbol is transmitted for an alphabet

1. https://github.com/tonellolab/capacity-approaching-autoencoders

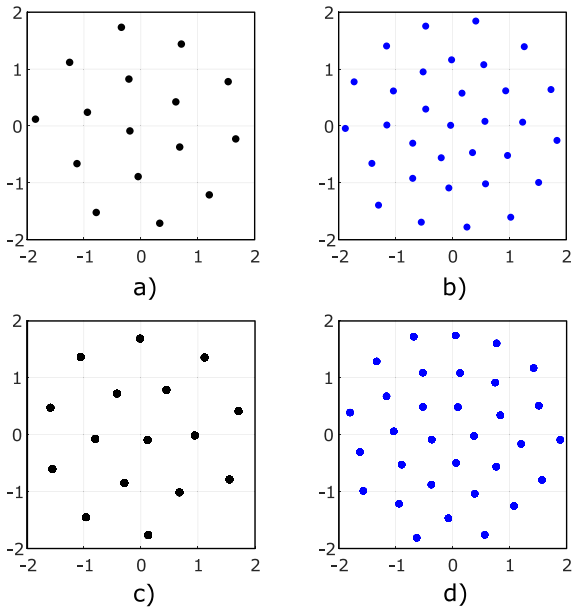


FIGURE 3. Constellation designed by the encoder during the end-to-end training with $\beta = 0.2$ and $\epsilon = 0.2$ and parameters (k, n): a) (4, 1), b) (5, 1), c) 2 dimensional t-SNE of AE(4, 2) and d) 2 dimensional t-SNE of AE(5, 2).

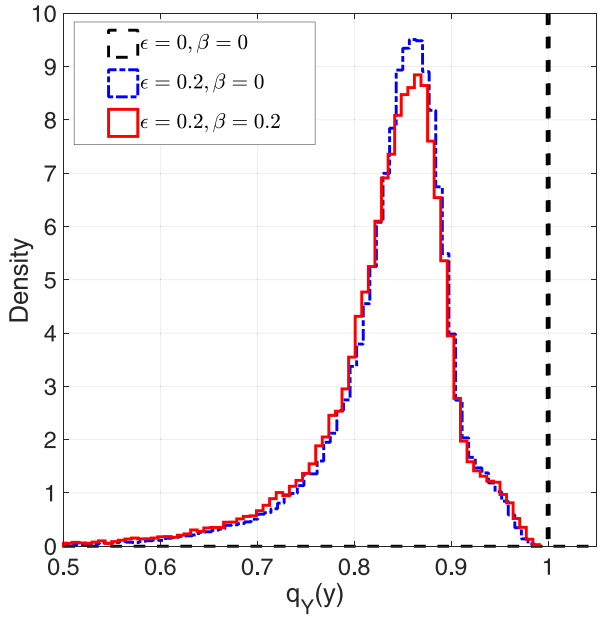


FIGURE 4. Distribution of the maximal value of the softmax layer of AE(2, 1) for three different scenarios: only cross-entropy training, cross-entropy with label smoothing, cross-entropy with mutual information and label smoothing.

dimension of $M = 16$ and $M = 32$. Fig. 3c and Fig. 3d show, instead, an optimal projection of the coded signals in a 2D space through the learned two-dimensional t-distributed stochastic neighbor embedding (t-SNE) [36]. We notice that the two pairs of constellations are similar, and therefore, even for codes of length $n = 2$, the mutual information pushes the autoencoder to learn the optimal signal constellation. As mentioned in the remarks section, the advantage is expected to be more pronounced the larger n is. However, there is a practical limitation to obtain constructive results for higher values of n since current mutual information estimators, including MINE, are not stable. This can be solved with novel NN architectures and training algorithms, which is one of the challenges the ML community is currently facing.

B. ENTROPY REGULARIZATION

In this section, we evaluate the advantage given by the mutual information and label smoothing regularization techniques over the solely cross-entropy training method. We analyze the ability to generalize its applicability and the ability to mitigate the overfitting problem of the trained autoencoder. In particular, we propose to study the distribution of the softmax output layer in the simple 4-QAM autoencoder AE(2, 1). The softmax output layer attempts to predict the output distribution $p_{\hat{S}|Y}(\hat{s}|y;\theta_D)$ and therefore provides insights about the generalization status of the network, where a smoother platikurtik output distribution is often synonym of generalization beyond the observed data [32] while a peaky output distribution usually stands for poor generalization.

For visualization purposes, we report the analysis of the maximal output distribution $q_Y(y) = \max_{\hat{S}} p_{\hat{S}|Y}(\hat{s}|y)$. We consider, as an example, the behavior of the probability density function q for a value of $E_b/N_0 = 7$ dB in three different training scenarios: only cross-entropy ($\epsilon = 0, \beta = 0$), cross-entropy with label smoothing ($\epsilon = 0.2, \beta = 0$), cross-entropy with mutual information and label smoothing regularizers ($\epsilon = 0.2, \beta = 0.2$). From Fig. 4, we can observe that label smoothing forces a less confident prediction compared to the common cross-entropy. Interestingly, the entropic effect of the mutual information regularizer, which acts primarily on the encoder block, is present also at the autoencoder output distribution even for a simple autoencoding scheme as the 4-QAM. Indeed, for 100 bins, the estimated entropy of $q_Y(y)$ without mutual information regularization is around 3.66 Nat, while the entropy of the output with mutual information is around 3.72 Nat.

We also report the entropy of $q_Y(y)$ varying the energy per bit to noise ratio E_b/N_0 for both cases with and without mutual information regularization. In both cases, label smoothing for a more stable estimation is used. The network should be capable to predict correct outputs outside the training region, thus, a high entropy is desired. As depicted in Fig. 5, the entropy of the maximal output when regularized with both mutual information and label smoothing is greater than the entropy of the maximal output when only label smoothing is used, for positive values of E_b/N_0. This is coherent with the intuition provided in Section III-A, i.e., the mutual information term helps in preventing overfitting. An unexpected result comes from the evolution of the estimated entropy varying the energy per bit to noise ratio: the entropy appears to have a local maximum around 0 dB when the noise power is comparable to the signal's one.

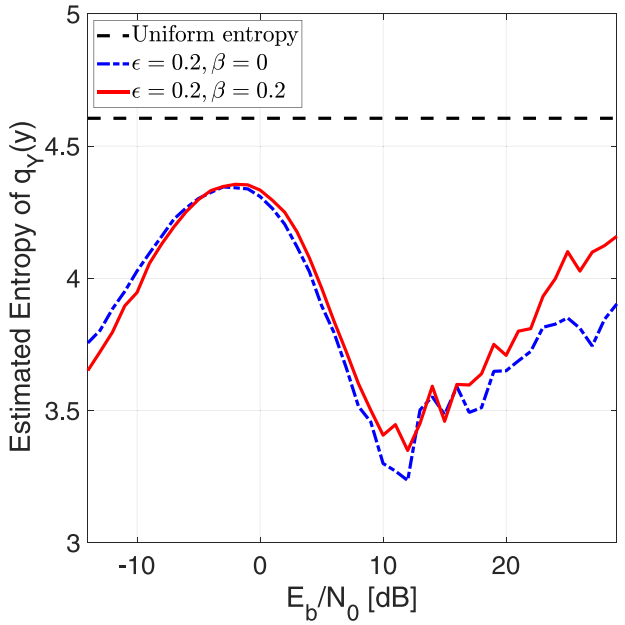


FIGURE 5. Estimated entropy of $q_Y(y)$ for different values of the energy per bit to noise ratio in the 4-QAM Gaussian autoencoder AE(2, 1).

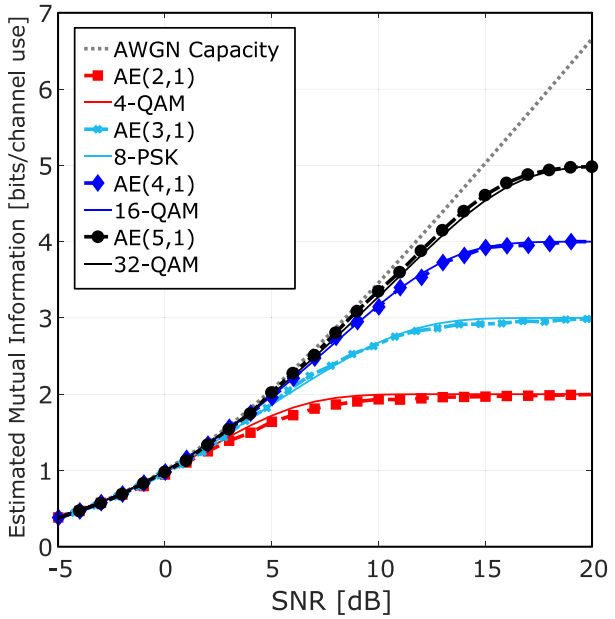


FIGURE 6. Estimated mutual information achieved with $\beta = 0.2$ and $\epsilon = 0.2$ for different dimension of the alphabet M with code length $n = 1$ over an AWGN channel.

Therein, the network is essentially uniformly guessing the predicted message, without any confidence. When the noise power decreases, the network confidently places probability masses following the density in Fig. 4. However, for values of $E_b/N_0 > 10$ dB, the entropies of the predictions $q_Y(y)$ start increasing again, and the autoencoder trained with mutual information regularization maintains a better generalization ability. The point of minimal entropy can be thought as a transition point from continuous (in amplitude) received signals, towards discrete valued signals. These results stimulate further analysis of entropy trends in future research work.

C. CAPACITY-APPROACHING CODES OVER DIFFERENT CHANNELS

The mutual information block inside the autoencoder can be exploited to design capacity-approaching codes, as discussed in Section IV. To show the potentiality of the method, we analyze the achieved rate, e.g., the mutual information, in three different scenarios. The first one considers the transmission over an AWGN channel, for which we know the exact closed form capacity. The second and third ones, instead, consider the transmission over an additive uniform noise channel and over a Rayleigh fading channel, for which we do not know the capacity in closed form. However, we expect the estimated mutual information to be a tight lower bound for the real channel capacity, especially at low SNRs.

1) AWGN CHANNEL

Let us consider a discrete memory-less channel with input-output relation given by (assuming complex signals)

$$Y_i = X_i + N_i, \tag{23}$$

where the noise samples $N_i \sim \mathcal{CN}(0, \sigma^2)$ are i.i.d. and independent of X_i. It is well known that with a power constraint on the input signal $\mathbb{E}[|X_i|^2] \leq P$, the channel capacity is achieved by $X_i \sim \mathcal{CN}(0, P)$ and is equal to

$$C = \log_2(1 + \text{SNR}) \ \ \text{[bits/channel use]}. \tag{24}$$

The rate-driven autoencoder attempts to maximize the mutual information during the training progress by modifying, at each iteration, the input distribution $p_X(x)$. Thus, given the input parameters, it produces optimal codes for which the estimation of the mutual information is provided by MINE. Fig. 6 illustrates the achieved and estimated mutual information when $\beta = 0.2$ and $\epsilon = 0.2$ for different values of the alphabet cardinality M. A comparison is finally made with established M-QAM schemes. We remark that for discrete-input signals with distribution $p_X(x)$, the mutual information is given by

$$I(X;Y) = \sum_x p_X(x) \cdot \mathbb{E}_{p_Y(y|x)}\left[\log \frac{p_Y(y|x)}{p_Y(y)}\right], \tag{25}$$

and in particular with uniformly distributed symbols (only geometric shaping), $p_X(x) = 1/M$. It is found that the autoencoder constructively provides a good estimate of mutual information. In addition, for the case $M = 32$, the conventional QAM signal constellation is not optimal, since the autoencoder AE(5, 1) performs geometric signal shaping and finds a constellation that can offer higher information rate as it is visible in the range $13 - 16$ dB. Lastly, if we code over two channel uses, i.e., $n = 2$, an improvement in mutual information can be attained. This is shown in Fig. 10 where a comparison between channels affected by AWGN, uniform noise and Rayleigh fading is reported.

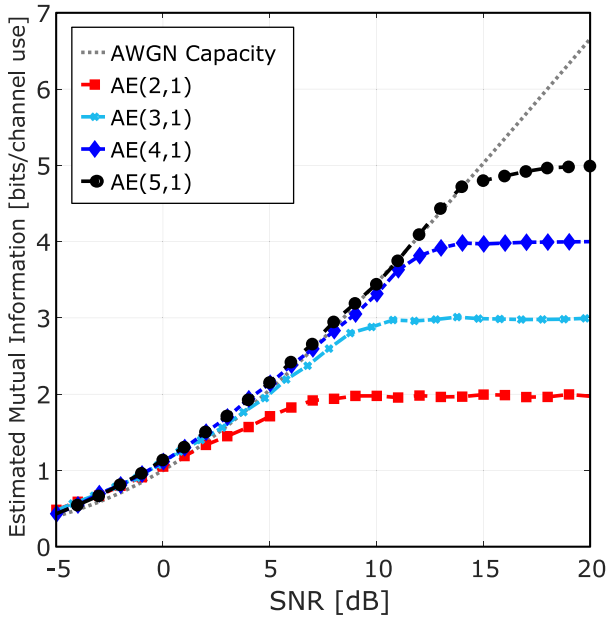


FIGURE 7. Estimated mutual information achieved with $\beta = 0.2$ and $\epsilon = 0.2$ for different dimension of the alphabet M with code $n = 1$ over an additive uniform noise channel.

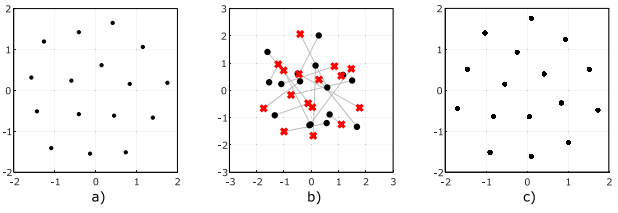


FIGURE 8. Constellation designed by the encoder during the end-to-end training with $\beta = 0.2$ and $\epsilon = 0.2$, uniform noise layer and parameters (k, n): a) (4, 1), b) pairs of transmitted coded symbols AE(4, 2), c) 2 dimensional t-SNE of AE(4, 2).

2) ADDITIVE UNIFORM NOISE CHANNEL

No closed form capacity expression is known when the noise N has uniform distribution $N \sim \mathcal{U}(-\frac{\Delta}{2}, \frac{\Delta}{2})$ under an average power constraint. However, Shannon proved that the AWGN capacity is the lowest among all additive noise channels of the form (23). Consistently, as depicted in Fig. 7, it is rather interesting to notice that the estimated mutual information for the uniform noise channel is higher than the AWGN capacity for low SNRs until it saturates to the coding rate R. Moreover, differently from the AWGN coding signal set, Fig. 10b also considers the complex signals generated by the encoder over two channel uses. As expected, the mutual information achieved by the code produced with the autoencoder AE(4, 2) is higher than with AE(2, 1), consistently with the idea that $n > 1$ introduces a temporal dependence in the code allowing to improve the decoding phase. In addition, Fig. 8a illustrates the constellation produced by the rate-driven autoencoders AE(4, 1) in the uniform noise case. Fig. 8b shows how the transmitted coded symbols (transmitted complex coefficients) vary for different channel uses while Fig. 8c, instead, displays the learned two-dimensional t-SNE constellation of the code produced by the AE(4, 2).

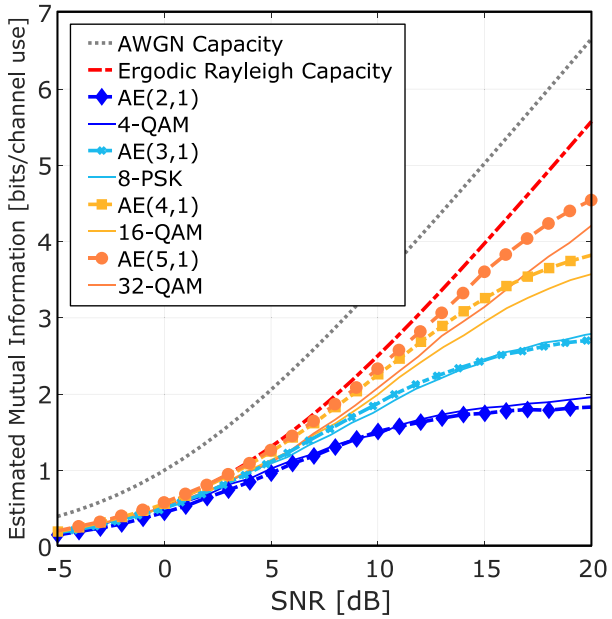


FIGURE 9. Estimated mutual information achieved with $\beta = 0.2$ and $\epsilon = 0.2$ for different dimension of the alphabet M with code length $n = 1$ over a Rayleigh channel.

3) RAYLEIGH FADING CHANNEL

As final experiment, we introduce fading in the communication channel, in particular we consider a Rayleigh fading channel of the form

$$Y_i = h_i X_i + N_i, \tag{26}$$

where $N_i \sim \mathcal{CN}(0, \sigma^2)$ and h_i is a random variable whose amplitude α belongs to the Rayleigh distribution $p_R(r)$ and is independent of the signal and noise. The ergodic capacity is given by

$$C = \mathbb{E}_{\alpha \sim p_R(r)}\Big[\log_2(1 + \alpha^2 \cdot \text{SNR})\Big]. \tag{27}$$

Fig. 9 shows the estimated mutual information attained by the autoencoder over a Rayleigh channel with several alphabet dimensions M and compares it with the conventional M-QAM schemes. In all the cases, the achieved mutual information is upper bounded by the ergodic Rayleigh capacity. Similarly to the uniform case, it is curious to notice that the achieved information rate with the rate-driven autoencoder is in some cases higher than the one obtained with the M-QAM schemes. In particular, with $M = 32$ the AE(5, 1) exceeds by 0.5 bit/channel uses at SNR $= 15$ dB the 32-QAM scheme. Lastly, Fig. 10c highlights the advantage of coding over two channel uses, especially in the range $5 - 15$ dB.

VI. CONCLUSION

This paper has firstly discussed the autoencoder-based communication system, highlighting the limits of the current

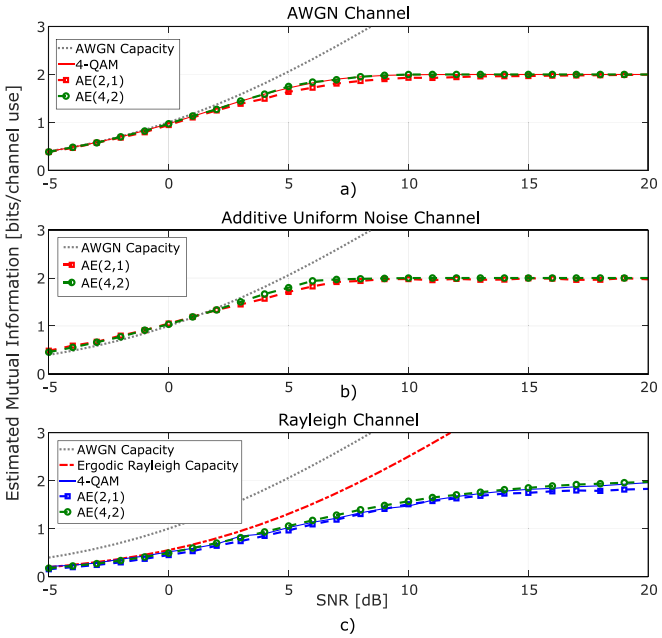


FIGURE 10. Estimated mutual information achieved with $\beta = 0.2$, $\epsilon = 0.2$ and rate $R = 2$ with code length $n = 1, 2$ over a) AWGN channel, b) Additive uniform noise channel, c) Rayleigh channel.

cross-entropy loss function used for the training process. A regularization term that accounts for the mutual information between the transmitted and the received signals has been introduced to design optimal coding-decoding schemes for fixed rate R, code-length n and given power constraint. The rationale behind the mutual information choice has been motivated exploiting the entropy regularization approach, the information bottleneck principle and the fundamental concept of channel capacity. In addition, an adaptation of the coding rate R allowed us to build a capacity learning algorithm enabled by the novel loss function in a scheme named capacity-driven autoencoder. Remarkably, the presented methodology does not make use of any theoretical a-priori knowledge of the communication channel and therefore opens the door to several future studies on intractable channel models, an example of which is the power line communication channel.

APPENDIX

We report a proof of the Lemma 1. Lemma 1 was stated in [12], but herein the proof is complete and slightly different.

Lemma 1 (See [12]): Let $s \in \mathcal{M}$ be the transmitted message and let (x, y) be samples drawn from the joint distribution $p_{XY}(x, y)$. If $x = f(s; \theta_E)$ is an invertible function representing the encoder and if $p_{\hat{S}|Y}(\hat{s}|y; \theta_D) = g(y; \theta_D)$ is the decoder block, then the cross-entropy function $\mathcal{L}(\theta_E, \theta_D)$ admits the following decomposition

$$\mathcal{L}(\theta_E, \theta_D) = H(S) - I_{\theta_E}(X; Y) + \\ + \mathbb{E}_{y\sim p_Y(y)}\Big[D_{\mathrm{KL}}\Big(p_{X|Y}(x|y)||p_{\hat{S}|Y}(x|y; \theta_D)\Big)\Big].$$

Proof: The cross-entropy loss function can be rewritten as follows

$$\begin{aligned}\mathcal{L}(\theta_E, \theta_D) &= \mathbb{E}_{(x,y)\sim p_{XY}(x,y)}\Big[-\log\Big(p_{\hat{S}|Y}(\hat{s}|y; \theta_D)\Big)\Big] \\ &= -\sum_{x,y} p_{XY}(x, y)\log\Big(p_{\hat{S}|Y}(\hat{s}|y; \theta_D)\Big) \\ &= -\sum_{x,y} p_{XY}(x, y)\log(p_X(x)) + \\ &+ \sum_{x,y} p_{XY}(x, y)\log\left(\frac{p_X(x)}{p_{\hat{S}|Y}(\hat{s}|y; \theta_D)}\right)\end{aligned}$$

Using the encoder hypothesis, the first term in the last expression corresponds to the source entropy $H(S)$. Therefore,

$$\begin{aligned}\mathcal{L}(\theta_E, \theta_D) &= H(S) + \sum_{x,y} p_{XY}(x, y)\log\left(\frac{p_X(x)\cdot p_Y(y)}{p_{XY}(x, y)}\right) + \\ &+ \sum_{x,y} p_{XY}(x, y)\log\left(\frac{p_{XY}(x, y)}{p_{\hat{S}|Y}(\hat{s}|y; \theta_D)\cdot p_Y(y)}\right) \\ &= H(S) - I(X; Y) + \\ &+ \sum_{x,y} p_{XY}(x, y)\log\left(\frac{p_{X|Y}(x|y)}{p_{\hat{S}|Y}(\hat{s}|y; \theta_D)}\right) \\ &= H(S) - I_{\theta_E}(X; Y) + \\ &+ \mathbb{E}_y\Big[D_{\mathrm{KL}}\Big(p_{X|Y}(x|y)||p_{\hat{S}|Y}(x|y; \theta_D)\Big)\Big].\end{aligned}$$

■

REFERENCES

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, 1948. [Online]. Available: https://ieeexplore.ieee.org/document/6773024/

[2] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cogn. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[3] A. M. Tonello, N. A. Letizia, D. Righini, and F. Marcuzzi, "Machine learning tips and tricks for power line communications," *IEEE Access*, vol. 7, pp. 82434–82452, 2019.

[4] S. Dorner, S. Cammerer, J. Hoydis, and S. T. Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.

[5] Y. Jiang, H. Kim, H. Asnani, S. Kannan, S. Oh, and P. Viswanath, "Turbo autoencoder: Deep learning based channel codes for point-to-point communication channels," in *Advances in Neural Information Processing Systems 32*. Red Hook, NY, USA: Curran Assoc., Inc., 2019, pp. 2758–2768.

[6] F. Alberge, "Deep learning constellation design for the AWGN channel with additive radar interference," *IEEE Trans. Commun.*, vol. 67, no. 2, pp. 1413–1423, Feb. 2019.

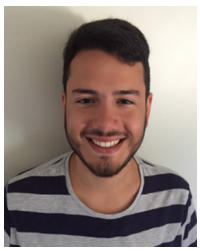
[7] A. Felix, S. Cammerer, S. Dörner, J. Hoydis, and S. T. Brink, "OFDM-autoencoder for end-to-end learning of communications systems," in *Proc. IEEE 19th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2018, pp. 1–5.

[8] I. J. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, vol. 2, 2014, pp. 2672–2680.

[9] T. J. O'Shea, T. Roy, N. West, and B. C. Hilburn, "Physical layer communications system design over-the-air using adversarial networks," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, 2018, pp. 529–532.

[10] D. Righini, N. A. Letizia, and A. M. Tonello, "Synthetic power line communications channel generation with autoencoders and gans," in *Proc. IEEE Int. Conf. Commun. Control Comput. Technol. Smart Grids (SmartGridComm)*, 2019, pp. 1–6.

[11] N. A. Letizia, A. M. Tonello, and D. Righini, "Learning to synthesize noise: The multiple conductor power line case," in *Proc. IEEE Int. Symp. Power Line Commun. Appl. (ISPLC)*, May 2020, pp. 1–6.
[12] M. Stark, F. Ait Aoudia, and J. Hoydis, "Joint learning of geometric and probabilistic constellation shaping," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, 2019, pp. 1–6.
[13] S. Cammerer, F. A. Aoudia, S. Dörner, M. Stark, J. Hoydis, and S. T. Brink, "Trainable communication systems: Concepts and prototype," *IEEE Trans. Commun.*, vol. 68, no. 9, pp. 5489–5503, Sep. 2020.
[14] R. Fritschek, R. F. Schaefer, and G. Wunder, "Deep learning for channel coding via neural mutual information estimation," in *Proc. IEEE 20th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2019, pp. 1–5.
[15] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE J.*, vol. 37, no. 2, pp. 233–243, 1991.
[16] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 1096–1103.
[17] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contracting auto-encoders: Explicit invariance during feature extraction," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 1–8.
[18] A. Makhzani and B. J. Frey, "k-sparse autoencoders," in *Proc. 2nd Int. Conf. Learn. Represent. (ICLR)*, 2014,
[19] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," 2014. [Online]. Available: arXiv:1312.6114.
[20] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," in *Proc. 37th Annu. Allerton Conf. Commun. Control Comput.*, 1999, pp. 368–377.
[21] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, p. 19.
[22] A. Achille and S. Soatto, "Emergence of invariance and disentanglement in deep representations," *J. Mach. Learn. Res.*, vol. 19, no. 1, pp. 1–34, 2018.
[23] M. I. Belghazi *et al.*, "Mutual information neural estimation," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, Jul. 2018, pp. 531–540.
[24] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," 2017. [Online]. Available: arXiv:1703.00810.
[25] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.
[26] Y.-I. Moon, B. Rajagopalan, and U. Lall, "Estimation of mutual information using kernel density estimators," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 52, pp. 2318–2321, Sep. 1995.
[27] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 69, Jun. 2004, Art. no. 066138.
[28] X. Nguyen, M. J. Wainwright, and M. I. Jordan, "Estimating divergence functionals and the likelihood ratio by convex risk minimization," *IEEE Trans. Inf. Theory*, vol. 56, no. 11, pp. 5847–5861, Nov. 2010.
[29] B. Poole, S. Ozair, A. van den Oord, A. Alemi, and G. Tucker, "On variational bounds of mutual information," in *Proc. 36th Int. Conf. Mach. Learn.*, 2019, pp. 5171–5180.
[30] M. D. Donsker and S. R. S. Varadhan, "Asymptotic evaluation of certain markov process expectations for large time. IV," *Commun. Pure Appl. Math.*, vol. 36, no. 2, pp. 183–212, 1983.
[31] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–15.
[32] G. Pereyra, G. Tucker, J. Chorowski, L. Kaiser, and G. E. Hinton, "Regularizing neural networks by penalizing confident output distributions," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017, pp. 1–12.
[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 2818–2826.
[34] M. Abadi *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016. [Online]. Available: arXiv:1603.04467.
[35] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Red Hook, NY, USA: Curran Assoc., Inc., 2019.
[36] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.



NUNZIO A. LETIZIA (Graduate Student Member, IEEE) received the Laurea Magistrale degree in electrical engineering (*summa cum laude*) from the University of Udine, Italy, in 2018. He is currently pursuing the Ph.D. degree with the University of Klagenfurt, Austria, where he is currently a University Assistant. During his studies, he was awarded with a full scholarship with Scuola Superiore dell'Universitá degli studi di Udine. His research interests spread from mathematics and physics to communication engineering. His research areas are in the field of signal processing and statistical learning.

ANDREA M. TONELLO (Senior Member, IEEE) received the Dr.Eng. degree (Hons.) in electronics and the Dr.Res. degree in electronics and telecommunications from the University of Padova, Italy, in 1996 and 2002, respectively. From 1997 to 2002, he was with Bell Labs-Lucent Technologies, Whippany, NJ, USA, as a Member of the Technical Staff. Then, he was promoted to Technical Manager and appointed to the Managing Director of the Bell Labs Italy Division. In 2003, he joined the University of Udine, Udine, Italy, where he became an Aggregate Professor in 2005, and an Associate Professor in 2014. He is currently a Professor of Embedded Communication Systems with the University of Klagenfurt, Klagenfurt, Austria. He is also the Founder of the spinoff company, WiTiKee. He received several awards, including the Distinguished Visiting Fellowship from the Royal Academy of Engineering, U.K., in 2010, the IEEE VTS and COMSOC Distinguished Lecturer Awards in 2011, 2015, and 2018, the UC3M Chair of Excellence (2019–2020) and nine best paper awards. He served/s as an Associate Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, IEEE TRANSACTIONS ON COMMUNICATIONS, IEEE ACCESS, and *IET Smart Grid*. He has been the Chair of the IEEE ComSoc Technical Committee on Power Line Communications from 2014 to 2018. He is currently the Chair of the IEEE ComSoc TC on Smart Grid Communications. He has been appointed as the Director of Industry Outreach of the IEEE ComSoc for the term 2020–2021.