Joint Source-Channel Coding Over Additive Noise Analog Channels Using Mixture of Variational Autoencoders

Yashas Malur Saidutta, *Student Member, IEEE*, Afshin Abdi, *Member, IEEE*, and Faramarz Fekri, *Fellow, IEEE*

***Abstract*— In this paper, we present a learning scheme for Joint Source-Channel Coding (JSCC) over analog independent additive noise channels. We formulate the learning problem by showing that the minimization loss function from rate-distortion theory, is upper bounded by the loss function of the Variational Autoencoder (VAE). We show that when the source dimension is greater than the channel dimension, the encoding of two source samples in the neighborhood of each other need not be near each other. Such discontinuous projection needs to be accounted for by using multiple encoders and selecting an encoder to encode samples on a particular side of the discontinuity. We explore two selection methodologies, one based on an intuitive rule and the other where it is posed as a learning task in a Mixture-of-Experts (MoE) setup. We analyze the gradients of these methods and reason why the latter is better at avoiding local optima. We show the efficacy of the proposed methodology by simulating the performance of the system for JSCC of Gaussian sources over AWGN channels and showing that the learned solutions are close to or better than the ones proposed earlier. The proposed methodology is also naturally capable of generalizing to other source distributions which we showcase by simulating for Laplace sources. The learned systems are also robust to changes in channel conditions. Further, a single system can be trained to generalize over a range of channel conditions provided the channel conditions are known at both the transmitter and the receiver. Finally, we evaluate our proposed methodology on three different image datasets and showcase consistent improvement over existing methods due to the VAE formulation.**

***Index Terms*— Joint source-channel coding, machine learning, deep learning, Variational Autoencoders.**

I. Introduction

JOINT Source-Channel Coding (JSCC) has been a complex and intriguing problem that has captured the interest of researchers alike for more than half a century. Of particular interest is the problem of JSCC over analog channels, the roots of which can be traced back to the works of Shannon in the 1940s [1]. Shannon visualized JSCC over analog channels as a geometric mapping of a symbol in the source space projected onto a lower-dimensional surface embedded in the source space. However, two main questions arise, "Why analog communication?" and "Why Joint Source-Channel Coding?". Even though digital communication is popular, it suffers from a notable setback, i.e., a system designed for a particular channel condition does not show graceful performance change if the channel conditions vary from the design parameters [2], [3]. To answer the second question, the separation theorem [4] assumes infinite delay and complexity, which might be problematic for many practical communication systems. In this work, to exploit the insight of prior research, we initially focus on the JSCC of Gaussian sources over AWGN channels for bandwidth compression. We propose a learned neural network based solution that brings diverse ideas from machine learning like Variational Autoencoders (VAEs) and Mixture-of-Experts (MoE) to achieve/match state-of-the-art performance. Additionally, we show that the designed methodology can be generalized to other source distributions like Laplacian sources and images.

A. Joint Source-Channel Coding of Gaussian Sources over AWGN Channels

The basic premise of the problem is to send a sample from an m-dimensional multivariate Gaussian source over a k-dimensional AWGN channel. It can also be alternatively viewed as sending m symbols of a scalar Gaussian source over k uses of an AWGN channel. Bandwidth compression is achieved when $m > k$. The encoder is a function map denoted as $g_e : \mathbb{R}^m \rightarrow \mathbb{R}^k$ and the decoder is denoted as $g_d : \mathbb{R}^k \rightarrow \mathbb{R}^m$. A more concrete definition of the problem will be given in Sec. II. Reference [5] suggested the use of linear encoders that achieve the Shannon limit when $m = k$. However, for $m \neq k$, linear encoders leave large performance gaps from the optimal Shannon limit when channel noise is low [6], [7]. To overcome this, [8]–[10] suggested the use of a Vector Quantization (VQ) based method called Power Constrained Channel Optimized Vector Quantization (PCCOVQ). However, the size of the resulting codebook increases exponentially with increasing source or channel dimensions, and also increases as the channel conditions improve. This leads to scalability issues. By leveraging the insight obtained from the PCCOVQ codebook, a parametric

Manuscript received July 18, 2020; revised December 1, 2020; accepted February 1, 2021. Date of publication May 10, 2021; date of current version June 17, 2021. This work was supported by the National Science Foundation under Award ID MLWiNS-2003002 and in part by Intel Company. *(Corresponding author: Yashas Malur Saidutta.)*

Yashas Malur Saidutta and Faramarz Fekri are with the Department of Electrical and Computer Engineering, Geogia Institute of Technology, Atlanta, GA 30318 USA (e-mail: yashas.saidutta@gatech.edu; faramarz.fekri@gatech.edu).

Afshin Abdi was with the Department of Electrical and Computer Engineering at Geogia Institute of Technology, Atlanta, GA 30318 USA. He is now with the Qualcomm Technologies, Inc., San Diego, CA 92121 USA (e-mail: abdi@gatech.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2021.3078489.

Digital Object Identifier 10.1109/JSAC.2021.3078489

curve was suggested [11], [12]. This was able to overcome the scalability issues for $m = 2, k = 1$. Intuition was used to extend the use of these parametric curves when $m > 2, k = 1$ [2], [13]–[16]. Ad-hoc solutions to the $k \neq 1$ problem were proposed in [2] that make use of $m, k = 1$ systems as sub-components. However, the reliance on intuition to parametrize the encoders led to poor performance. Further, the use of MMSE decoder in [2] requires $\mathcal{O}(p^m)$ computations to decode a received symbol, where p is the number of points on the grid per dimension. To relax the parametric assumption, [3] considered encoders and decoders as a mapping defined by a table and used functional optimization to learn very efficient maps. However, using a tabular mapping leads to scalability issues similar to that of PCCOVQ. Additionally, if we consider that the input dimension of the encoder and decoder table is defined over grid size of p points per dimension, the computational cost of updating the encoder/decoder table is exponential, i.e., $\mathcal{O}(p^{m+k})$. To overcome this, we suggested the use of Deep Neural Networks to learn the encoders and decoders while exploiting optimization [17]–[19]. In this paper, we further improve upon these neural network based solutions and showcase their learning capability.

B. Deep Neural Networks for Joint Source-Channel Coding

In recent years, deep neural networks were explored for JSCC of data with an explicit structure like images [20]–[23] and text [24]. For data lacking explicit structure, there have been applications of deep learning in areas like physical layer communication [25]–[30] and channel coding [31]–[35] to name a few. However, when it comes to JSCC, there are few works like [20], [36]. The solutions presented in [36] only explored the application on binary data over a binary AWGN channel. The closest to our work in the setting of images is that of [20]. However, the formulation of the loss is similar to that of an autoencoder. Here, we show theoretically that the Variational Autoencoder framework is more naturally suited to solve the problem.

C. Variational Autoencoders

The VAE was proposed as a mechanism to train a deep generative model [37]. VAEs have been used in various applications like data generation [38], semi-supervised learning [39], topic modeling of documents [40], disentangling factors of data generation [41], [42] etc. However, it is of particular interest to us because the VAE has the same encoder-decoder structure like a JSCC system. Secondly, the VAE training involves an optimization that has two terms: a reconstruction error term and a Kullback-Leibler (KL) divergence term. We show that, if viewed through the lens of Joint Source-Channel Coding, the second term can be reinterpreted as a power constraint term. This insight also helps set the groundwork for the future use of the abundant VAE literature, for furthering the design of JSCC over analog-channels systems for general sources.

D. Mixture-of-Experts

The Mixture-of-Experts (MoE) methodology was a general idea proposed to encapsulate the principle of "Divide and conquer" [43]. The system consists of an ensemble of learners (called experts) and a classifier (called selector/gate). The selector activates an individual component of the ensemble based on the input. The MoE methodology can be used in different ways. Our interest is to use it in a competitive setting to combat discontinuity that the encoders face during JSCC. In a competitive setting, the system attempts to not only learn the experts but also their region of expertise while simultaneously encouraging the system to associate each input region with a single expert. The requirement of training a selector along with an ensemble of experts adds a layer of non-convexity to the training process. A lot of research has focused on training such models using various methods like the Newton-Raphson method [44], Expectation Maximization [45], tensor decomposition [46], genetic training [47] etc. In our problem, we use the original loss function proposed by [43] with a modification where we train the experts and the selector in an alternating manner.

The contributions of this paper are as follows:

1) We propose the use of VAEs to learn the encoders and decoders for JSCC of sources over independent additive noise channels.
2) We show that VAEs are very similar to the JSCC system and their loss function minimizes an upper-bound on the one obtained from rate-distortion theory.
3) We show that discontinuous projections play an important role in bandwidth compression and propose the use of multiple encoders with a universal decoder. Each encoder network provides a possible encoding, only one of which is selected for transmission. We explore two possibilities of selection, one based on an intuitive rule and the other where the selection itself is posed as a learning problem.
4) We show that using the proposed system of VAEs in conjunction with MoEs helps us not to rely on intuition from imagination or prior work. Thus, we can improve upon existing solutions for the JSCC of Gaussian sources over AWGN channels.
5) We further show that the methodology is not dependent on the source distribution being Gaussian by showcasing its performance over Laplace sources.
6) We show that the solutions are robust to channel noise variations of up to ± 5dB compared to the channel noise conditions it is designed for. This robustness can be further improved by training a single system over a range of channel conditions.
7) Finally, we experiment on three different image datasets. We show that the efficacy of the dual encoder is affected by the limited size of the datasets and the fact that the images are isolated points distanced from each other in a high dimensional space. However, the reformulation of the JSCC over analog channels as a VAE lends itself to superior performance over existing methods.

For notation, we use bold upper case letters to indicate vector random variables, uppercase letters to denote random variables and bold lower case letters to denote vectors. We use the notation $\mathbf{x}_{(i)}$ to denote the i^{th} component of a vector.

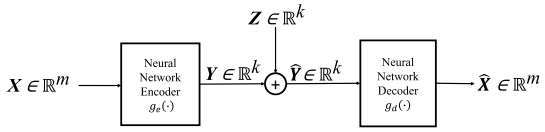


Fig. 1. Analog point to point communication system.

II. Problem Definition and Shannon Limits

A. Problem Definition

Fig. 1 shows an analog point-to-point communication. In our problem of interest, the source signal is an m-dimensional random variable with i.i.d. components. In the case of Gaussian sources, each component is zero mean and with a variance of σ_x^2. The encoder $g_e(\cdot)$ transforms the input signal $\mathbf{x}$ to a k-dimensional symbol denoted by $\mathbf{y}$. This is then transmitted across a k-dimensional additive noise channel. The noise is represented by $\mathbf{z}$. In the case of an AWGN channel, each noise component is i.i.d. zero mean with variance σ_n^2. The decoder then attempts to reconstruct the transmitted signal from the noisy received symbol $\hat{\mathbf{y}}$. An analogous way of looking at this problem is that m independent symbols from a 1-dimensional source are transmitted over k independent uses of the channel. The objective of the system is to minimize some distortion measure between the input and the reconstruction while ensuring a power constraint is satisfied. Most popularly the squared error defined as $\mathbb{E}\left[||\boldsymbol{X}-\hat{\boldsymbol{X}}||_2^2\right]$ is used. The transmission power constraint is defined as $\frac{1}{k}\mathbb{E}\left[||\boldsymbol{Y}||_2^2\right] \leq P_T$. Here, P_T is the transmission power constraint on a single channel use. The performance of the system is characterized by the Signal to Distortion Ratio (SDR) achieved at a particular Channel Signal to Noise Ratio (CSNR). They are both defined as

$$\mathrm{SDR(dB)} = 10\log_{10}\left(\frac{\sigma_x^2}{\frac{1}{m}\mathbb{E}\left[||\boldsymbol{X}-\hat{\boldsymbol{X}}||_2^2\right]}\right) \tag{1a}$$

$$\mathrm{CSNR(dB)} = 10\log_{10}\left(\frac{\frac{1}{k}\mathbb{E}\left[||\boldsymbol{Y}||_2^2\right]}{\sigma_n^2}\right). \tag{1b}$$

B. Asymptotic Shannon Limits for Gaussian Sources

The analogous view of the problem described in Sec. II-A provides us with a way to derive the asymptotic Shannon limits. The rate-distortion of a scalar Gaussian random variable under the squared distortion criterion is given by [48]

$$R(D) = \frac{1}{2}\log_2^+\left(\frac{\sigma_x^2}{D}\right). \tag{2}$$

where, σ_x^2 is the variance of the random variable, D is the distortion and $\log_2^+(\cdot)$ is used to represent $\max(0, \log_2(\cdot))$. The capacity of an AWGN channel with power constraint P_T and noise variance σ_n^2 is given by [48]

$$C_{\mathrm{AWGN}} = \frac{1}{2}\log_2\left(1+\frac{P_T}{\sigma_n^2}\right). \tag{3}$$

To ensure reliable transmission (under the assumption that we are operating in the regime where $D \leq \sigma_x^2$) of m samples across k AWGN channel uses, we need $mR(D) \leq kC_{\mathrm{AWGN}}$. The optimal distortion D_{opt} is obtained when the inequality is satisfied by an equality. Thus, the optimal SDR can be written as

$$SDR_{\mathrm{opt}}(\mathrm{dB}) = 10\frac{k}{m}\log_{10}\left(1+\frac{P_T}{\sigma_n^2}\right) \tag{4}$$

III. Joint Source-Channel Coding and Variational Autoencoders

In this section, we consider the similarity between Variational Autoencoders (VAE) [37] and Joint Source-Channel Coding. First, we provide background on VAEs. Next, we showcase the similarity between VAEs and the Joint Source-Channel Coding problem. Finally, we show that the VAEs optimize an upper bound on the minimization loss function from rate-distortion theory.

A. VAE Background

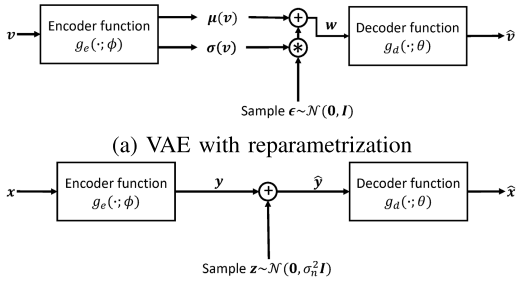
The VAE is a deep generative model that transforms a latent variable $\mathbf{W}$ with some prescribed latent distribution into a data random variable $\mathbf{V}$ whose distribution is unknown. However, samples from the data distribution are available. Let the generator of the VAE be denoted as $g_d(\cdot;\theta)$ where $g_d(\cdot)$ is a deep neural network with parameters θ. Then, the objective of the generative model is to maximize the log-likelihood of the observed data

$$\max_\theta \mathbb{E}_{\mathbf{V}}\left[\log\left(p_\theta(\mathbf{v})\right)\right] = \max_\theta \mathbb{E}_{\mathbf{V}}\left[\log \mathbb{E}_{\mathbf{W}}\left[p_\theta(\mathbf{v}|\mathbf{w})\right]\right] \tag{5}$$

The formulation of $p_\theta(\mathbf{v}|\mathbf{w})$ depends on the type of reconstruction loss to be minimized. Usually, squared error is used as reconstruction error and $p_\theta(\mathbf{v}|\mathbf{w})$ becomes $\mathcal{N}\left(\mathbf{v};\hat{\mathbf{v}},\sigma^2 I\right)$, where $\hat{\mathbf{v}} = g_d(\mathbf{w};\theta)$ and σ^2 is a hyperparameter. To efficiently maximize the log-likelihood, knowledge about the intractable posterior $p_\theta(\mathbf{w}|\mathbf{v})$ is required. Instead, a variational approximation provided by a separate inference (encoder) network is used. This approximation denoted by $q_\phi(\mathbf{w}|\mathbf{v})$, is modeled as $\mathcal{N}\left(\mathbf{w};\boldsymbol{\mu}_\phi(\mathbf{v}),\mathrm{diag}\left(\boldsymbol{\sigma}_\phi^2(\mathbf{v})\right)\right)$, where $\boldsymbol{\mu}_\phi(\cdot)$ and $\boldsymbol{\sigma}_\phi(\cdot)$ are generated by the encoder network $g_e(\cdot;\phi)$. In the end-to-end training objective both the generator likelihood is improved and the KL divergence between the variational approximation and the true posterior is minimized. This is called as the Evidence Lower BOund (ELBO) loss function is proposed [37]. The computation of the ELBO loss function involves a sampling operation. Since the sampling operation does not have defined gradients, [37] suggested a reparametrization trick. Thus the final form of the ELBO loss function used to train VAEs becomes [37]

$$\max_{\theta,\phi}\mathbb{E}_{\mathbf{V},\boldsymbol{\epsilon}\sim\mathbf{E}}\left[\log p_\theta(\mathbf{v}|\boldsymbol{\mu}_\phi(\mathbf{v})+\boldsymbol{\sigma}_\phi(\mathbf{v})\circledast\boldsymbol{\epsilon})\right] - \mathbb{E}_{\mathbf{V}}\left[KL(q_\phi(\mathbf{w}|\mathbf{v})||p(\mathbf{w}))\right]. \tag{6}$$

Here, $\mathbf{E}$ is a standard normal random variable with independent dimensions whose dimensionality is equal to that of the latent dimension, $\boldsymbol{\epsilon}$ is a sample from $\mathbf{E}$, and $\circledast$ is an element-wise multiplication operator between two vectors.



(a) VAE with reparametrization

(b) Joint Source Channel Coding System with AWGN noise

Fig. 2. Joint source channel coding and variational Autoencoders.

B. Similarity Between Joint Source-Channel Coding and VAEs

Fig. 2 shows the similarity between the reparametrized VAE and the Joint Source-Channel Coding setup. The major differences between the two setups are:

1) The VAE latent variable $\mathbf{W}$ is the analogue of the noisy received signal $\hat{\mathbf{Y}}$.
   a) In VAEs, the distribution of $\mathbf{w}$ is assumed to be the standard normal distribution with independent components.
   b) In JSCC, the distribution of $\hat{\mathbf{Y}}$ is dependent on the distribution of the source and the channel noise.
2) The covariance matrix:
   a) In VAEs, $\mathbf{W}$ is sampled from a normal distribution whose mean and covariance matrix are both determined by the encoder.
   b) In JSCC, $\hat{\mathbf{Y}}$ is sampled from a normal distribution (AWGN channel), whose mean is determined by the encoder. The covariance matrix is a characteristic of the channel.

After replacing $\mathbf{v}$ and $\mathbf{w}$ with their JSCC counterparts in (6), the KL divergence term becomes $\mathbb{E}_{\mathbf{X}}\left[KL(q_\phi(\hat{\mathbf{y}}|\mathbf{x})||p(\hat{\mathbf{y}}))\right]$. Since, the channel is AWGN, $q_\phi(\hat{\mathbf{y}}|\mathbf{x})$ is normally distributed. As the differential entropy of a Gaussian distribution is only dependent on the covariance matrix, $\mathbb{E}_{\mathbf{X}}\left[KL(q_\phi(\hat{\mathbf{y}}|\mathbf{x})||p(\hat{\mathbf{y}}))\right]$ can be simplified to $\mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[\log p(\hat{\mathbf{y}})\right]$ in the maximization objective. Applying all this, we can simplify the loss function of VAEs (6) and use it to train the JSCC system as

$$\max_{\theta,\phi} \mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[\log p_\theta(\mathbf{x}|\hat{\mathbf{y}})\right] + \mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[\log p(\hat{\mathbf{y}})\right]. \tag{7}$$

Although the above loss function (7) is got from VAEs based on the similarity of the system functioning, we show in Sec. III-C that this loss function has a theoretical significance. Further, even though the above loss function (7) assumes AWGN channels, the same loss function holds for any independent additive noise channel. This is because differential entropy for any distribution is translation invariant.

Now, we choose the distribution to model $p_\theta(\mathbf{x}|\hat{\mathbf{y}})$. For ease of notation, let us define $\hat{\mathbf{x}} := g_d(\hat{\mathbf{y}})$. Different choices reduce different error metrics between $\mathbf{x}$ and $\hat{\mathbf{x}}$. For example,

1) If $p_\theta(\mathbf{x}|\hat{\mathbf{y}}) = \mathcal{N}(\mathbf{x}; \hat{\mathbf{x}}, \lambda I)$, then the system learns to minimize the squared error, i.e., $||\mathbf{x}-\hat{\mathbf{x}}||_2^2$. The value of λ determines the operating point, i.e., the power constraint of the JSCC system.
2) If $p_\theta(\mathbf{x}|\hat{\mathbf{y}}) = \prod_{l=1}^{k} \frac{1}{2}\exp^{-\frac{1}{\lambda}||\mathbf{x}_{(l)}-\hat{\mathbf{x}}_{(l)}||_1}$, i.e., each dimension is modeled by an i.i.d. Laplace distribution with scale parameter λ, then the system learns to minimize the absolute error, i.e., $||\mathbf{x}-\hat{\mathbf{x}}||_1$.

For the sake of simplicity we assume the use of squared distortion for the rest of the paper. The loss function in (7) becomes

$$\min_{\theta,\phi} \mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[||\mathbf{x}-\hat{\mathbf{x}}||_2^2\right] - \lambda \mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[\log p(\hat{\mathbf{y}})\right]. \tag{8}$$

1) Distribution of Received Codewords: Additionally, in the case of VAEs, $p(\mathbf{w})$ is assumed to have a standard independent component multivariate Gaussian. However, for JSCC this need not be true. Even though the asymptotic matching distribution for transmission over an AWGN channel is the Gaussian distribution, in the delay limited case it need not be true. Instead we assume that $p(\hat{\mathbf{y}})$ has an independent component Generalized Gaussian Distribution (GGD). This can be written as

$$p(\hat{\mathbf{y}}) = \prod_{l=1}^{k} \frac{\beta_l}{2\alpha_l \Gamma(1/\beta_l)} e^{-\left(\frac{|\hat{\mathbf{y}}_{(l)}|}{\alpha_l}\right)^{\beta_l}} \tag{9}$$

where, α_l and β_l are the scale and shape parameters; and $\Gamma(\cdot)$ represents the gamma function. The GGD generalizes on the Laplacian and the Gaussian distributions. If $\beta = 1$ the PDF becomes that of a Laplace distribution (double exponential distribution) and if $\beta = 2$ the PDF becomes that of a Gaussian distribution. Since, we do not know the values of α_l and β_l we pose this as a learning problem. The training methodology for this becomes clear in the next subsection.

Putting all this together, for the JSCC of Gaussian sources over AWGN channels where the reconstruction error is represented by MSE, the loss function used to train the system is

$$\min_{\theta,\phi} \mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[||\mathbf{x}-\hat{\mathbf{x}}||_2^2 + \lambda \sum_{l=1}^{k}\left(\frac{|\hat{\mathbf{y}}_{(l)}|}{\alpha_l}\right)^{\beta_l}\right]. \tag{10}$$

C. Information Theoretic View of Using VAEs for JSCC

Before going into the theorem let us define $q_\phi(\hat{\mathbf{y}}) \triangleq \int q_\phi(\hat{\mathbf{y}}|\mathbf{x})p(\mathbf{x})\mathrm{d}\mathbf{x}$. This is the PDF of $\hat{\mathbf{Y}} = g_e(\mathbf{X};\phi) + \mathbf{Z}$.

Theorem 1: The minimization objective function of VAEs used for JSCC tasked with minimizing the squared error distortion while subject to independent additive noise is:

$$\mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[||\mathbf{x}-\hat{\mathbf{x}}||_2^2\right] - \lambda \mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[\log p(\hat{\mathbf{y}})\right] + c, \tag{11}$$

where c is a constant. This objective is an upper bound on the minimization objective function obtained from rate-distortion theory, provided the following assumptions are satisfied.

1) *The JSCC system operates at channel capacity, i.e., $I(\hat{\mathbf{X}};\mathbf{X}) = kC$.*
2) *The encoder and decoder are deterministic.*

Proof: The objective from rate distortion theory in terms of the distortion-rate function for m-dimensional source can

be written as,

$$\underset{p(\hat{\mathbf{x}}|\mathbf{x}):I(\hat{\mathbf{X}};\mathbf{X})\leq mR}{\arg\min} \frac{1}{m}\mathbb{E}_{\mathbf{X},\hat{\mathbf{X}}}\left[||\mathbf{x}-\hat{\mathbf{x}}||_2^2\right] \tag{12}$$

For reliable communication we have $mR \leq kC$. We can write the Lagrangian of this constrained optimization as

$$\underset{p(\hat{\mathbf{x}}|\mathbf{x})}{\arg\min} \frac{1}{m}\mathbb{E}_{\mathbf{X},\hat{\mathbf{X}}}\left[||\mathbf{x}-\hat{\mathbf{x}}||_2^2\right] + \lambda(I(\hat{\mathbf{X}};\mathbf{X}) - kC) \tag{13}$$

where, $\lambda \geq 0$. To minimize the distortion, we assume that the system will try to use the full channel capacity, i.e., we assume that the optimal solution is on the boundary of the constraint set. This assumption allows us to solve for $\lambda = -\frac{dD_{\text{opt}}}{d(kC)} = \lambda_0$ [49] giving us

$$\underset{p(\hat{\mathbf{x}}|\mathbf{x})}{\arg\min} \frac{1}{m}\mathbb{E}_{\mathbf{X},\hat{\mathbf{X}}}\left[||\mathbf{x}-\hat{\mathbf{x}}||_2^2\right] + \lambda_0(I(\hat{\mathbf{X}};\mathbf{X}) - kC). \tag{14}$$

Since the deterministic encoder and decoder parametrized by ϕ and θ determine the distribution $p(\hat{\mathbf{x}}|\mathbf{x})$. We can write (14) as

$$\underset{\phi,\theta}{\arg\min}\, \mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[||\mathbf{x}-\hat{\mathbf{x}}||_2^2\right] + m\lambda_0(I(\hat{\mathbf{X}};\mathbf{X}) - kC). \tag{15}$$

It remains to show that the objective function in (11) is an upper bound on the objective function of (15).

Since $\mathbf{X} \rightarrow \mathbf{Y} \rightarrow \hat{\mathbf{Y}} \rightarrow \hat{\mathbf{X}}$ forms a Markov chain, from the data processing inequality we have, $I(\hat{\mathbf{X}};\mathbf{X}) \leq I(\hat{\mathbf{Y}};\mathbf{Y})$. Thus we get

$$\begin{aligned}
&\mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[||\mathbf{x}-\hat{\mathbf{x}}||_2^2\right] + m\lambda_0(I(\hat{\mathbf{X}};\mathbf{X}) - kC) \\
&\quad\leq \mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[||\mathbf{x}-\hat{\mathbf{x}}||_2^2\right] + m\lambda_0(I(\hat{\mathbf{Y}};\mathbf{Y}) - kC) \\
&\quad= \mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[||\mathbf{x}-\hat{\mathbf{x}}||_2^2\right] + m\lambda_0\ H(\hat{\mathbf{Y}}) + c.
\end{aligned} \tag{16}$$

Here $c \triangleq -\lambda_0(H(\mathbf{Z}) + kC)$ is a constant w.r.t. parameters θ and ϕ.

The entropy of $H(\hat{\mathbf{Y}})$ is computed with respect to the distribution of noisy received codewords denoted as $q_\phi(\hat{\mathbf{y}})$

$$\mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[||\mathbf{x}-\hat{\mathbf{x}}||_2^2\right] - m\lambda_0\mathbb{E}_{\hat{\mathbf{Y}}\sim q_\phi(\hat{\mathbf{y}})}\left[\log(q_\phi(\hat{\mathbf{y}}))\right] + c. \tag{17}$$

Since, $q_\phi(\hat{\mathbf{y}})$ is intractable to compute, we use an approximation $p(\hat{\mathbf{y}})$. We multiply and divide the argument of the $\log$ term in (17) by $p(\hat{\mathbf{y}})$. After rearranging the terms we get

$$\begin{aligned}
&\mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[||\mathbf{x}-\hat{\mathbf{x}}||_2^2\right] \\
&\quad -m\lambda_0\mathbb{E}_{\hat{\mathbf{Y}}\sim q_\phi(\hat{\mathbf{y}})}\left[\log(p(\hat{\mathbf{y}}))\right] \\
&\quad -m\lambda_0\ KL(q_\phi(\hat{\mathbf{y}})||p(\hat{\mathbf{y}})) + c.
\end{aligned} \tag{18}$$

Since, the KL divergence is always ≥ 0, by dropping it we get an upper bound that can be written as

$$\mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[||\mathbf{x}-\hat{\mathbf{x}}||_2^2 - m\lambda_0\log(p(\hat{\mathbf{y}}))\right] + c \tag{19}$$

which is the same as (11) with $\lambda = m\lambda_0$. □

Even though, we have presented the theorem for squared error distortion, the proof holds for any differentiable distortion metric such as $||\mathbf{x}-\hat{\mathbf{x}}||_p^p$ where $||\cdot||_p$ is the $\mathcal{L}_p$ norm.

There are two sources that result in the objective becoming an upper bound. The first is due to the data processing inequality $I(\hat{\mathbf{X}};\mathbf{X}) \leq I(\hat{\mathbf{Y}};\mathbf{Y})$, and the second is due to $KL(q_\phi(\hat{\mathbf{y}})||p(\hat{\mathbf{y}})) \geq 0$. To ensure a tight bound w.r.t. the second source of relaxation, the distribution $p(\hat{\mathbf{y}})$ should be designed to minimze the KL divergence. We do so by fitting the parameters of $p(\hat{\mathbf{y}})$ to minimize the negative log-likelihood of samples from $q_\phi(\hat{\mathbf{y}})$ i.e. $\arg\min_{\alpha,\beta} -\mathbb{E}\left[\log\left(p(\hat{\mathbf{y}};\alpha,\beta)\right)\right]$. Here, α and β are parameters of the distribution $p(\hat{\mathbf{y}})$. These parameters are separate from the neural network parameters θ, ϕ.

Proposition 1: For JSCC of Gaussian sources over AWGN channels subject to square distortion, $\lambda_0 = \ln 2\frac{2D_{opt}}{m}$

Proof: When $D = D_{\text{opt}}$, $mR(D_{\text{opt}}) = kC$. Using the definition of $R(D)$ in (2), we get

$$D_{\text{opt}} = \sigma_x^2 2^{-\frac{2kC}{m}} \tag{20}$$

The value of λ_0 can be obtained as

$$\lambda_0 = -\frac{dD_{opt}}{d(kC)} = \ln 2\frac{2D_{opt}}{m}. \tag{21}$$

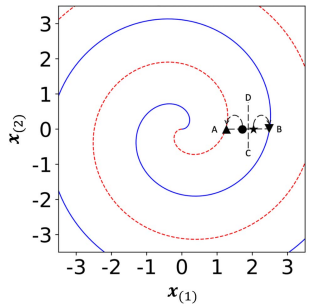
□

IV. Discontinuity in Joint Source-Channel Coding

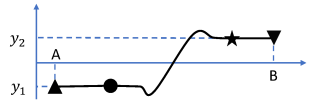
In bandwidth compression, we attempt to represent a source generating signals in an m-dimensional space by projecting it onto a k-dimensional space. Since the data occupies an m-dimensional space and is being represented by a k-dimensional point when $m > k$, the representation is lossy. Using an example, we show that for $m = 2$, $k = 1$ Gaussian JSCC over AWGN samples, good performing solutions need to make use of discontinuity. It is important to note that the discontinuity is not in the curves on which the source symbols are projected onto. It is in the mapping function which the encoder is trying to learn. The basis for discontinuous mapping is also supported by topology theory that shows us that it is not possible to perform dimensionality reduction using a continuous map [1], [50].

To better understand this consider the $m = 2$, $k = 1$ system and the solution of [2] in Fig. 3a. The encoded signal space consists of two spirals (solid blue and dashed red). Any source symbol is represented as a point in the 2D plane. It is then projected onto the closest spiral. The encoding of the projected point is the distance to the projected point from the origin, along the spiral curve (denoted as d). The distance is transformed using a function f that performs a mapping $f(d) : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ mapping. Finally, if the projected point lies on the blue spiral, $+f(d)$ is transmitted and if it lies on the red spiral $-f(d)$ is transmitted. Such a system has proven very successful. Even the functional optimization methods of [3] obtained a similar solution and achieved a gap of $0.5dB$ from the asymptotic Shannon limit.

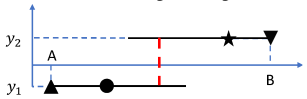
From Fig. 3 we, observe that a spiral-like system has discontinuity during projection, and it requires at least two encoders to model it. Consider two points for encoding, represented by ★ and ●. In Fig. 3a although the two points are close, their projections onto the spirals, represented by ▲ and ▼ respectively, are far away and their final encodings y_1 and y_2 are of the opposite sign. Consider the possibility of using a single encoder as shown in Fig. 3b. Since neural networks are continuous, the encoder network is forced to



(a) Boundary in 2 : 1 encoding.



(b) Effects of using a single encoder.



(c) Problem is overcome using two encoders.

Fig. 3. Boundary effects on a single neural network encoder in 2 : 1 encoding.

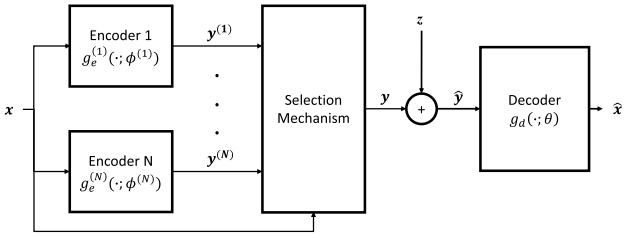


Fig. 4. Multi-encoder system with selection mechanism.

interpolate between the two values. This can lead to spurious encodings in the vicinity of the boundary. Neural networks can learn to closely approximate discontinuous functions. However, the presence of a decoder which depends on the encoder and vice-versa biases the system to poor performing solutions when a single encoder network is used. Alternatively, if there are two encoder networks, one encoder can learn on one side of the boundary (the boundary is represented by the line CD) and the other encoder on the other side. So, given two encoders and an appropriate selection mechanism, we can effectively model the discontinuity and improve the performance of the system.

Fig. 4 shows the proposed JSCC system employing multiple encoder networks. The multiple encoder networks provide various possible encodings. The selection mechanism then selects only one of the encodings for transmission. Finally, the noisy received codeword is processed by the decoder neural network. The individual encoder neural networks' encoding functions are represented as $g_e^{(i)}(\cdot;\phi^{(i)})$ where, $i \in \{1,\ldots,N\}$ indexes the encoder networks, N represents the number of encoder networks and $\phi^{(i)}$ represents the parameters of the i^{th} encoder network. The encoded value provided by the i^{th} encoder network is represented as $\mathbf{y}^{(i)}$. The decoder neural network implements the decoding function represented by $g_d(\cdot,\theta)$, where θ represents the parameters of the decoder network.

In the following subsections, we explore two ideas for modeling the selection mechanism. The first uses an intuition-based rule as the basis for selection. The second approach removes this intuition-based selection process and trains a separate selector neural network. This network automatically learns which encoder network to choose as a function of the source signal. Finally, we analyze the gradients between the two methods and try to understand their similarity and differences.

A. Rule Based Selection Methodology

Based on the discontinuity hypothesis presented above, the most obvious methodology of selection is to select the encoder that leads to the least approximation error. Thus, the encoding function can be defined as

$$g_e(\mathbf{x}) = g_e^{(j)}(\mathbf{x}) \text{ where, } j = \arg\min_i ||\mathbf{x} - g_d(g_e^{(i)}(\mathbf{x}))||_2^2. \tag{22}$$

However, implementation of this idea revealed a shortcoming. Since the system is solely guided by the projection error, the methodology learns a solution that prefers the use of high power encodings. This eventually leads the system to squander the power budget on higher probability symbols at the cost of high errors on the lower probability symbols. On an average this leads to poorer overall performance. To circumvent this, it is necessary to account for power during the selection process.

Let us denote the individual loss of encoder i encoding a sample $\mathbf{x}$ subject to noise sample $\mathbf{z}$ as

$$\mathcal{L}_{\text{MoE}}^{(i)}(\mathbf{x},\mathbf{z}) = ||\mathbf{x} - g_d(g_e^{(i)}(\mathbf{x}) + \mathbf{z})||_2^2 + \lambda \sum_{l=1}^{k} \left(\frac{|\left(\mathbf{g}_e^{(i)}(\mathbf{x}) + \mathbf{z}\right)_{(l)}|}{\alpha_l} \right)^{\beta_l}. \tag{23}$$

As the noise applied by the channel is unknown during the selection, we use the noiseless version as a proxy. Thus, the overall encoding function $g_e(\mathbf{x})$ is defined as

$$g_e(\mathbf{x}) = g_e^{(j)}(\mathbf{x}) \text{ where, } j = \arg\min_i \mathcal{L}_{\text{MoE}}^{(i)}(\mathbf{x},\mathbf{0}) \tag{24}$$

During training, for every source sample $\mathbf{x}$ an encoder is chosen. This encoding is perturbed by the channel noise and then decoded by the decoder. The loss as in (10) is computed between the chosen encoder and the decoder. The gradients w.r.t. the loss are then applied to the decoder and the chosen encoder. The overall training loss function can be defined as

$$\mathcal{L}_{\text{MoE\_rule}} = \sum_{i=1}^{N} \mathbb{1}_{i=\arg\min_j \mathcal{L}_{\text{MoE}}^{(j)}(\mathbf{x},\mathbf{0})} \mathcal{L}_{\text{MoE}}^{(i)}(\mathbf{x},\mathbf{z}) \tag{25}$$

where, $\mathbb{1}_{i=\arg\min_j \mathcal{L}_{\text{MoE}}^{(j)}(\mathbf{x},\mathbf{0})}$ is an indicator function that returns 1 if the encoder network-i has the least $\mathcal{L}_{\text{MoE}}^{(j)}(\mathbf{x},\mathbf{0})$ $\forall j \in \{1,\ldots,N\}$. We abbreviate this loss as the MoE_$RULE$ loss to indicate the Mixture-of-Experts RULE loss.

Algorithm 1: Rule based training

Initialize: The encoders and decoder neural networks with parameters $\{\phi^{(1)}, \ldots, \phi^{(N)}, \theta\}$ are initialized using variance-scaling initialization [51];
for $iter := 1$ *to* $iter_{max}$ **do** *Training loop*
  Sample $\mathbf{x}$ from $\mathbf{X}$ and $\mathbf{z}$ from $\mathbf{Z}$;
  for $i = 1$ *to* N **do**
    Compute $\mathbf{y}^{(i)} = g_e^{(i)}\left(\mathbf{x}; \phi^{(i)}\right)$;
    Compute $\mathcal{L}_{\text{MoE}}^{(i)}(\mathbf{x}, \mathbf{0})$ as given in (23);
  end
  Evaluate rule: $j = \arg\min_i \mathcal{L}_{\text{MoE}}^{(i)}(\mathbf{x}, \mathbf{0})$;
  Set: $\mathbf{y} = \mathbf{y}^{(j)}$;
  Channel: $\hat{\mathbf{y}} = \mathbf{y} + \mathbf{z}$;
  Decode: $\hat{\mathbf{x}} = g_d(\hat{\mathbf{y}}; \theta)$;
  Compute Loss: Compute the loss $\mathcal{L}$ as given in (25);
  Apply gradients: $\theta \leftarrow \theta + \eta \nabla_\theta \mathcal{L}$ and $\phi^{(j)} \leftarrow \phi^{(j)} + \eta \nabla_{\phi^{(j)}} \mathcal{L}$
end

A clearer functioning of the overall training of the system using the rule-based selection criterion is given in Algo. 1. For ease of notations, we assume a batch size of 1 while describing the algorithm. It is important to note that the gradients of the loss are applied on the decoder and only on the chosen encoder network j. This is because the loss is computed using the encoding from the encoder network j. Applying the gradients on any other encoding network will just corrupt the training process.

Further, since the prior work on Gaussian joint source-channel coding advocated the use of spiral like solutions in [2], [3], [13]–[16], we augmented the input to each of the encoder networks with an extended hyper-spherical coordinate representation of the source symbol $\mathbf{x}$. The standard hyperspherical coordinate representation is defined as

$$\begin{aligned} \boldsymbol{r}_{(1)} &= ||\mathbf{x}||_2 \\ \boldsymbol{r}_{(j+1)} &= \cos^{-1}\left(\frac{\mathbf{x}_{(j)}}{\boldsymbol{r}_{(1)}}\right) \text{ where } j \in \{1, ..m-2\} \\ \boldsymbol{r}_{(m)} &= \begin{cases} \cos^{-1}\left(\dfrac{x_{m-1}}{\sqrt{x_{m-1}^2 + x_m^2}}\right) & \text{if } \mathbf{x}_{(m)} \geq 0 \\ 2\pi - \cos^{-1}\left(\dfrac{x_{m-1}}{\sqrt{x_{m-1}^2 + x_m^2}}\right) & \text{if } \mathbf{x}_{(m)} < 0. \end{cases} \end{aligned} \tag{26}$$

However, due to a discontinuity in the representation when $\mathbf{x}_{(m)}$ changes sign, using the standard hyperspherical coordinate representations causes learning difficulty for the neural networks. Instead we use an extended representation where $\mathbf{r}_{ext} = \left[\mathbf{r}_{(1)}, \mathbf{r}_{(2)}, \ldots, \mathbf{r}_{(m-1)}, \sin(\mathbf{r}_{(m)}), \cos(\mathbf{r}_{(m)})\right]$. We only used the hyperspherical coordinates for the experiments conducted using the rule based selection methodology. For the self learned selection methodology presented next, we removed the use of such intuition completely.

B. Self Learned Selection Methodology

In this methodology, the selection mechanism uses a selector network. The selector network, whose function is represented as $g_s(\cdot; \zeta)$ (parametrized by ζ), accepts the input source symbol $\mathbf{x}$ and outputs a multinomial probability vector $\mathbf{p}$ with size N (number of encoder networks) for that input source symbol. $\mathbf{p}_{(i)}$ represents the probability of selecting encoder network-i to provide the encoding for that source symbol. Since we want a system where one encoder specializes in a region and the other encoders do not, we focus on using a loss that encourages the system to learn competitive encoder networks.

One of the most straightforward loss functions to learn competitive mixture-of-experts can be written as

$$\mathcal{L}_{\text{MoE\_LIN}} = \mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[\sum_{i=1}^{N} \mathbf{p}_{(i)} \mathcal{L}_{\text{MoE}}^{(i)}(\mathbf{x}, \mathbf{z})\right] \tag{27}$$

We call this the Mixture-of-Experts LINear (MoE_LIN) loss. However, both in our experiments and as found by [43], this loss did not perform well. Instead, the following replacement was suggested, which we call the Mixture-of-Experts EXPonential (MoE_EXP) loss

$$\mathcal{L}_{\text{MoE\_EXP}} = \mathbb{E}_{\mathbf{X},\mathbf{Z}}\left[-\log\left(\sum_{i=1}^{N} \mathbf{p}_{(i)} \exp^{-\mathcal{L}_{\text{MoE}}^{(i)}(\mathbf{x},\mathbf{z})}\right)\right]. \tag{28}$$

$\mathcal{L}_{\text{MoE\_EXP}}$ is a better loss function to train the mixture-of-experts system because the gradients to the individual experts are not only weighted by the probabilities prescribed by the selector network but also by the values of the losses themselves [43]. This encourages the system to specialize.

Algorithm 2: Self learned selection training

Initialize: The encoders and decoder neural networks with parameters $\{\phi^{(1)}, \ldots, \phi^{(N)}, \theta\}$ are initialized using variance-scaling initialization [51];
Initialize: The selector network with parameters ζ;
for $iter := 1$ *to* $iter_{max}$ **do** *Training loop*
  Sample: $\mathbf{x} \sim \mathbf{X}$ and $\mathbf{z} \sim \mathbf{Z}$;
  for $i = 1$ *to* N **do**
    Compute $\mathcal{L}_{\text{MoE}}^{(i)}(\mathbf{x}, \mathbf{z})$ as given in (23);
  end
  Compute Loss: Compute the loss $\mathcal{L}_{\text{MoE\_EXP}}$ as given in (28);
  Apply gradients to the decoder: $\theta \leftarrow \theta + \eta \nabla_\theta \mathcal{L}_{\text{MoE\_EXP}}$;
  Apply gradients to the selector: $\zeta \leftarrow \zeta + \eta \nabla_\zeta \mathcal{L}_{\text{MoE\_EXP}}$;
  for $i = 1$ *to* N **do** *Apply gradients to the encoders*
    Apply gradients to encoder network-i: $\phi^{(i)} \leftarrow \phi^{(i)} + \eta \nabla_{\phi^{(i)}} \mathcal{L}_{\text{MoE\_EXP}}$;
  end
end

The overall training of the system using the self-learned selection criterion is given in Algo. 2. For ease of notations, we again assume a batch size of 1. Unlike the rule-based

training, gradients are applied to all the encoders so that they can improve their respective encodings. However, as the next subsection makes it clear, the loss function is formulated in such a way that a larger gradient is applied to the better performing encoder network.

C. Comparison of Rule Based and Self Learned Selection Methods by Gradient Analysis

To better understand the effect on the individual encoder networks, let us look at the gradients given by the various loss functions. For ease of writing, let us assume the case of $k = 1$. The gradient of $\mathcal{L}_{\text{MoE\_LIN}}$ w.r.t. $y^{(i)}$ is given as

$$\frac{\mathrm{d}\mathcal{L}_{\text{MoE\_LIN}}}{\mathrm{d}y^{(i)}} = \mathbf{p}_i \frac{\mathrm{d}\mathcal{L}^{(i)}_{\text{MoE}}(\mathbf{x},\mathbf{z})}{\mathrm{d}y^{(i)}} \tag{29}$$

Similarly, the gradient of $\mathcal{L}_{\text{MoE\_EXP}}$ w.r.t. $y^{(i)}$ is given as

$$\frac{\mathrm{d}\mathcal{L}_{\text{MoE\_EXP}}}{\mathrm{d}y^{(i)}} = \frac{\mathbf{p}_{(i)} \exp^{-\mathcal{L}^{(i)}_{\text{MoE}}(\mathbf{x},\mathbf{z})}}{\sum_{l=1}^{N} \mathbf{p}_{(l)} \exp^{-\mathcal{L}^{(l)}_{\text{MoE}}(\mathbf{x},\mathbf{z})}} \frac{\mathrm{d}\mathcal{L}^{(i)}_{\text{MoE}}(\mathbf{x},\mathbf{z})}{\mathrm{d}y^{(i)}} \tag{30}$$

The gradient from $\mathcal{L}_{\text{MoE\_EXP}}$ is better because the gradient is scaled based on the relative performance of the encoder network-i w.r.t. all other encoder networks. If the encoder network-i is the best performing encoder network for that $\mathbf{x}$, it will get the highest weight and the other encoder-networks' gradients will be damped down. In contrast, the gradient of $\mathcal{L}_{\text{MoE\_LIN}}$ is only scaled by the selector network probability, which initially can be very off thus interfering with the ability of the encoder networks to specialize [43].

The gradient of the rule based loss function $\mathcal{L}_{\text{MoE\_}RULE}$ can be written as

$$\frac{\mathrm{d}\mathcal{L}_{\text{MoE\_}RULE}}{\mathrm{d}y^{(i)}} = \mathbb{1}_{i=\arg\min_j \mathcal{L}^{(j)}_{\text{MoE}}(\mathbf{x},\mathbf{0})} \frac{\mathrm{d}\mathcal{L}^{(i)}_{\text{MoE}}(\mathbf{x},\mathbf{z})}{\mathrm{d}y^{(i)}}. \tag{31}$$

We can see that the gradient in (30) is similar to a soft version of the gradient in (31). The rule based system only trains the chosen encoder where as the mixture-of-experts system with the loss (28) trains all the encoder networks while softly giving more preference to the better performing network. This prevents spurious initialization artefacts from leading to selection biases against encoder networks in the beginning of the training. Further, when an encoder network has much better performance than other encoder networks for some $\mathbf{x}$ (, i.e., $\mathcal{L}^{(i)}_{\text{MoE}}(\mathbf{x},\mathbf{0}) \gg \mathcal{L}^{(j)}_{\text{MoE}}(\mathbf{x},\mathbf{0})\ \forall j \in \{1,\ldots,N\}$ and $j \neq i$), the gradients become approximately the same.

V. Experimental Results

A. Implementation Details

All the neural networks are fully connected neural networks with three layers. The encoder networks and the decoder network each have $40m$ hidden neurons, where m is the source dimension. The selector network has $10m$ neurons in each layer. We use the RBF activation function defined as

$$\mathbf{h}_n = \exp\left(-\boldsymbol{\rho}_n \circledast (W_n \mathbf{h}_{n-1} + \mathbf{b}_n)^2\right) \tag{32}$$

where $\mathbf{h}_n$ is the output of the n^{th} layer of the neural network, W_n is the a matrix that represents the weights between layer n and $n-1$, $\boldsymbol{\rho}_n$ is the scaling vector, and $\mathbf{b}_n$ are the biases. We use a large batch size of 3000 to ensure that there are sufficient samples in the low probability regions of the source distribution.

The neural networks are trained for each power setting and m, k values separately for a maximum of 10^7 iterations. The weights of the network are initialized using the variance scaling initializer [51]. Prior training of the selector network is required to ensure that its selection of any encoder is equiprobable. This ensures that spurious selector network initialization of the does not bias the training to train only a subset of the encoders. We try out five possible initializations by training the systems for 10^5 iterations. We test each of these networks with 5×10^4 samples and select the best performing network for further training. This is done because the networks tend to get stuck in local optima. The results are presented by testing the system over a batch of 10^7 samples.

When training $k \neq 1$ systems, we observed that the system has a tendency to move towards sub-optimal solutions by increasing the power across a subset of channels while ensuring that the average transmission power constraint is satisfied. To overcome this, we modify the loss functions in (10) and (23) to individually weigh the channel powers, i.e., the term $\sum_{l=1}^{k} \left(\frac{|\hat{\mathbf{y}}_{(l)}|}{\alpha_l}\right)^{\beta_l}$ in (10) is replaced by $\sum_{l=1}^{k} \lambda_l \left(\frac{|\hat{\mathbf{y}}_{(l)}|}{\alpha_l}\right)^{\beta_l}$. The λ_l are dynamically adjusted every 10^4 iterations by measuring the power across each of the channels. It is observed that eventually λ_l all approach 1 indicating that their purpose is to modify the optimization landscape to prevent the system from moving towards local optima.

Finally, when P_T is close to σ_n^2 ($\text{CSNR} = 0dB, 5dB$), we find that the system moves to a local optima where it learns a trivial encoding $g_e(\mathbf{x}) = 0\ \forall\ \mathbf{x}$. This is because the two terms of the loss function (10) are almost equal. So, we add a regularization term as below to the final loss function (10)

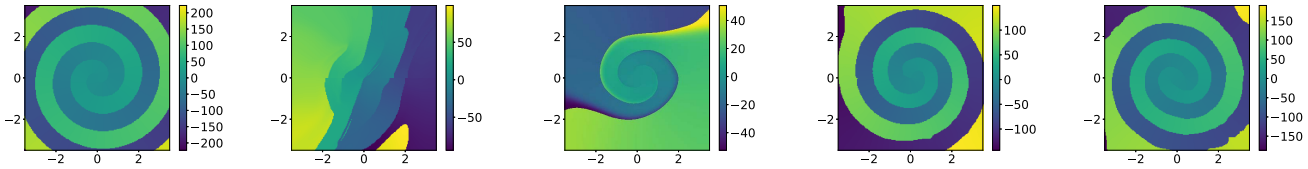
$$\sum_{l=1}^{k} \frac{1}{2} \log \frac{P_T}{\text{var}\left(\mathbf{y}_{(l)}\right)} + \frac{\text{var}\left(\mathbf{y}_{(l)}\right) + \left(\text{mean}\left(\mathbf{y}_{(l)}\right)\right)^2}{2P_T} - \frac{1}{2}. \tag{33}$$

This regularization term forces the KL divergence between the input distribution across each channel to be $\mathcal{N}(0, P_T)$. We do not modify the individual encoder loss (23).

For training the α_l and β_l, we incorporated certain insights to prevent overfitting. Firstly, α_l was fixed as a function of β_l by fixing the variance of individual GGD distributions to be $P_T + \sigma_n^2$. Secondly, we set $\beta_l = \beta\ \forall\ l \in \{1,\ldots,k\}$. We then fit the β parameter by initializing it to 2 and restricting its range to $[1, 2]$. The training for β was done once in every 10^4 iterations of the encoder decoder training by performing gradient descent on the negative of maximum likelihood computed over 10^4 samples.

We observed that during the training, gradients to the selector network in Mixture-of-Experts can saturate. To avoid this we used the standard loss function of (28) to train the encoders and the decoder networks. To train the selector networks we used the modified loss function

$$\mathcal{L}_{SEL} = \gamma \mathcal{L}_{\text{MoE\_EXP}} + (1-\gamma)\mathcal{L}_{\text{MoE\_LIN}} \tag{34}$$



(a) Spiral encoder [2] (b) Single NN encoder (c) Projection based MoE (d) Rule based MoE (e) Self learned MoE

Fig. 5. Encoding functions for $m = 2, k = 1$ systems at CSNR $= 30$dB.

TABLE I

ENCODER(E), DECODER(D), AND SELECTOR(S) NETWORK ARCHITECTURES USED ON MNIST, FASHION MNIST (FMNIST), AND CIFAR-10 DATASETS. FULLY CONNECTED LAYERS ARE REPRESENTED AS $\mathrm{F}(J)$, WHERE J IS THE NUMBER OF NEURONS. CONVOLUTIONAL LAYERS ARE REPRESENTED AS $\mathrm{C}(S, F)$ WHERE S IS THE STRIDE FOR DOWNSAMPLING, AND F IS THE NUMBER OF FILTERS. $\mathrm{C}^T(S, F)$ REPRESENTS THE TRANSPOSE CONVOLUTIONAL LAYER WITH THE SAME ARGUMENTS AS THE CONVOLUTIONAL LAYER EXCEPT THAT S IS THE STRIDE FOR UPSAMPLING. THE KERNEL SIZES FOR BOTH THESE LAYERS ARE 4

Component	Architecture
MNIST(E)	$\{\mathrm{F}(200), \mathrm{F}(200)\}$
MNIST(D)	$\{\mathrm{F}(200), \mathrm{F}(200)\}$
MNIST(S)	$\{\mathrm{F}(512), \mathrm{F}(128), \mathrm{F}(64), \mathrm{F}(2)\}$
FMNIST(E)	$\{\mathrm{C}(1,32), \mathrm{C}(2,32), \mathrm{C}(2,64), \mathrm{F}(k)\}$
FMNIST(D)	$\{\mathrm{F}(3136), \mathrm{C}^T(2,32), \mathrm{C}^T(2,32), \mathrm{C}^T(1,1)\}$
FMNIST(S)	$\{\mathrm{C}(1,32), \mathrm{C}(2,32), \mathrm{C}(2,64), \mathrm{F}(512), \mathrm{F}(128), \mathrm{F}(32), \mathrm{F}(2)\}$
CIFAR-10(E)	$\{\mathrm{C}(2,16), \mathrm{C}(2,32), \mathrm{C}(2,64), \mathrm{C}(1,128), \mathrm{F}(k)\}$
CIFAR-10(D)	$\{\mathrm{F}(2048), \mathrm{C}^T(1,128), \mathrm{C}^T(2,64), \mathrm{C}^T(2,32), \mathrm{C}^T(2,16), \mathrm{C}^T(1,3)\}$
CIFAR-10(S)	$\{\mathrm{C}(2,16), \mathrm{C}(2,32), \mathrm{C}(2,64), \mathrm{C}(2,128), \mathrm{F}(128), \mathrm{F}(32), \mathrm{F}(2)\}$

which gives stronger gradients to the selector network. The value of γ is slowly decayed from 1 to 0.01 0.01 as training progresses. However, for $m = 3, k = 2$ this leads to overfitting of the selector network and we the standard $\mathcal{L}_{\text{MoE\_EXP}}$ loss. The learning rate was initialized to 10^{-3} and decayed by 0.97 every 10^5 training iterations for Gaussian and Laplace sources.

Table I lists the structure of the encoder, decoder, and selector networks used for the experiments on MNIST, Fashion MNIST, and CIFAR-10 datasets. The architecture for MNIST and Fashion MNIST used $ReLU(\cdot)$ activation functions. The architecture for the CIFAR-10 employed the GDN activation function proposed by [52]. All the systems were trained with an ADAM optimizer [53] with the default parameters. For images we used a constant learning rate of 10^{-4}. All implementations were done in TensorFlow [54].

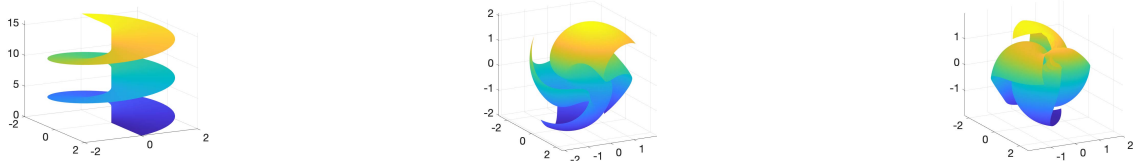
B. Visualizing the Encoders and the Decoders

Fig. 5 showcases the various encoding functions for the $m = 2, k = 1$ system at CSNR $= 30$dB. The two axes of the figures represent the two dimensions of the input samples $\mathbf{x}$. The value of encoding is represented by the color. Fig. 5a is the encoder proposed by [2]. The encoding is a projection onto a spiral followed by a transformation, as explained in Sec. IV. The system achieves a simulated SDR that is 0.96dB less than $\mathrm{SDR}_{\text{opt}}$. Fig. 5b is the encoding solution learned by a single encoder system. The simulated SDR of this setup is 2.86dB away from the Shannon limit. Fig. 5c is the encoding solution learned when the selection between the encoder networks is based on the projection error as described in (22) whose SDR is 1.3dB 1.3dB away from the Shannon limit. Fig. 5d shows the encoder learned by the rule-based multi-encoder system whose SDR is 1.1dB away from the Shannon limit. Fig. 5e shows the encoding solution learned by the self-learned selection based MoE. The SDR of the system is 0.96dB away from the Shannon limit. The takeaway from these figures is four-fold. Firstly, more than one encoder network is crucial for the system performance. Secondly, the selection criterion plays an important role in the type of solution learned. Thirdly, randomly initialized neural networks approach similar solutions as proposed in literature without any constraints guiding or requiring them to do so. Finally, the neural network based systems also learn a projection based encoding. This is confirmed by the constant encoding regions that are spiral in structure.

Fig. 6 shows the various decoding surfaces for $m = 3, k = 2$. As we cannot easily visualize the encoding surface, we visualize the decoding surface. Based on intuition, [13] proposed the helicoid surface for projection an example of which is shown in Fig. 6a. However, both the learned systems in Fig. 6b and Fig. 6c (for CSNR $= 20$dB) show non trivial solutions whose decoding surfaces are not easily parametrized. Thus neural networks can help us find solutions to those systems where human intuition fails. Quantitatively at CSNR $= 30$dB, the rule-based system's SDR is 1.5dB away from the Shannon limit, and the self-learned is 1.71dB away from the Shannon limit, whereas the helicoid system is 2.7dB away from the Shannon limit.

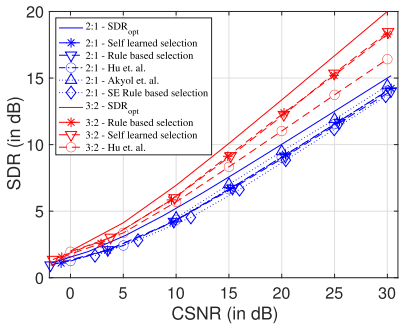
C. Simulation Results

Fig. 7 shows the simulation results of the various learned systems across CSNR values ranging from 0dB to 30dB. We also appropriately plot the comparisons with relevant literature. Fig. 7a shows the performance of the rule-based and the self learned systems for $m = 2, k = 1$ and $m = 3, k = 2$ along with results from [2], [3], and the squared error selection system (22). For, $m = 2, k = 1$, we find that at CSNR $= 30$dB, both the self learned system and [2]

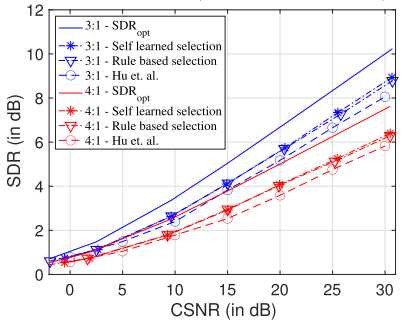


(a) Helicoid decoding surface [13] (b) Rule based decoding surface (c) Self learned decoding surface

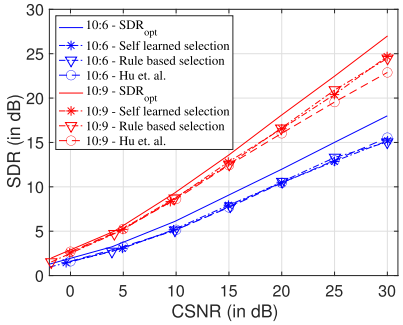
Fig. 6. Decoding surfaces for $m = 3, k = 2$.



(a) Simulations of $m = 2, k = 1$ and $m = 3, k = 2$.



(b) Simulations of $m = 3, k = 1$ and $m = 4, k = 1$.



(c) Simulations of $m = 10, k = 6$ and $m = 10, k = 9$

Fig. 7. Simulation of proposed systems across various CSNRs compared with Akyol et. al. [3] and Hu et. al. [2].

have a simulated SDR, 0.96dB away from the Shannon limit. The system designed by [3] achieves a better 0.5dB gap. We also find that the squared error or projection error based selection system performs worse than both the other learned systems. This is true for all values of m, k, and we do not plot it in the subsequent graphs to avoid cluttering. We find that for $m = 3, k = 2$ both the learned systems drastically outperform the results of [2]. The self-learned system achieves a gap of 1.71dB from the Shannon limit and the rule based a gap of 1.5dB in comparison with the 3.61dB gap of [2]. All the numbers are at CSNR $= 30$dB. Fig. 7b shows the simulation performance of both the self learned and rule based system along with [2]. Both, the learning based systems outperform the parametric curve based system. Additionally, the self learned selection system outperforms the rule based learning system by 0.17dB and 0.11dB at CSNR $= 30$dB for $m = 3, k = 1$ and $m = 4, k = 1$ respectively. Fig. 7c plots the system performance for $m = 10, k = 6$ and $m = 10, k = 9$. For $m = 10, k = 6$ the parametric curve based system, self-learned systems and rule-based systems achieve gaps of 2.46dB, 2.96dB, and 2.84dB respectively from the Shannon limit. For $m = 10, k = 9$, the learned systems significantly outperform the parametric curve based systems. The parametric curve based system, self-learned systems and rule-based systems achieve gaps of 4.11dB, 2.55dB, and 2.30dB respectively from the Shannon limit.

Complexity Analysis: Based on our neural network architecture (two consecutive layers of size $40m$) our time and space complexity of both encoders and decoders are $\mathcal{O}(m^2)$. The training complexity for every iteration is also $\mathcal{O}(m^2)$. Since, there are no implementation details or complexity discussion in [2], we explore a nearest-neighbor based projection method. Using the k-d tree to implement the nearest neighbbor search, we compute the complexity when $k = 1$. Assume that p is the number of points on a grid on a single dimension of source or channel space. The encoding complexity on average is $\mathcal{O}(\log p)$, and the space complexity is $\mathcal{O}(p)$. The decoding complexity is $\mathcal{O}(p^m)$ and the space complexity is $\mathcal{O}(p)$. For the case of $k \neq 1$, the complexity depends on other design factors. For example, a $10 : 6$ system can be implemented using four $2 : 1$ blocks and two $1 : 1$ blocks, or using one $5 : 1$ block and five $1 : 1$ blocks. The training complexity for every iteration of [3] is $\mathcal{O}(p^{m+k})$. The space complexity of the encoder is $\mathcal{O}(p^m)$ and that of the decoder is $\mathcal{O}(p^k)$.

D. Generalization Experiments

The generalization experiments are performed for the self-learned selection methodology since it is the more flexible setup.

1) Robustness to Changes in Noise Power: For the experiments in Sec. V-C, we tested the system at the CSNR at which it was trained. In Fig. 8, we vary the CSNR$_{Te}$, i.e., the testing CSNR to be ± 5dB of the CSNR$_{Tr}$, i.e., the training CSNR. We find that even if the CSNR$_{Te}$ = CSNR$_{Tr} \pm 5$dB the performance of the system is at most 1dB lesser than the system tested at CSNR$_{Tr}$.

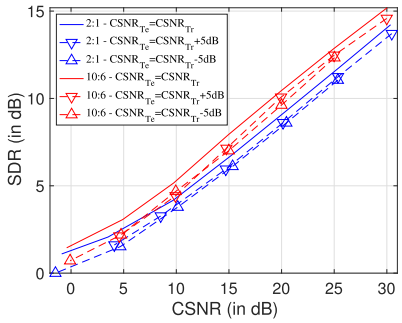


Fig. 8. Robustness of the self-learned $m = 2, k = 1$ and $m = 10, k = 6$ systems with changes in CSNR.

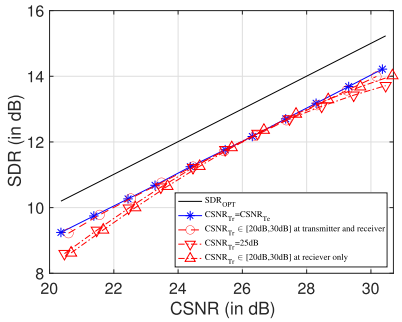


Fig. 9. Simulation results of a single $m = 2, k = 1$ system trained over the range of CSNR $\in [20\text{dB}, 30\text{dB}]$.

2) Training Over a Range of CSNR: Although, as shown in the previous experiment, the learned systems are pretty robust, is it possible to have a single system that can operate over a range of CSNRs when the operating CSNR is known? To answer this, we modified the encoder, decoder, and the selector networks to accept an extra input of CSNR. The size of the neural network was retained to be the same as before. The training was randomized by sampling random CSNR in the range of $[20\text{dB}, 30\text{dB}]$ and training the system for one iteration for that value. We also explored the setup when the CSNR is available only at the receiver. We then compared the system with two benchmarks. The first is where separate systems were trained for every CSNR $= \{20\text{dB}, 21\text{dB}, \ldots, 30\text{dB}\}$. The second is the system trained at CSNR $= 25$dB and tested over the range of CSNR $\in [20\text{dB}, 30\text{dB}]$. We used two metrics for comparison, the average of distance from SDR_{opt} computed at CSNR $= \{20\text{dB}, \ldots, 30\text{dB}\}$ and the maximum distance from SDR_{opt} computed over the same points. The system trained over the range of CSNRs achieved an average gap of 1.03dB and a maximum gap of 1.09dB. The system trained with CSNR available only at the receiver achieved an average gap of 1.22dB and a maximum gap of 1.73dB. The separately trained systems achieved an average gap of 0.97dB and a maximum gap of 0.99dB. The system trained at CSNR $= 25$dB achieved an average gap of 1.22dB and a maximum gap of 1.65dB. The system trained over a range of CSNRs can generalize much better than the system trained at a fixed CSNR. This indicates that the system, provided the CSNR is known, can learn multiple modes of operation specific to each CSNR. However, the CSNR information is transmitter. In comparison, the system with the CSNR available only at the receiver performs poorly. Consider the example of $m = 2, k = 1$ case where the encoders learn a projection map onto some spiral. The operating CSNR decides how tightly wound this spiral is. As the CSNR increases, the arms of the spiral come closer to each other. Without this information at the transmitter, the encoder learns a solution that is best for the average case. This results in degraded performance.

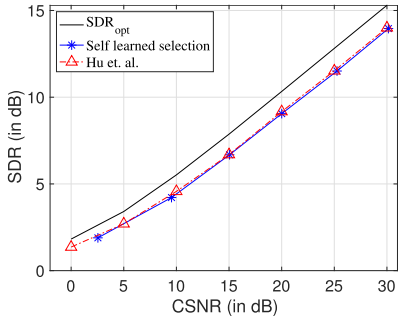


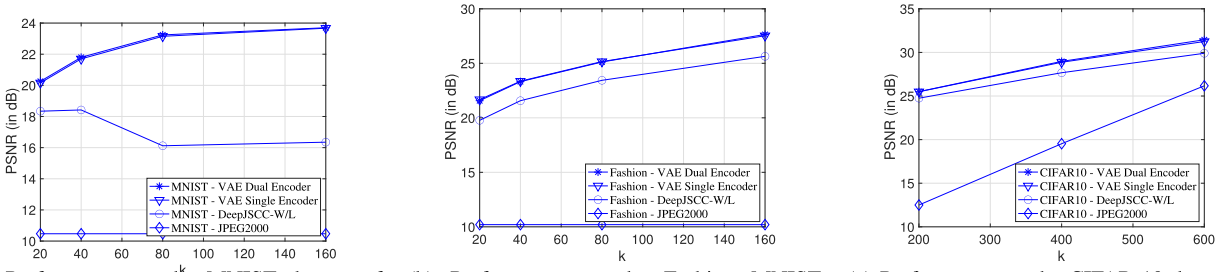
Fig. 10. Simulations of $m = 2, k = 1$ for a Laplace distributed source.

3) Generalization to Other Input Distributions: Laplacian Sources: The system designed in Sec. IV-B can generalize to other source distributions too. In Fig. 10 we show the performance of systems $m = 2, k = 1$ trained for a Laplace distributed source. We also plot the results of [2] for comparison. Our design methodology is not modified in any fashion to accommodate information about the new source distribution. We compare the performance of our system with the solution proposed in [2]. The performance between both the systems is indistinguishable.

E. Joint Source-Channel Coding for Images

Figs. 11 showcases the performance of our JSCC system on images for the MNIST, the Fashion MNIST dataset [55] and the CIFAR-10 dataset [56] for CSNR $= 20$dB. We compared the results against two methods. The first is a baseline of JPEG2000 operating along with a capacity acheiving channel coding and modulation scheme. The target compression ratio is computed as $\frac{n}{kC_{\text{AWGN}}}$. The compression ratios for MNIST and Fashion MNIST datasets correspond to $\{94.2, 47.1, 23.6, 11.8\}$. The compression ratios for CIFAR-10 corresponds to $\{36.9, 18.5, 12.3\}$. For those images where such a compression ratio cannot be acheived the mean value of the image is transmitted. The compression ratios tested here are beyond the capability of JPEG2000, particularly for the MNIST and Fashion MNIST datasets. We also compared our performance with that of DeepJSCC Wireless (DeepJSCC-W/L) which we implmented [20]. PSNR is the metric used for comparison. For, the MNIST dataset we observed that DeepJSCC-W/L had some issues with local optima for $k = \{80, 160\}$. This could be because of the simple architecture considered for that experiment.

We observe that both the single and the dual encoder VAE outperform the JPEG2000 and DeepJSCC-W/L for all the datasets and configurations tested upon. However, the dual



(a) Performance on the MNIST dataset of images for $k = \{20, 40, 80, 160\}$.

(b) Performance on the Fashion MNIST dataset of images for $k = \{20, 40, 80, 160\}$.

(c) Performance on the CIFAR-10 dataset of images for $k = \{200, 400, 600\}$.

Fig. 11. JSCC of images.

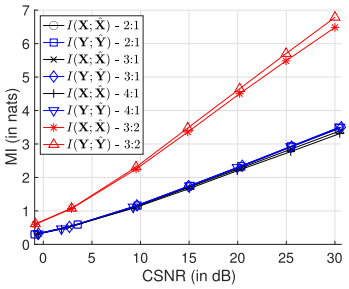


Fig. 12. Plot showcasing the estimated $I(\mathbf{X};\hat{\mathbf{X}})$ and $I(\mathbf{Y};\hat{\mathbf{Y}})$ for $2:1, 3:1, 3:2$, and $4:1$ systems.

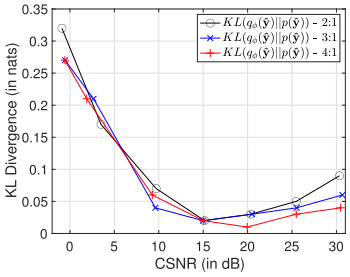


Fig. 13. Plot showcasing the estimated $KL(q_\phi(\hat{\mathbf{y}}) \,||\, p(\hat{\mathbf{y}}))$ for $2:1, 3:1$ and $4:1$ systems.

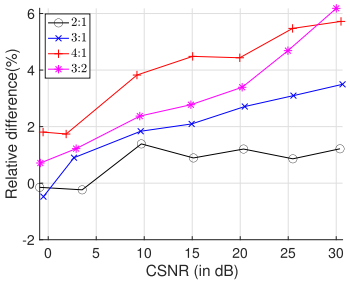


Fig. 14. Plot showcasing $\left(1.0 - \frac{I(\mathbf{X};\hat{\mathbf{X}})}{kC_{\text{AWGN}}}\right) 100.0$ vs CSNR.

encoder does not give any improvement in performance when compared to the single encoder. We hypothesized this to be because of the nature of the dataset and its size. The dual encoder excels when two sufficiently close datapoints have to be projected to two different and possibly distant values. However, the relatively small size of the image datasets, allows the system to interpret the images as relatively distant and isolated from each other, which renders the dual encoder setup unnecessary. To test out this hypothesis we performed an experiment on Gaussian JSCC with both small and large number of training samples, for $m = 10, k = 9$ at $CNSR = 30$dB. We first trained the dual and the single encoder systems with each training batch being sampled as a new set of samples. For reference, the Shannon limit for this configuration is 27.0dB. The dual encoder setup achieved a performance of SDR $= 24.7$dB. The single encoder setup trained achieved SDR $= 22.7$dB, which is 2dB worse than the dual encoder. For the next set of experiments we initialized a common dataset of 50000 samples and trained both the single and dual encoder setup using the common dataset. We found that both the dual and single encoder VAE performed at SDR $= 11.2$dB. Not only is this much worse than the performance when fresh batches of samples were used, the use of limited samples also affects the efficacy of the dual encoder over the single encoder. Overcoming this bottleneck can drastically improve the performance of JSCC for images. This is left as an open future research direction.

F. Empirical Analysis of the Theorem

Here we discuss some of the empirical insights into the tightness of the bound proposed in Theorem 1 along two directions. First, we focus on the two relaxations that cause the VAE ELBO objective to be an upper bound on the Rate-Distortion objective. We show that the quantities $I(\hat{\mathbf{Y}};\mathbf{Y}) - I(\hat{\mathbf{X}};\mathbf{X})$ and $KL(q_\phi(\hat{\mathbf{y}})||p(\hat{\mathbf{y}}))$ are small. We also test our assumption that the system operates close to the channel capacity, whether $I(\hat{\mathbf{X}};\mathbf{X})$ is close to kC? These quantities are computed using the trained systems. Since these quantities are sample estimates of Mutual Information (MI) and KL divergence, we restrict ourselves to analyzing the low-dimensional cases, $k = 1$ or $k = 2$ for Gaussian sources.

1) Mutual Information Gap: Fig. 12 showcases the estimated values of $I(\mathbf{X};\hat{\mathbf{X}})$ and $I(\mathbf{Y};\hat{\mathbf{Y}})$. We estimated the Mutual Information using both a non-parametric nearest neighbor MI estimator [57] and a neural network based method [58]. The largest difference between $I(\mathbf{Y};\hat{\mathbf{Y}})$ and $I(\mathbf{X};\hat{\mathbf{X}})$ was $0.3nats$ at CSNR $= 30$dB for the $3:2$ system.

2) KL Divergence Gap: Fig. 13 showcases the estimated values of $KL(q_\phi(\hat{\mathbf{y}})||p(\hat{\mathbf{y}}))$. The KL divergence is estimated as a Monte-Carlo estimate of $\mathbb{E}_{\hat{\mathbf{y}}\sim q_\phi(\hat{\mathbf{y}})}\left[\log(q_\phi(\hat{\mathbf{y}})) - \log(p(\hat{\mathbf{y}}))\right]$. $\log(p(\hat{\mathbf{y}}))$ is computed

using the model that we fit on $\hat{\mathbf{y}}$ during training. We estimate $\log(q_\phi(\hat{\mathbf{y}}))$ as $\log\left(\mathbb{E}_{\mathbf{y}\sim g_e(X)}\left[p_{\mathbf{Z}}(\hat{\mathbf{y}} \mid \mathbf{y})\right]\right)$ where, $p_{\mathbf{Z}}(\hat{\mathbf{y}}|\mathbf{y})$ represents the channel model. Since the number of samples required for a reliable estimate of these quantities increases exponentially with the dimension of $\mathbf{Y}$, we restrict our computation to the $k = 1$ case. We find that for $\mathrm{CSNR} \geq 10$dB the KL divergence is $\leq 0.1 nats$ for all the systems. For training the $\mathrm{CSNR} \leq 5$dB systems the use of the extra regularization term (33) to force the system away from trivial solutions, interferes with the distribution of the samples $\mathbf{y}$, which in turn leads to a poor fit.

3) Capacity Gap: Fig. 14 showcases the relative difference between total channel capacity and $I(\mathbf{X};\hat{\mathbf{X}})$ i.e. $\left(1.0 - \frac{I(\mathbf{X};\hat{\mathbf{X}})}{kC_{\mathrm{AWGN}}}\right) \times 100.0$. This is to test the first assumption of the Theorem 1. We find that the largest relative difference is 6.18% at $\mathrm{CSNR} = 30$dB for the $3:2$ system, which is reasonably small.

VI. Conclusion

In this paper, we proposed a scheme for Joint Source-Channel Coding over analog additive noise channels that leverages the knowledge of Variational Autoencoders. Specifically, we showed that the minimization objective obtained from rate-distortion theory is upper bounded by the objective used to train the VAEs for JSCC. When the source dimension is greater than the channel dimension, two source samples that are close to each other can be encoded in such a way that in the encoding space they are not in each other's neighborhood. To account for such projections, we proposed a Mixture-of-Encoders setup where for each sample one of the multiple encoders is selected for transmission. We proposed two selection methodologies, one based on an intuitive rule and another where the selection criterion is learned. The simulated performance of both these methods for JSCC of Gaussian sources over AWGN channels showed that the learned systems perform close to or better than existing methods. Further, these learned systems are robust to changes in channel conditions. Finally, we showed that the proposed methodology generalizes to other source distributions like the Laplacian and more realistic setups like images.

References

[1] C. E. Shannon, "Communication in the presence of noise," *Proc. Inst. Radio Eng.*, vol. 37, no. 1, pp. 10–21, Jan. 1949.

[2] Y. Hu, J. Garcia-Frias, and M. Lamarca, "Analog joint source-channel coding using non-linear curves and MMSE decoding," *IEEE Trans. Commun.*, vol. 59, no. 11, pp. 3016–3026, Nov. 2011.

[3] E. Akyol, K. B. Viswanatha, K. Rose, and T. A. Ramstad, "On zero-delay source-channel coding," *IEEE Trans. Inf. Theory*, vol. 60, no. 12, pp. 7473–7489, Dec. 2014.

[4] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul./Oct. 1948.

[5] T. Goblick, "Theoretical limitations on the transmission of data from analog sources," *IEEE Trans. Inf. Theory*, vol. IT-11, no. 4, pp. 558–567, Oct. 1965.

[6] K.-H. Lee and D. Petersen, "Optimal linear coding for vector channels," *IEEE Trans. Commun.*, vol. COM-24, no. 12, pp. 1283–1290, Dec. 1976.

[7] T. Basar, B. Sankur, and H. Abut, "Performance bounds and optimal linear coding for discrete-time multichannel communication systems," *IEEE Trans. Inf. Theory*, vol. IT-26, no. 2, pp. 212–217, Mar. 1980.

[8] V. A. Vaishampayan, "Combined source-channel coding for bandlimited waveform channels," Ph.D. dissertation, Dept. Elect. Eng., Univ. Maryland, Heights, MD, USA, 1989.

[9] V. A. Vaishampayan and N. Farvardin, "Joint design of block source codes and modulation signal sets," *IEEE Trans. Inf. Theory*, vol. 38, no. 4, pp. 1230–1248, Jul. 1992.

[10] A. Fuldseth and T. A. Ramstad, "Bandwidth compression for continuous amplitude channels based on vector approximation to a continuous subset of the source signal space," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Oct. 1997, pp. 3093–3096.

[11] T. A. Ramstad, "Shannon mappings for robust communication," *Telektronikk*, vol. 98, no. 1, pp. 114–128, 2002.

[12] F. Hekland, G. E. Oien, and T. A. Ramstad, "Using 2:1 Shannon mapping for joint source-channel coding," in *Proc. Data Compress. Conf.*, 2005, pp. 223–232.

[13] P. Floor and T. Ramstad, "Dimension reducing mappings in joint source-channel coding," in *Proc. 7th Nordic Signal Process. Symp.*, Jun. 2006, pp. 282–285.

[14] P. A. Floor, "On the design and analysis of Shannon-Kotel'nikov mappings for joint-source-channel coding," Ph.D. dissertation, Dept. Electron. Telecommun., Norwegian Univ. Sci. Technol., Gjøvik, Norway, May 2007.

[15] F. Hekland, P. A. Floor, and T. A. Ramstad, "Shannon-kotel-nikov mappings in joint source-channel coding," *IEEE Trans. Commun.*, vol. 57, no. 1, pp. 94–105, Jan. 2009.

[16] P. A. Floor, T. A. Ramstad, and N. Wernersson, "Power constrained channel optimized vector quantizers used for bandwidth expansion," in *Proc. 4th Int. Symp. Wireless Commun. Syst.*, Oct. 2007, pp. 667–671.

[17] Y. M. Saidutta, A. Abdi, and F. Fekri, "M to 1 joint source-channel coding of Gaussian sources via dichotomy of the input space based on deep learning," in *Proc. Data Compress. Conf. (DCC)*, Mar. 2019, pp. 487–497.

[18] Y. M. Saidutta, A. Abdi, and F. Fekri, "Joint source-channel coding for Gaussian sources over AWGN channels using variational autoencoders," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Jul. 2019, pp. 1327–1331.

[19] Y. M. Saidutta, A. Abdi, and F. Fekri, "Joint source-channel coding of Gaussian sources over AWGN channels via manifold variational autoencoders," in *Proc. 57th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2019, pp. 514–520.

[20] E. Bourtsoulatze, D. Burth Kurka, and D. Gunduz, "Deep joint source-channel coding for wireless image transmission," 2018, *arXiv:1809.01733*. [Online]. Available: http://arxiv.org/abs/1809.01733

[21] M. Jankowski, D. Gunduz, and K. Mikolajczyk, "Deep joint source-channel coding for wireless image retrieval," in *Proc. ICASSP - IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 5070–5074.

[22] D. B. Kurka and D. Gunduz, "Deep joint source-channel coding of images with feedback," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 5235–5239.

[23] D. B. Kurka and D. Gunduz, "DeepJSCC-f: Deep joint source-channel coding of images with feedback," *IEEE J. Sel. Areas Inf. Theory*, vol. 1, no. 1, pp. 178–193, Dec. 2020.

[24] N. Farsad, M. Rao, and A. Goldsmith, "Deep learning for joint source-channel coding of text," in *Proc. IEEE ICASSP*, Dec. 2018, pp. 2326–2330.

[25] T. J. O'Shea, K. Karra, and T. C. Clancy, "Learning to communicate: Channel auto-encoders, domain specific regularizers, and attention," in *Proc. IEEE Int. Symp. Signal Process. Inf. Technol. (ISSPIT)*, Dec. 2016, pp. 223–228.

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

[27] T. J. O'Shea, T. Erpek, and T. Charles Clancy, "Deep learning based MIMO communications," 2017, *arXiv:1707.07980*. [Online]. Available: http://arxiv.org/abs/1707.07980

[28] A. Felix, S. Cammerer, S. Dörner, J. Hoydis, and S. ten Brink, "OFDM-autoencoder for End-to-End learning of communications systems," 2018, *arXiv:1803.05815*. [Online]. Available: http://arxiv.org/abs/1803.05815

[29] Y. Liao, N. Farsad, N. Shlezinger, Y. C. Eldar, and A. J. Goldsmith, "Deep neural network symbol detecftion for millimeter wave communications," 2019, *arXiv:1907.11294*. [Online]. Available: http://arxiv.org/abs/1907.11294

[30] P. Yang, Y. Xiao, M. Xiao, Y. L. Guan, S. Li, and W. Xiang, "Adaptive spatial modulation MIMO based on machine learning," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 9, pp. 2117–2131, Sep. 2019.

[31] E. Nachmani, Y. Be'ery, and D. Burshtein, "Learning to decode linear codes using deep learning," in *Proc. 54th Annu. Allerton Conf. Commun., Control, Comput. (Allerton)*, Sep. 2016, pp. 341–346.

[32] E. Nachmani, E. Marciano, D. Burshtein, and Y. Be'ery, "RNN decoding of linear block codes," 2017, *arXiv:1702.07560*. [Online]. Available: http://arxiv.org/abs/1702.07560
[33] E. Nachmani, E. Marciano, L. Lugosch, W. J. Gross, D. Burshtein, and Y. Be'ery, "Deep learning methods for improved decoding of linear codes," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 119–131, Feb. 2018.
[34] L. Lugosch and W. J. Gross, "Neural offset min-sum decoding," in *Proc. IEEE Int. Symp. Inf. Theory (ISIT)*, Oct. 2017, pp. 1361–1365.
[35] H. Kim, Y. Jiang, S. Kannan, S. Oh, and P. Viswanath, "Deepcode: Feedback codes via deep learning," 2018, *arXiv:1807.00801*. [Online]. Available: http://arxiv.org/abs/1807.00801
[36] K. Choi, K. Tatwawadi, T. Weissman, and S. Ermon, "NECST: Neural joint source-channel coding," *CoRR*, vol. abs/1811.07557, pp. 1–4, Oct. 2018.
[37] D. P Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114
[38] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, "Ladder variational autoencoders," in *Adv. neural Inf. Process. Syst.*, 2016, pp. 3738–3746.
[39] W. Xu, H. Sun, C. Deng, and Y. Tan, "Variational autoencoder for semi-supervised text classification," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
[40] A. Srivastava and C. Sutton, "Autoencoding variational inference for topic models," 2017, *arXiv:1703.01488*. [Online]. Available: http://arxiv.org/abs/1703.01488
[41] I. Higgins *et al.*, "BETA-VAE: Learning basic visual concepts with a constrained variational framework," in *Proc. ICLR*, 2017.
[42] H. Kim and A. Mnih, "Disentangling by factorising," 2018, *arXiv:1802.05983*. [Online]. Available: http://arxiv.org/abs/1802.05983
[43] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Comput.*, vol. 3, no. 1, pp. 79–87, Feb. 1991.
[44] K. Chen, L. Xu, and H. Chi, "Improved learning algorithms for mixture of experts in multiclass classification," *Neural Netw.*, vol. 12, no. 9, pp. 1229–1252, Nov. 1999.
[45] L. Xu, M. I. Jordan, and G. E. Hinton, "An alternative model for mixtures of experts," in *Adv. neural Inf. Process. Syst.*, vol. 1995, pp. 633–640.
[46] A. Makkuva, P. Viswanath, S. Kannan, and S. Oh, "Breaking the gridlock in mixture-of-experts: Consistent and efficient algorithms," in *Int. Conf. Mach. Learn.*, vol. 2019, pp. 4304–4313.
[47] M. Versace, R. Bhatt, O. Hinds, and M. Shiffer, "Predicting the exchange traded fund DIA with a combination of genetic algorithms and neural networks," *Expert Syst. Appl.*, vol. 27, no. 3, pp. 417–425, Oct. 2004.
[48] T. M. Cover and J. A. Thomas, *Elements Information Theory*. Hoboken, NJ, USA: Wiley, 2012.
[49] R. C. Robbins. (2006). *Economic Applications of Lagrange Multipliers*. [Online]. Available: https://sites.math.northwestern.edu/clark/285/2006-07/handouts/lagrange-econ.pdf
[50] W. Hurewicz and H. Wallman, *Dimension Theory (PMS-4)*, vol. 4. Princeton, NJ, USA: Princeton Univ. Press, 2015.
[51] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2015, pp. 1026–1034.
[52] C. Cai, L. Chen, X. Zhang, and Z. Gao, "End-to-end optimized ROI image compression," *IEEE Trans. Image Process.*, vol. 29, pp. 3442–3457, 2020.
[53] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[54] M. Abadi *et al.*, "TensorFlow: A system for large-scale machine learning," in *Proc. OSDI*, vol. 16, 2016, pp. 265–283.
[55] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*. [Online]. Available: http://arxiv.org/abs/1708.07747
[56] A. Krizhevsky *et al.*, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. 78, 2009. [Online]. Available: https://scholar.google.com/scholar?hl=en&as_sdt=0,44&cluster=2409602007304826 95
[57] G. Ver Steeg. (2000). *Non-Parametric Entropy Estimation Toolbox (Npeet)*. [Online]. Available: https://github.com/gregversteeg/NPEET
[58] M. I. Belghazi *et al.*, "Mutual information neural estimation," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 531–540.

Yashas Malur Saidutta (Student Member, IEEE) received the B.Tech. degree from the National Institute of Technology Karnataka, India. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the Georgia Institute of Technology, with a focus on developing machine learning methods for applications in communication systems. His research interests include Bayesian optimization and reinforcement learning.

Afshin Abdi (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the Georgia Institute of Technology in 2020. He is currently a Research Engineer with Qualcomm Technologies, Inc., working on graph neural networks and applications of deep learning in coding and communication. His research encompassed the areas of distributed machine learning and applications of machine learning in coding, communication, and biological signal processing.

Faramarz Fekri (Fellow, IEEE) received the Ph.D. degree from the Georgia Institute of Technology, Atlanta, GA, USA, in 2000. Since 2000, he has been a Faculty Member with the School of Electrical and Computer Engineering, Georgia Institute of Technology, where he is currently a Professor and a GTRI Fellow. His research interests include machine learning, signal processing, source and channel coding, information theory in biology, statistical inference in large data, and inductive logic programming. He received the Sony Faculty Innovation Award, the National Science Foundation Career Award, the Southern Center for Electrical Engineering Education (SCEEE) Young Faculty Development Award, and the Outstanding Young Faculty Award of the School of Electrical and Computer Engineering. He also serves on the Technical Program Committees for several IEEE conferences. He served on the Editorial Board for the IEEE TRANSACTIONS ON COMMUNICATIONS and the *Physical Communication* (PHYCOM) (Elsevier) Journal. He is also an Associate Editor of the IEEE TRANSACTIONS ON MOLECULAR, BIOLOGICAL, AND MULTI-SCALE COMMUNICATIONS.