DeepJSCC-Q: Constellation Constrained Deep Joint Source-Channel Coding

Tze-Yang Tung, David Burth Kurka, Mikolaj Jankowski, Deniz Gündüz
Department of Electrical and Electronics Engineering
Imperial College London

***Abstract*—Recent works have shown that modern machine learning techniques can provide an alternative approach to the long-standing joint source-channel coding (JSCC) problem. Very promising initial results, superior to popular digital schemes that utilize separate source and channel codes, have been demonstrated for wireless image and video transmission using deep neural networks (DNNs). However, end-to-end training of such schemes requires a differentiable channel input representation; hence, prior works have assumed that any complex value can be transmitted over the channel. This can prevent the application of these codes in scenarios where the hardware or protocol can only admit certain sets of channel inputs, prescribed by a digital constellation. Herein, we propose *DeepJSCC-Q*, an end-to-end optimized JSCC solution for wireless image transmission using a finite channel input alphabet. We show that *DeepJSCC-Q* can achieve similar performance to prior works that allow any complex valued channel input, especially when high modulation orders are available, and that the performance asymptotically approaches that of unconstrained channel input as the modulation order increases. Importantly, *DeepJSCC-Q* preserves the graceful degradation of image quality in unpredictable channel conditions, a desirable property for deployment in mobile systems with rapidly changing channel conditions.**

***Index Terms*—Joint source-channel coding, wireless image transmission, image compression, deep neural networks, deep learning**

I. Introduction

Source coding and channel coding are two essential steps in modern data transmission. The former reduces the redundancy within the source signal, preserving essential information needed to reconstruct the signal within a certain fidelity. For example, in image transmission, commonly used compression schemes, such as JPEG and BPG, allow for the reduction in communication load with minimal loss in reconstruction quality. Channel coding, on the other hand, introduces structured redundancy to allow reliable decoding in the presence of channel imperfections. A diagram of a typical communication system employing separate source and channel coding is shown in Fig. 1.

This work was supported by the European Research Council (ERC) through project BEACON (No. 677854), and by CHIST-ERA grants CHIST-ERA-18-SDCDN- 001 (funded by EPSRC-EP/T023600/1) and CHIST-ERA-20-SICT-004 (funded by EPSRC-EP/W035960/1).

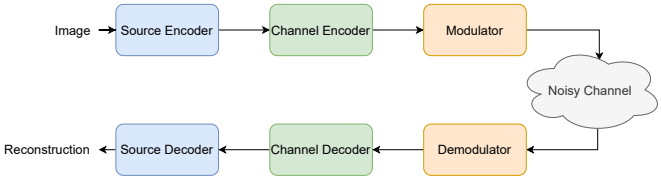


Fig. 1. Diagram of a typical separation-based digital image communication system.

It was proven by Shannon that the separation of source and channel coding is without loss of optimality when the blocklength goes to infinity [1]. Nevertheless, in practical applications we are limited to finite blocklengths, and it is known that combining the two coding steps, that is, joint source-channel coding (JSCC), can achieve lower distortion for a given finite blocklength than separate source and channel coding [2]–[4]. The most straightforward approach to JSCC is to optimize the various parameters between source coding, channel coding and modulation in a cross-layer framework. Although many such schemes have been proposed over the years [5]–[7], none were able to provide sufficient gains to justify the increased complexity.

A more fundamental approach is to redesign a JSCC scheme from scratch, directly mapping the source signal to the modulated channel input, without conversion to bits at all. It was shown recently in [8], that deep neural networks (DNNs) can be used to break the complexity barrier of designing JSCC schemes for wireless image transmission. The scheme, called *DeepJSCC*, showed appealing properties, such as lower end-to-end distortion for a given channel blocklength compared with state-of-the-art digital compression schemes [9]–[11], flexibility to adapt to different source or channel models [8], [9], ability to exploit channel feedback [12], and capability to produce adaptive-bandwidth transmission schemes [13]. Importantly, graceful degradation of image quality with respect to decreasing channel quality means that DeepJSCC is able to avoid the *cliff-effect* that all separation-based schemes suffer from; which refers to the phenomenon where the image becomes un-decodable when the channel quality falls below a certain threshold resulting in unreliable transmission.

Another strength of DeepJSCC is that it learns a communication scheme from scratch, optimizing all transformations in a data-driven manner using autoencoders

[14] with a non-trainable differentiable channel model in the bottleneck layer. This simplifies the JSCC design procedure, and allows adaptation to any particular source or channel domain and quality measure. Part of that simplification stems from the fact that DeepJSCC not only combines source and channel coding into one single mapping, but it also removes the constellation diagrams used in digital schemes. In digital communications, channel encoded bits are mapped to the elements of a two-dimensional finite constellation diagram, such as quadrature amplitude modulation (QAM), phase shift keying (PSK), or amplitude shift keying (ASK). In contrast, in DeepJSCC, the encoder can transmit arbitrary complex-valued channel symbols, within a power constraint. Although transmission of physical signals will ultimately be analog, regardless of whether digital or DeepJSCC coding schemes are used, commercially available hardware will typically implement a particular communication standard, such as Bluetooth or Wifi, which includes pre-defined constellation order and designs. For example, the IEEE 802.11ad physical layer (PHY) specification [15] states the use of BPSK, QPSK, 16-QAM and 64-QAM. Therefore, in practice, DeepJSCC can be difficult to implement without custom hardware.

In this work, we investigate the effects of constraining the transmission either to a limited number of channel input symbols, or to a predefined constellation imposed externally. This constraint can be crucial for the adoption of DeepJSCC in commercially available hardware (e.g., radio transmitters), where modulators are hard-coded for efficiency, and limit the output space available to the encoder. For example, the IEEE 802.15.4 [16] PHY only specifies the use of offset quadrature phase shift keying (OQPSK) in its PHY specification, without channel coding. This means, DeepJSCC schemes can use hardware implementing this standard if it were trained with a OQPSK constellation. Successfully incorporating fixed channel input constellations in DNN driven JSCC can drive the adoption of such schemes in practice. Therefore, in this paper we introduce a new strategy for JSCC of images, called *DeepJSCC-Q*, which allows for the transmission of the content through fixed pre-defined constellations.

We note that the problem at hand is a JSCC problem over a discrete-input additive white Gaussian noise (AWGN) channel. For any given finite blocklength, we have a finite number of codewords that can be transmitted. Hence, the goal is to find the mapping from the input images to these codewords and a matching decoder mapping that minimizes the average end-to-end distortion. However, finding these mappings directly is a formidable challenge. We formulate this problem as an end-to-end JSCC problem with a quantizer in the middle. An encoder DNN extracts the features of the input image, which are then quantized to the constellation points. These constellation points are transmitted over the channel and the receiver tries to recover the input image from its noisy observations using another DNN. Hence, the problem becomes the training of an autoencoder architecture with a non-differentiable quantization layer followed by a non-trainable channel layer in the middle.

The main contributions of this paper are:

- We propose a DeepJSCC scheme for wireless image transmission that utilizes a discrete channel input constellation, called *DeepJSCC-Q*.
- We show that, even with this constraint, it can achieve superior performance compared to separate source and channel coding using better portable graphics (BPG) codec [17] followed by low density parity check (LDPC) codes [18].
- Unlike separate source and channel coding, *DeepJSCC-Q* does not suffer from the *cliff-effect* even though both schemes use the same constellations.
- We also show that given a large enough constellation, *DeepJSCC-Q* achieves performance close to that is achieved by the unconstrained DeepJSCC scheme [8]. This emphasizes a main difference between DeepJSCC and digital transmission, where a larger constellation implies a higher communication rate.
- Finally, we show that new constellations can be learned for a given modulation order that outperforms conventional constellation designs.

II. Related Works

JSCC of images for wireless transmission has received numerous attention over the years. Earliest communication systems were purely analog, where the source signal is directly modulated unto the carrier waveform, corresponding to a most direct form of JSCC. With the advances in digital communication and compression techniques, separation based communication systems became dominant. In order to overcome the limitations of the separation approach, many works focus on optimizing the various parameters of the employed source and channel codes with the goal of minimizing the end-to-end distortion. For example, in [6], a general framework for matching source and channel code rates using a parametric distortion model was proposed. Their approach is to match the source code rate to the channel code and channel statistics in a source-rate-based optimization approach. Similarly, in [5], a cross-layer optimization of the source code rate, channel code rate and transmitter power for quality of service (QoS) is proposed.

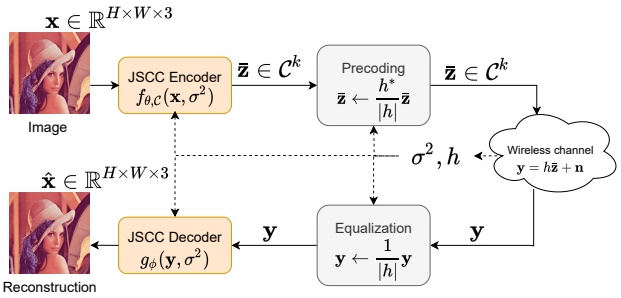
A slightly different approach considers unequal error protection (UEP) to achieve reliability despite channel uncertainty. This approach typically separates the source into a base layer and potentially multiple enhancement layers, with the base layer given the greatest amount of error protection to ensure a baseline reconstruction quality. In [7], the source image is split into multiple layers in the discrete wavelet transform (DWT) domain before a quantizer and channel code is applied. The bit rate allocation between the quantizer and channel code of each layer is optimized using an end-to-end rate distortion model. Similarly, in [19], turbo codes [20] and Reed-Solomon codes [21] are used to achieve UEP. In

[22], instead of using different channel codes, the authors propose hierarchical modulation to achieve UEP. However, none of these schemes were able to achieve adequate gains to justify the increase in system complexity stemming from the joint optimization of the parameters of multiple codes and the successive decoding needed at the receiver.

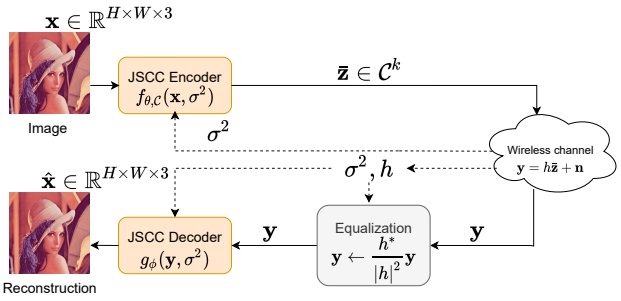
A more fundamental approach is to design the communication system from scratch, without considering any digital interface. Different from the pure analog modulation schemes, source signal is sampled and transmitted using modulated pulses. However, mapping the source samples to channel inputs is in general a difficult multi-dimensional optimization problem. Recently, it was shown that DNNs can be used to break the complexity barrier of designing JSCC for wireless image transmission [8]. By setting up the encoder and decoder in an autoencoder configuration with a non-trainable channel layer in between, and by using the mean-squared error (MSE) metric as the loss function between the input and the output, the DNN encoder was able to learn a function which maps input images to channel inputs, and vice versa at the decoder. They demonstrated that the resultant JSCC encoder and decoder, called *DeepJSCC*, was able to surpass the performance of JPEG2000 [23] compression followed by LDPC codes [18] for channel coding. Importantly, they also showed that such schemes can avoid the *cliff-effect* exhibited by all separation-based schemes, which is when the channel quality deteriorates below the minimum channel quality to allow successful decoding of the deployed channel code, leading to a cliff-edge drop-off in the end-to-end performance. Since then, various works have extended this result to further demonstrate the ability to exploit channel feedback [12] and adapt to various bandwidth requirements without retraining [13]. In [24], the viability of DeepJSCC in an orthogonal frequency division multiplexing (OFDM) system was shown, and in [25] it was shown that such schemes can be extended to multi-user scenarios using the same decoder. DNN-aided JSCC for wireless video transmission is studied in [26].

However, an implicit assumption in all of the works above is the ability for the communication hardware to transmit arbitrary complex valued channel inputs. This may not be true as many commercially available hardware have hard-coded standard protocols, making these methods less viable for real world deployment. As such, in [27], an image transmission problem is considered over a discrete input channel, where the input representation is learned using a variational autoencoder (VAE) by assuming a Bernoulli prior. They consider the transmission of MNIST images [28] over a binary erasure channel (BEC) and showed that their scheme performs better than a VAE that only performs compression paired with an LDPC code. An extension to this work [29] improved the results by using an adversarial loss function. In contrast to these works, we investigate the transmission of natural images over a differentiable channel model with a finite channel input alphabet.

The optimization of the input constellation of a digital communication system has also been a long standing research challenge in information and communication theory [30]–[34]. From a channel coding perspective, the goal is to maximize the mutual information between the channel input and output over the distribution of the channel input alphabet. This is inline with Shannon's theorem, which states that the capacity of the channel is defined over the channel input distribution. Although these methods tend to use hand-crafted designs in the past, more recently, DNNs have also been used to carry out constellation learning. For example, in [35], the authors consider the transmission of a uniform binary source using only a fixed set of channel input alphabet, and show that the probability distribution of the constellation points can be learned by optimizing the bit error rate for a given channel signal-to-noise ratio (SNR). They also showed that the constellation points themselves can be part of the trainable parameters, resulting in even better performance. Herein, we are concerned with optimizing the channel input constellation and distribution for natural image sources, rather than the transmission of uniform binary data sequences.



(a) Scenario 1: both the transmitter and receiver have full CSI knowledge.



(b) Scenario 2: the transmitter knows the noise power while the receiver has full CSI knowledge.

Fig. 2. Overview of problem definition for both CSI scenarios.

III. Problem Statement

We consider the problem of wireless image transmission over a noisy channel, in which communication is performed by transmitting one out of a finite set of symbols at each channel use. An input image $\mathbf{x} \in \{0, ..., 255\}^{H\times W\times C}$ (where H, W and C represent the image's height, width and color channels, respectively) is mapped with an en-

coder function $f : \{0,...,255\}^{H\times W\times C} \mapsto \mathcal{C}^k$, where $\mathcal{C} = \{c_1,...,c_M\} \subset \mathbb{C}$ is the channel input alphabet with $|\mathcal{C}| = M$. We impose an average transmit power constraint $\bar{P}$, such that

$$\frac{1}{k}\sum_{i=1}^{k}|\bar{z}_i|^2 \leq \bar{P}, \tag{1}$$

where $\bar{z}_i$ is the ith element of vector $\bar{\mathbf{z}}$. The channel input vector $\bar{\mathbf{z}}$ is transmitted through a noisy channel, with the transfer function $\mathbf{y} = \Upsilon(\bar{\mathbf{z}}) = h\bar{\mathbf{z}} + \mathbf{n}$, where $h \in \mathbb{C}$ is the channel gain and $\mathbf{n} \sim CN(0, \sigma^2\mathbf{I}_{k\times k})$ is a complex Gaussian vector with dimensionality k. Finally, a receiver passes the channel output through a decoder function $g : \mathbb{C}^k \mapsto \{0,...,255\}^{H\times W\times C}$ to produce a reconstruction of the input $\hat{\mathbf{x}} = g(\mathbf{y})$.

We consider both the static and fading channel scenarios. In the former, h is constant, and it is known by both the transmitter and the receiver, corresponding to an AWGN channel. In the case of a fading channel, we assume that h takes on an independent value during the transmission of each image, and its realization is known only by the receiver. Since the channel state information (CSI) is known by the receiver, it can perform channel equalization as

$$\mathbf{y} \leftarrow \frac{h^*}{|h|^2}\mathbf{y}, \tag{2}$$

where h^* is the complex conjugate of h. The equalized symbols are then passed to the decoder for decoding. If the transmitter also has knowledge of h, as in the case of a static channel, then the transmitter can perform precoding, such that

$$\bar{\mathbf{z}} \leftarrow \frac{h^*}{|h|}\bar{\mathbf{z}}. \tag{3}$$

After channel equalization at the receiver, we equivalently obtain an AWGN channel with signal-to-noise ratio (SNR)

$$\text{SNR} = 10\log_{10}\left(\frac{|h|^2\bar{P}}{\sigma^2}\right)dB. \tag{4}$$

For the fading scenario, the average channel SNR is given by

$$\text{SNR} = 10\log_{10}\left(\frac{\mathbb{E}[|h|^2]\bar{P}}{\sigma^2}\right)dB. \tag{5}$$

A diagram illustrating both scenarios is shown in Fig. 2. The *bandwidth compression ratio* is defined as

$$\rho = \frac{k}{H\times W\times C} \text{ channel symbols/pixel}, \tag{6}$$

which measures how much compression we apply to the images, with smaller number reflecting more compression.

To measure the reconstruction quality, we use two metrics: peak signal-to-noise ratio (PSNR) and multi-scale structural similarity index (MS-SSIM). They are defined as

$$\text{PSNR}(\mathbf{x},\hat{\mathbf{x}}) = 10\log_{10}\left(\frac{A^2}{\text{MSE}(\mathbf{x},\hat{\mathbf{x}})}\right)\text{dB}, \tag{7}$$

where $\text{MSE}(\mathbf{x},\hat{\mathbf{x}}) = ||\mathbf{x}-\hat{\mathbf{x}}||_2^2$ and A is the maximum possible value for a given pixel. For a 24-bit RGB pixel, $A = 255$.

The multi-scale structural similarity index (MS-SSIM) is defined as:

$$\text{MS-SSIM}(\mathbf{x},\hat{\mathbf{x}}) = [l_M(\mathbf{x},\hat{\mathbf{x}})]^{\alpha_M}\prod_{j=1}^{M}[a_j(\mathbf{x},\hat{\mathbf{x}})]^{\beta_j}[b_j(\mathbf{x},\hat{\mathbf{x}})]^{\gamma_j}, \tag{8}$$

where

$$l_M(\mathbf{x},\hat{\mathbf{x}}) = \frac{2\mu_{\mathbf{x}}\mu_{\hat{\mathbf{x}}}+v_1}{\mu_{\mathbf{x}}^2+\mu_{\hat{\mathbf{x}}}^2+v_1}, \tag{9}$$

$$a_j(\mathbf{x},\hat{\mathbf{x}}) = \frac{2\sigma_{\mathbf{x}}\sigma_{\hat{\mathbf{x}}}+v_2}{\sigma_{\mathbf{x}}^2+\sigma_{\hat{\mathbf{x}}}^2+v_2}, \tag{10}$$

$$b_j(\mathbf{x},\hat{\mathbf{x}}) = \frac{\sigma_{\mathbf{x}\hat{\mathbf{x}}}+v_3}{\sigma_{\mathbf{x}}\sigma_{\hat{\mathbf{x}}}+v_3}, \tag{11}$$

$\mu_{\mathbf{x}}$, $\sigma_{\mathbf{x}}^2$, $\sigma_{\mathbf{x}\hat{\mathbf{x}}}^2$ are the mean and variance of $\mathbf{x}$, and the covariance between $\mathbf{x}$ and $\hat{\mathbf{x}}$, respectively. v_1, v_2, and v_3 are coefficients for numeric stability; α_M, β_j, and γ_j are the weights for each of the components. Each $a_j(\cdot,\cdot)$ and $b_j(\cdot,\cdot)$ is computed at a different downsampled scale of $\mathbf{x}$ and $\hat{\mathbf{x}}$. We use the default parameter values of (α_M, β_j, γ_j) provided by the original paper [36]. MS-SSIM has been shown to perform better in approximating the human visual perception than the more simplistic structural similarity index (SSIM) on different subjective image and video databases.

The overall goal of our design is to characterize the encoding and decoding functions f and g that maximize the average reconstructed image quality, measured by either Eqn. (7) or (8), between the input image $\mathbf{x}$ and its reconstruction at the decoder $\hat{\mathbf{x}}$, under the given constraints on the available bandwidth ratio ρ, average power P, and constellation $\mathcal{C}$.

IV. Proposed Solution

Herein, we propose *DeepJSCC-Q*, a DNN-based JSCC scheme and an end-to-end training strategy. In *DeepJSCC-Q* we model the encoder and decoder functions as DNNs parameterized by $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, respectively, and aim at learning the optimal parameters through training. Rather than constraining the encoder DNN to discrete outputs, which would require a huge output space, we will allow any output vector of dimension k, and employ a "quantization" layer to map the generated latent vectors to transmitted symbols, such that each quantization level represents a point in the constellation. We will introduce two quantization strategies, one where the constellation $\mathcal{C}$ is fixed, and another, where the constellation is also part of the parameters to be trained for a given constellation order M. By making the constellation part of the trainable parameters, we can also optimize the channel input geometry. As such, we separate the encoder f into two stages: first a DNN function $f_{\boldsymbol{\theta}} : \{0,...,255\}^{H\times W\times C} \mapsto \mathbb{C}^k$ maps an input image $\mathbf{x}$ to a complex latent representation $\mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{x})$ before a quantizer $q_{\mathcal{C}} : \mathbb{C}^k \mapsto \mathcal{C}^k$ maps the latent vector $\mathbf{z}$ to the channel input $\bar{\mathbf{z}} = q_{\mathcal{C}}(\mathbf{z})$.

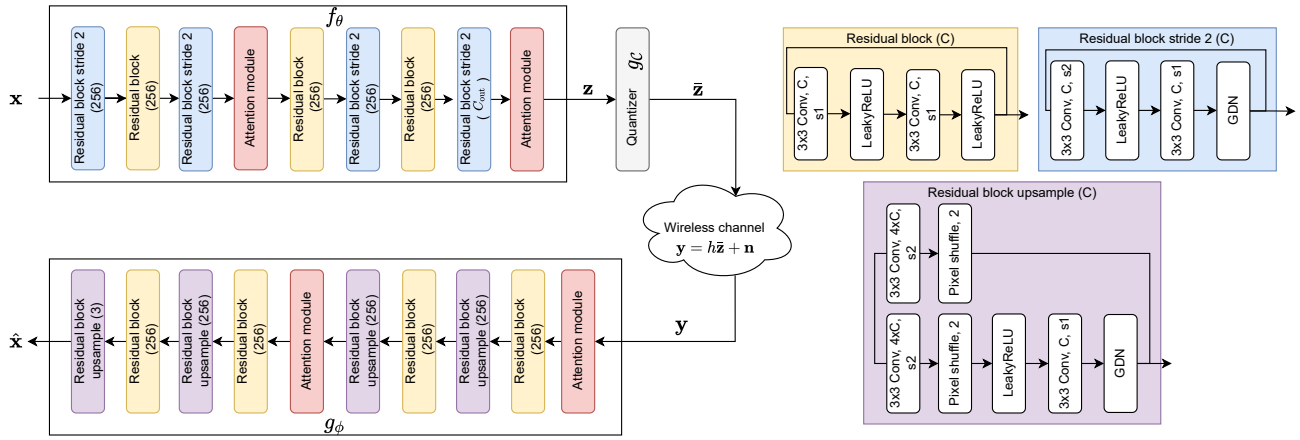


Fig. 3. Architecture of the proposed encoder and decoder models.

As in previous works [8], [9], we utilize an autoencoder architecture to jointly train the encoder and the decoder. We propose a fully convolutional encoder $f_{\boldsymbol{\theta}}$ and decoder $g_{\boldsymbol{\phi}}$ architecture as shown in Fig. 3. In the architecture, C refers to the number of channels in the output tensor of the convolution operation. C_{out} refers to the number of channels in the final output tensor of the encoder $f_{\boldsymbol{\theta}}$, which controls the number of channel uses k per image. The "Pixel shuffle" module, within the "Residual block upsample" module, is used to increases the height and width of the input tensor by reshaping it, such that the channel dimensions are reduced while the height and width dimensions are increased. This was first proposed in [37] as a less computationally expensive method for increasing the CNN tensor dimensions without requiring large number of parameters, like tranpose convolutional layers. The GDN layer refers to *generalised divisive normalization*, initially proposed in [38], and has been shown to be effective in density modeling and compression of images. The Attention layer refers to the simplified attention module proposed in [39], which reduces the computation cost of the attention module originally proposed in [40]. The attention mechanism has been used in both [39] and [40] to improve the compression efficiency by focusing the neural network on regions in the image that require higher bit rate. In our model, this will allow the model to allocate channel bandwidth and power resources optimally. $g_\mathcal{C}$ refers to the two quantization strategies, which we will introduce next.

A. Quantization

In order to produce an encoder that outputs channel symbols from a finite constellation, we perform quantization of the latent vector generated by the encoder, $\bar{\mathbf{z}} = g_\mathcal{C}(\mathbf{z})$. We will consider two quantization strategies: the *soft-to-hard quantizer*, first introduced by [41], and the *learned soft-to-hard quantizer*, which is our extension of the soft-to-hard quantizer that allows the constellation $\mathcal{C}$ to be learned as well. We will first describe the soft-to-hard quantizer.

1) Soft-to-hard quantizer: Given the encoder output $\mathbf{z}$, we first apply a "hard" quantization, which simply maps element $z_i \in \mathbf{z}$ to the nearest symbol in $\mathcal{C}$. This forms the channel input $\bar{\mathbf{z}}$. However, this operation is not differentiable. In order to obtain a differentiable approximation of the hard quantization operation, we will use the "soft" quantization approach, proposed in [41]. In this approach, each quantized symbol is generated as the softmax weighted sum of the symbols in $\mathcal{C}$ based on their distances from z_i; that is,

$$\tilde{z}_i = \sum_{j=1}^{M} \frac{e^{-\sigma_q d_{ij}}}{\sum_{n=1}^{M} e^{-\sigma_q d_{in}}} c_j, \tag{12}$$

where σ_q is a parameter controlling the "hardness" of the assignment, and $d_{ij} = ||z_i - c_j||_2^2$ is the squared l_2 distance between the latent value z_i and the constellation point c_j. As such, in the forward pass, the quantizer uses the hard quantization, corresponding to the channel input $\bar{\mathbf{z}}$, and in the backward pass, the gradient from the soft quantization $\tilde{\mathbf{z}}$ is used to update $\boldsymbol{\theta}$. That is,

$$\frac{\partial \bar{\mathbf{z}}}{\partial \mathbf{z}} = \frac{\partial \tilde{\mathbf{z}}}{\partial \mathbf{z}}. \tag{13}$$

A diagram illustrating the soft-to-hard quantizer for 4-QAM, also known as the quadrature phase shift keying (QPSK), is shown in Fig. 4.

We consider constellation symbols that are uniformly distributed in a square lattice over the complex plane, similar to QAM-modulation. For QAM constellation consisting of M symbols, denoted as M-QAM, we define the max amplitude A_{max} and inter symbol distance d_{sym} as:

$$A_{max} = \frac{(M-1)}{2}\sqrt{\frac{12\bar{P}}{(M^2-1)}}, \tag{14}$$

$$d_{sym} = \sqrt{\frac{12\bar{P}}{(M^2-1)}}, \tag{15}$$

where $\bar{P}$ is the average power of the constellation under

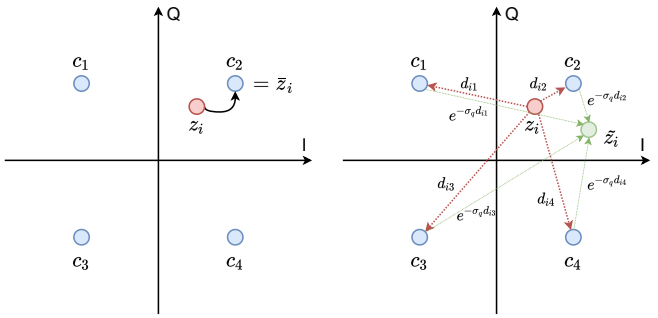


Fig. 4. Illustration of the soft-to-hard quantization procedure for a single value z_i using a QPSK constellation. The hard quatized value $\bar{z}_i$, on the left hand side, simply maps the latent value z_i to the nearest point in the constellation, while the soft quantized value $\tilde{z}_i$ is the softmax weighted sum of the constellation points according to the squared l_2 distance of z_i to each constellation point.

uniform distribution, i.e., $\mathbb{E}[\mathcal{C}^2] = \frac{1}{M}\sum_{i=1}^{M}|c_i|^2 = \bar{P}$.

2) Learned soft-to-hard quantizer: For the learned soft-to-hard quantization, the process is the same except the constellation $\mathcal{C}$ is part of the parameters to be trained. That is, given Eqn. (12), the gradient of the soft quantized value $\tilde{z}_i$ with respect to the input is

$$\frac{\partial \tilde{z}_i}{\partial z_i} = \frac{\partial \mathcal{C}}{\partial z_i}\frac{\partial \tilde{z}_i}{\partial \mathcal{C}}. \tag{16}$$

The constellation points are initialized in the same way, but are allowed to be updated during training. In order to maintain the power constraint, we normalize the symbol power after each update

$$c_i \leftarrow \frac{\sqrt{\bar{P}}}{\sqrt{\sum_{j=1}^{M} P(c_j)|c_j|^2}} c_i. \tag{17}$$

Since we do not have the distribution over the constellation points $P(c_j),\ j = 1, ..., M$, we estimate this probability distribution with a batch of input images $\{\mathbf{x}^v\}_{v=1}^{B}$, where B is the batch size, utilizing the convex weights used in the soft assignment in Eq. (12). Since the weights sum to 1, we can treat them as probabilities and average the probability of each constellation point over the training batch to obtain an estimate of $P(c_j)$. That is, the probability of selecting a constellation point c_j can be estimated as

$$\hat{P}(c_j) = \frac{1}{Bk}\sum_{v=1}^{B}\sum_{i=1}^{k}\frac{e^{-\sigma_q d_{ij}^v}}{\sum_{n=1}^{M} e^{-\sigma_q d_{in}^v}}, \tag{18}$$

where $\hat{P}(c_j)$ is the empirical probability the constellation point c_j, estimated from a batch of input images $\{\mathbf{x}^v\}_{v=1}^{B}$, $d_{ij}^v = ||z_i^v - c_j||_2^2$ is the squared l_2 distance between the ith element in the vth latent vector in the batch and the constellation point c_j. The constellation symbol power is then normalized as

$$c_i \leftarrow \frac{\sqrt{\bar{P}}}{\sqrt{\sum_{j=1}^{M} \hat{P}(c_j)|c_j|^2}} c_i. \tag{19}$$

We note that, compared to other discretization methods, such as VQ-EMA [42], we do not use a separate embedding loss to learn the geometry of the constellation. In VQ-EMA, the loss function is a combination of the reconstruction distortion loss and the embedding loss. The gradient of the reconstruction distortion loss is calculated by using the straight-through estimation of the quantization layer, while the gradient of the embedding loss is calculated using the l_2 distance of the embedding to the pre-quantized value. This allows the encoder parameters and the embedding vectors to be updated simultaneously, but requires careful tuning of the hyperparameters. In our method, by treating the constellation points as parameters of the encoder, the gradients for updating the constellation points are obtained via the softmax weighted sum, as shown in Eqn. (12). This avoids the need to separately tune the balance between distortion loss and embedding loss and we empirically found that this approach performs better than the VQ-EMA approach in our use case.

B. Training Strategy

In order to promote exploration of the available constellation points, we introduce a regularization term based on the Kullback-Leibler (KL) divergence between the distribution $P(\mathcal{C})$ and a uniform distribution over the constellation set $\mathcal{U}(\mathcal{C})$. The KL divergence between two distributions P_W and P_V is defined as

$$D_{\mathrm{KL}}(P_W \mid\mid P_V) = \mathbb{E}\left[\log\left(\frac{P_W}{P_V}\right)\right], \tag{20}$$

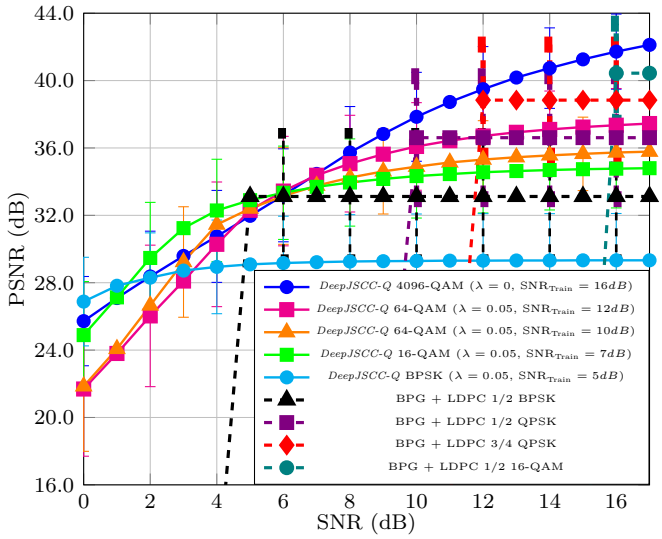
and it measures how different the two distributions are, with $D_{\mathrm{KL}}(P_W \mid\mid P_V) = 0 \iff P_W = P_V$. By regularizing the distortion loss with the KL divergence $D_{\mathrm{KL}}(P(\mathcal{C}) \mid\mid \mathcal{U}(\mathcal{C}))$, we encourage the quantizer $q_{\mathcal{C}}$ to explore the available constellation points, which may improve the end-to-end performance of the system. The distribution $P(\mathcal{C})$ is estimated as in Eqn. (18). Therefore, the final loss function we use for training is:

$$l(\mathbf{x}, \hat{\mathbf{x}}) = d(\mathbf{x}, \hat{\mathbf{x}}) + \lambda D_{\mathrm{KL}}(\hat{P}(\mathcal{C}) \mid\mid \mathcal{U}(\mathcal{C})), \tag{21}$$

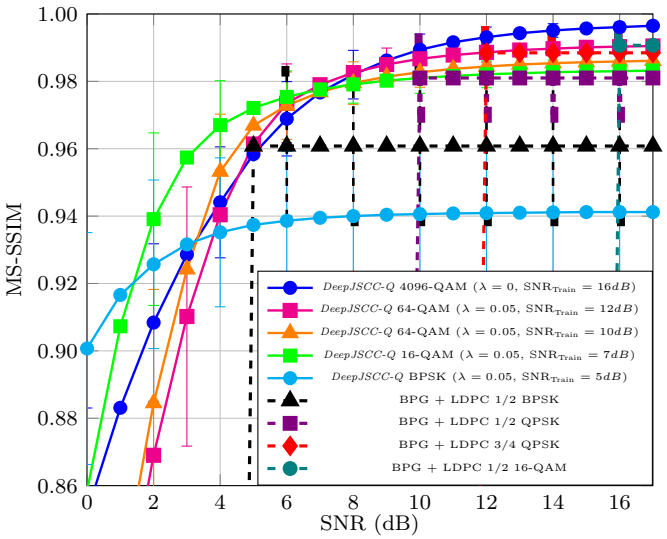
where $d(\cdot,\cdot)$ is the distortion measure (MSE if evaluating on the PSNR metric or 1-MS-SSIM if evaluating on the MS-SSIM metric) and λ is the weighting parameter to control the amount of regularization.

V. Experimental Results

Herein, we perform a series of experiments to demonstrate the performance of *DeepJSCC-Q*. For the first CSI acquisition scenario, defined in Sec. III, we will consider a constant channel gain magnitude $|h| = 1$, which implies a static AWGN channel (referred to as "AWGN" channel henceforth), while in the second scenario, we consider $h \sim CN(0,1)$ and will refer to it as the "slow fading" channel. We train the model for $\mathrm{SNR}_{\mathrm{Train}} \in \{7, 10, 16\}$dB on the ImageNet dataset [43] which consists of 1.2 million RGB images of various resolution. We split the dataset into 9 : 1 for training and validation, respectively. In order to train in batches, we take random crops of 128×128

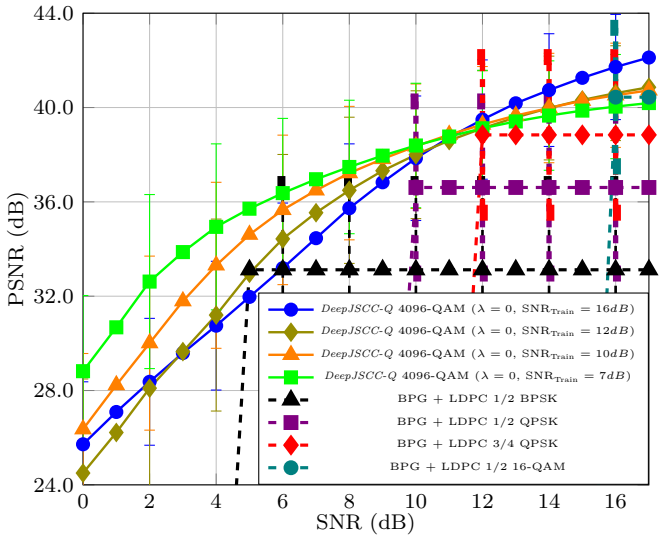


(a) PSNR

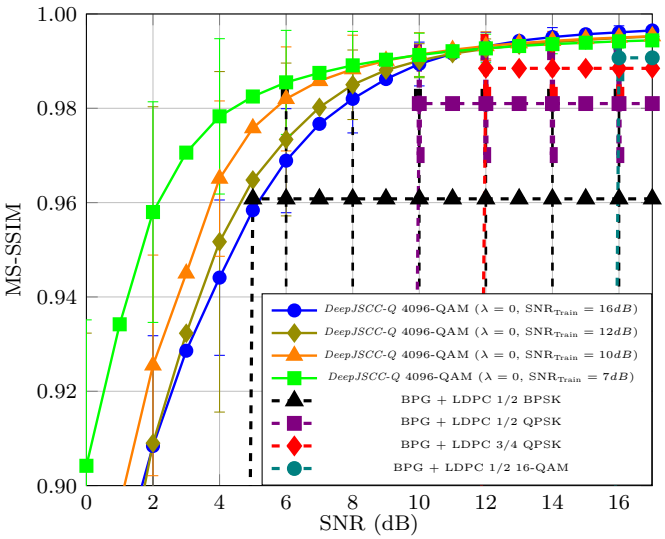


(b) MS-SSIM

Fig. 5. Comparison of *DeepJSCC-Q* trained for different $\text{SNR}_{\text{Train}}$ with the separation approach using BPG for source coding and LDPC codes for channel coding in the AWGN channel case.



(a) PSNR



(b) MS-SSIM

Fig. 6. Comparison of *DeepJSCC-Q* using modulation order $M = 4096$ trained for different $\text{SNR}_{\text{Train}}$ with the separation approach using BPG for source coding and LDPC codes for channel coding in the AWGN channel case.

from the training images. For final evaluation, we use the Kodak dataset[1] consisting of 24 768×512 images. We use the Pytorch [44] library and the Adam [45] optimizer with $\beta_1 = 0.9$, and $\beta_2 = 0.99$ to train our encoder and decoder networks. The learning rate was initialized at 0.0001 for the AWGN channel case, while for the slow fading case, we initialized at 0.00005. We use a batch size of 32 and early stopping with a patience of 8 epochs, where the maximum number of training epochs is 1000. We implement learning rate scheduling, where the learning rate is reduced by a factor of 0.8 if the loss does not improve for 4 epochs consecutively. We use an average constellation power $\bar{P} = 1$ and change the channel noise power σ^2 accordingly to obtain any given SNR. The soft-to-hard hardness parameter σ_q is linearly annealed using the annealing function

$$\sigma_q^{(t)} = \min\left(100, \sigma_q^{(t-1)} + 5\left\lfloor \frac{t}{10000} \right\rfloor\right), \tag{22}$$

where t is the parameter update step number and we initialize $\sigma_q^{(0)} = 5$. We set the weighting for the regularizer $\lambda = 0.05$ for the AWGN channel case when the size of the constellation is relatively small, i.e., $M < 4096$, as we empirically found it to be helpful to encourage the channel input to be more uniformly distributed across the constellation set, while for $M \geq 4096$, $\lambda = 0$ performed better, indicating that it is more beneficial to choose a subset of available symbols with higher probability than using all symbols with the same frequency. For the slow

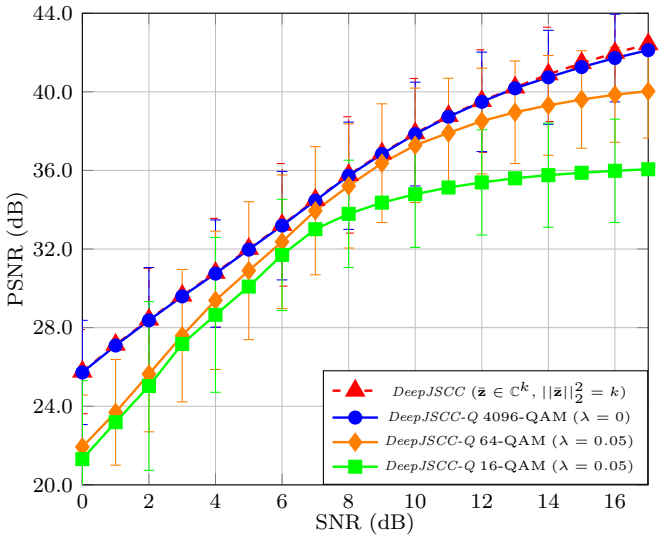
[1]http://r0k.us/graphics/kodak/

fading channel case, we found $\lambda = 0$ to produce better results regardless of the constellation order.

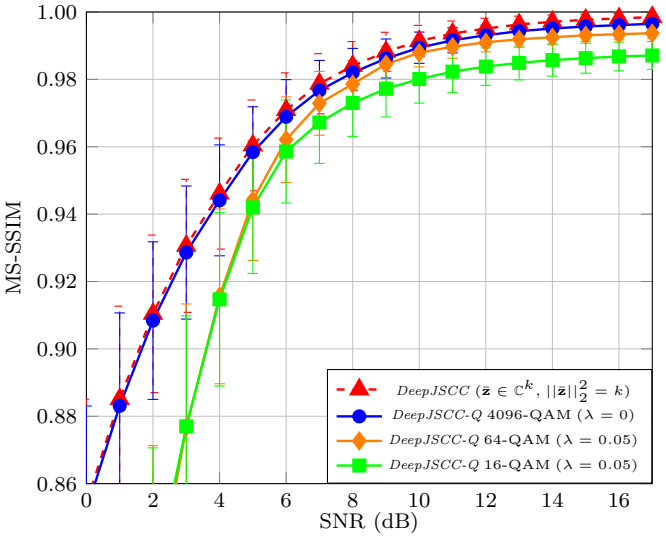
In order to compare the performance of our solution, we consider a baseline separation scheme, in which images are first compressed into bits using the BPG image compression codec [17] and then protected from channel distortion with low density parity check (LDPC) codes. The LDPC codes we use are from the IEEE 802.11ad standard [15], with block length 672 bits for both rate 1/2 and 3/4 codes. We will compare the average image quality over the evaluation dataset, with error bars showing the standard deviation of the image quality metric across the dataset. We refer to M-ary constellations using soft-to-hard quantization as "M-QAM", indicating the constellation is a standard square QAM, whereas for the learned constellations, we refer to them as "L-M".

Fig. 5 shows the result of *DeepJSCC-Q* trained for different $\text{SNR}_{\text{Train}}$ and modulation order M and tested over a range of channel SNR values for the AWGN channel case. For all M-QAM constellations shown here, *DeepJSCC-Q* exhibited graceful degradation of image quality with decreasing channel quality. This is similar to the DeepJSCC results in [8], but we are able to obtain the same behavior despite being constrained to a finite constellation. Moreover, when compared with the separation-based results, the *DeepJSCC-Q* 4096-QAM model performed almost exactly on the envelope of all the separation-based schemes. Although the *DeepJSCC-Q* models trained with modulation orders $M = 16, 64$ did not perform as well as the separation-based schemes, increasing the modulation order at those SNRs can improve the performance of *DeepJSCC-Q*, as shown in Fig. 6. We see that when we employ a modulation order of $M = 4096$ for $\text{SNR}_{\text{Train}} = 7, 10$dB, *DeepJSCC-Q* beats the separation-based schemes convincingly. This shows that, the end-to-end optimization of *DeepJSCC-Q* is fundamentally different from separation-based schemes, as the end-to-end distortion is directly affected by the channel distortion rather than the successful decoding of the channel code for a given source code rate.

We note that the best performance with the separation approach at each channel SNR requires choosing the right constellation size. Increasing the constellation order effectively increases the transmission rate. However, while the increase in the transmission rate increases the quality of the compressed image, this will also increase the error probability over the channel. Hence, we should choose the right constellation size for each channel SNR. On the other hand, for *DeepJSCC-Q*, given an SNR, the higher the constellation order, the better the end-to-end performance. This can be understood from the perspective of quantization error. A higher order modulation essentially corresponds to greater number of quantization levels of the encoder output $\mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{x})$, and thus a greater accuracy of $\bar{\mathbf{z}}$ in representing $\mathbf{z}$. Since DeepJSCC has already been shown to surpass the performance of BPG and LDPC codes by [9], naturally, as we increase the constellation order, the performance of *DeepJSCC-Q* will approach that



(a) PSNR



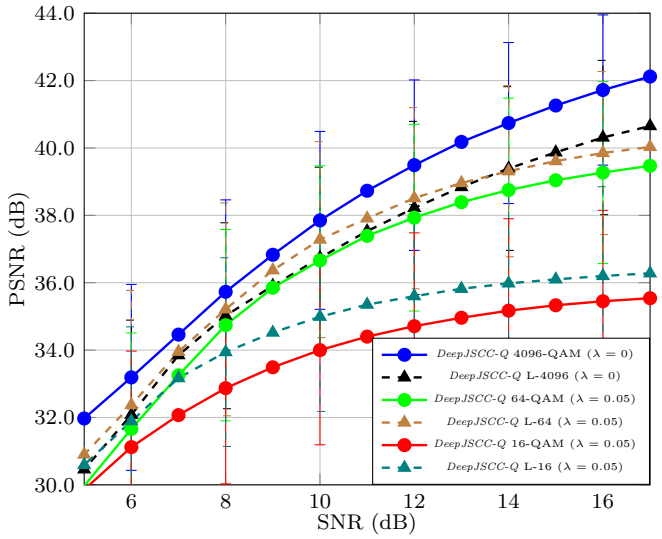
(b) MS-SSIM

Fig. 7. Comparison of *DeepJSCC-Q* with constellation orders $M \in \{16, 64, 4096\}$ against non-quantized *DeepJSCC* for the AWGN channel case ($\text{SNR}_{\text{Train}} = 16$ dB).

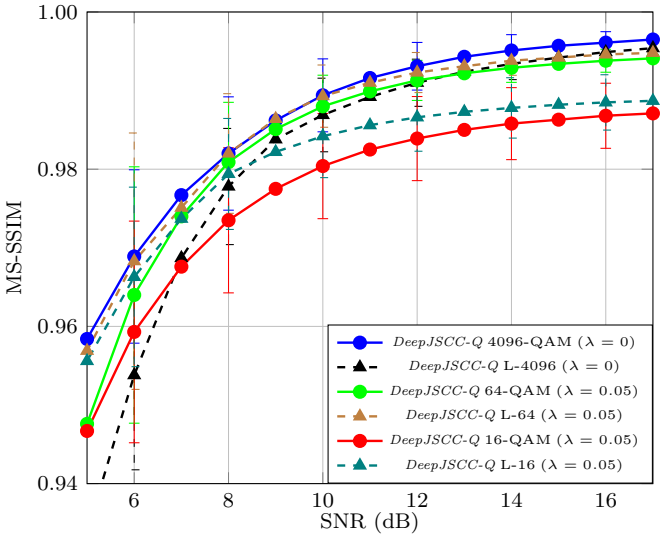
of DeepJSCC. This is also why *DeepJSCC-Q* with 4096-QAM constellation performs better than 64-QAM in Fig. 5 for SNR > 7, even when the 4096-QAM model was trained for a higher SNR than the 64-QAM model.

This observation is further supported by Fig. 7, where we can see that increasing the modulation order monotonically improves the performance of *DeepJSCC-Q*. Moreover, performance of *DeepJSCC-Q* asymptotically approaches unconstrained DeepJSCC as the modulation order increases, with the $|\mathcal{C}| = 4096$ model performing nearly the same. Note that to compare to DeepJSCC, we simply remove the quantizer $q_{\mathcal{C}}$ and normalize the output power of the encoder, such that the channel input is

$$\bar{\mathbf{z}} = \frac{\sqrt{kP}}{||\mathbf{z}||_2}\mathbf{z}. \tag{23}$$

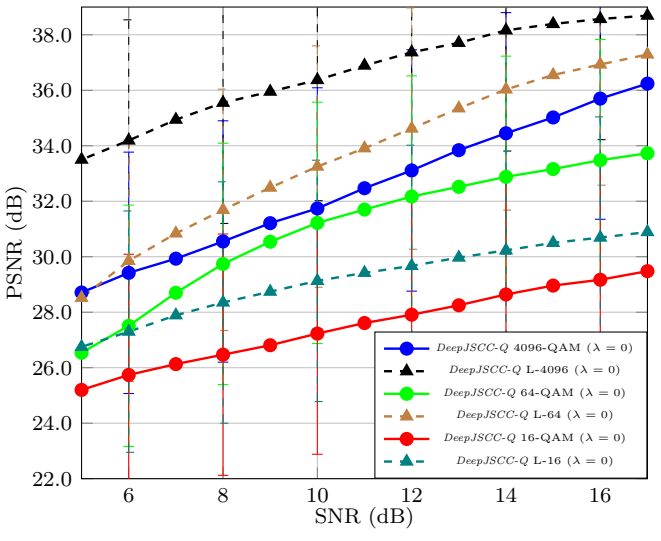


(a) PSNR

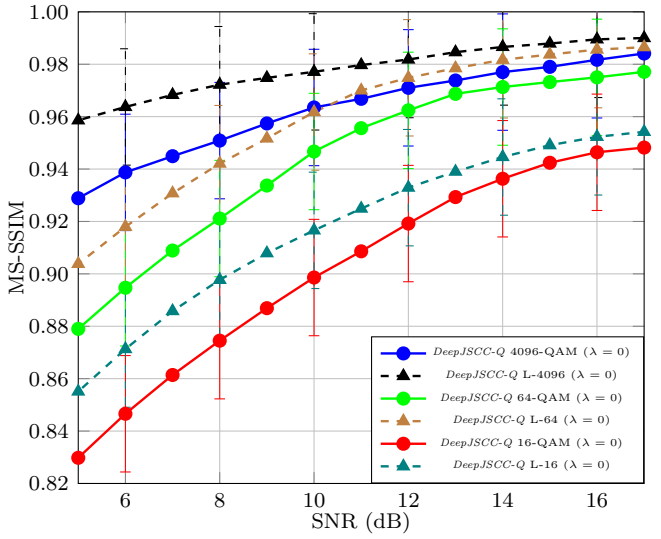


(b) MS-SSIM

Fig. 8. Comparison of *DeepJSCC-Q* trained using fixed and learned constellation for the AWGN channel case ($\text{SNR}_{\text{Train}} = 16$ dB).



(a) PSNR



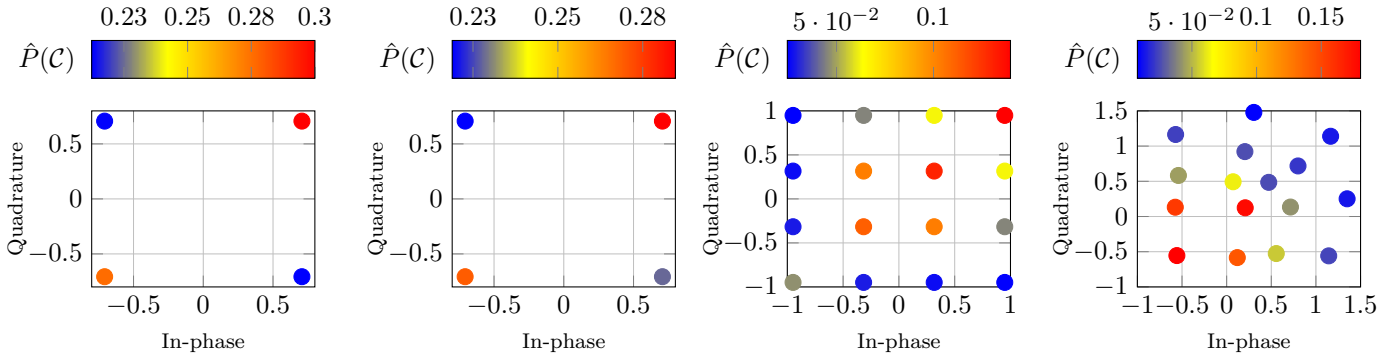
(b) MS-SSIM

Fig. 9. Comparison of *DeepJSCC-Q* trained using fixed and learned constellation for the slow fading channel case ($\text{SNR}_{\text{Train}} = 16$ dB).

It is worth mentioning that for the modulation orders considered in Fig. 7, the LDPC codes considered herein cannot be decoded successfully for the SNR range tested.

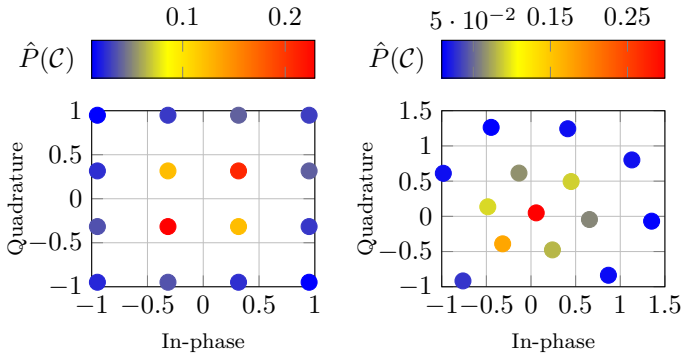
Next, we compare the models trained using soft-to-hard quantization with models trained using learned soft-to-hard quantization in Fig. 8 for the AWGN channel case. We see that across $M = \{16, 64, 4096\}$, only the L-4096 result performed worse than its 4096-QAM counterpart. This indicates that the degree of flexibility provided by the trainable constellation points can produce more coherent mapping between the source and channel input than a fixed M-QAM constellation. The reason that the performance of the L-4096 model falls short of the 4096-QAM model is likely because the order of the modulation for $M = 4096$ is sufficiently high that the 4096-QAM constellation is already near optimal, and the large size of the constellation makes it challenging to further improve its performance. This is supported by Fig. 7, where 4096-QAM performs nearly as well as the non-quantized result. Although the L-4096 result should perform at least as good as the 4096-QAM result, it is possible that the objective function leads to large changes to the constellation that produces suboptimal results. We believe that better tuning of the hyperparameters, such as the quantization hardness parameter σ_q, may alleviate this issue. A similar pattern can be seen for the slow fading channel case shown in Fig. 9, where the margins between the learned and QAM constellations are much bigger, with the learned constellations producing substantially better results than their QAM counterparts. In fact, the L-64 results even outperformed the 4096-QAM results in the PSNR metric. This shows the importance of optimizing channel input geometry and distribution for non-Gaussian channels.

To understand the type of constellations learned by



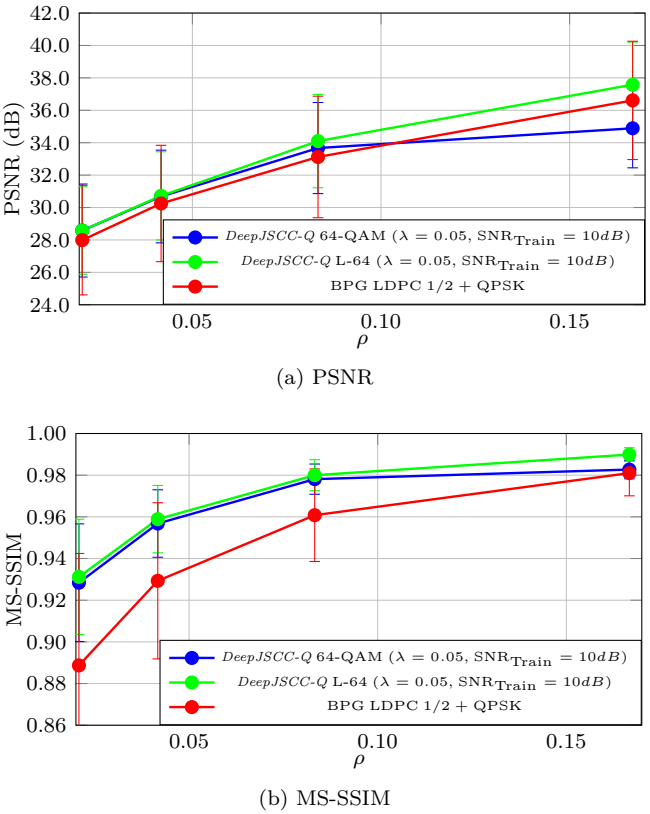
(a) 4-QAM (b) L-4 (c) 16-QAM (d) L-16

Fig. 10. Probability distribution of constellation points for fixed and learned constellations in the AWGN channel case.



(a) 16-QAM (b) L-16

Fig. 11. Probability distribution of constellation points for fixed and learned constellations using modulation order $M = 16$ in the slow fading channel case.



(a) PSNR

(b) MS-SSIM

Fig. 12. Comparison of *DeepJSCC-Q* for different bandwidth compression ratio ρ for the AWGN channel case.

DeepJSCC-Q using learned soft-to-hard quantization, we refer to Fig. 10, where we show the visualization of the constellation points and the probability of selecting each of the points for the AWGN channel case. We can see that for a low modulation order, such as $M = 4$ shown in Fig. 10a and 10b, the constellation points learned by *DeepJSCC-Q* are not very different from 4-QAM, and the distribution across the points is close to uniform. However, when we increase the modulation order to $M = 16$, we begin to see significant differences between L-16 and 16-QAM, as shown by Fig. 10c and 10d. The L-16 constellation is clearly non-square and not centered about the origin $(0, 0)$, with constellation points in the first quadrant having much higher power than those in the third quadrant. This asymmetry may be part of the reason for the performance gain as the channel noise is zero mean and symmetric, meaning using the high power constellation points in the first quadrant can increase the instantaneous SNR of the received signal. The average power is then maintained by choosing the constellations in the third quadrant much more frequently than the first quadrant. Note that this observation can be seen as a form of unequal error protection (UEP) as well, with few important features using high power constellation points (first quadrant) and less important features using lower power constellation points (third quadrant).

For the slow fading case, we also see substantial differences between L-16 and 16-QAM, as shown in Fig. 11. While both the L-16 and the 16-QAM constellations select the central constellation points more frequently in the slow fading channel, the L-16 constellation is much more circular. The L-16 constellation exhibits two circles around a center point. The outer constellation points also have more power than even the highest power constellation points in 16-QAM, with the average power maintained by radially decreasing the probability distribution. Moreover,

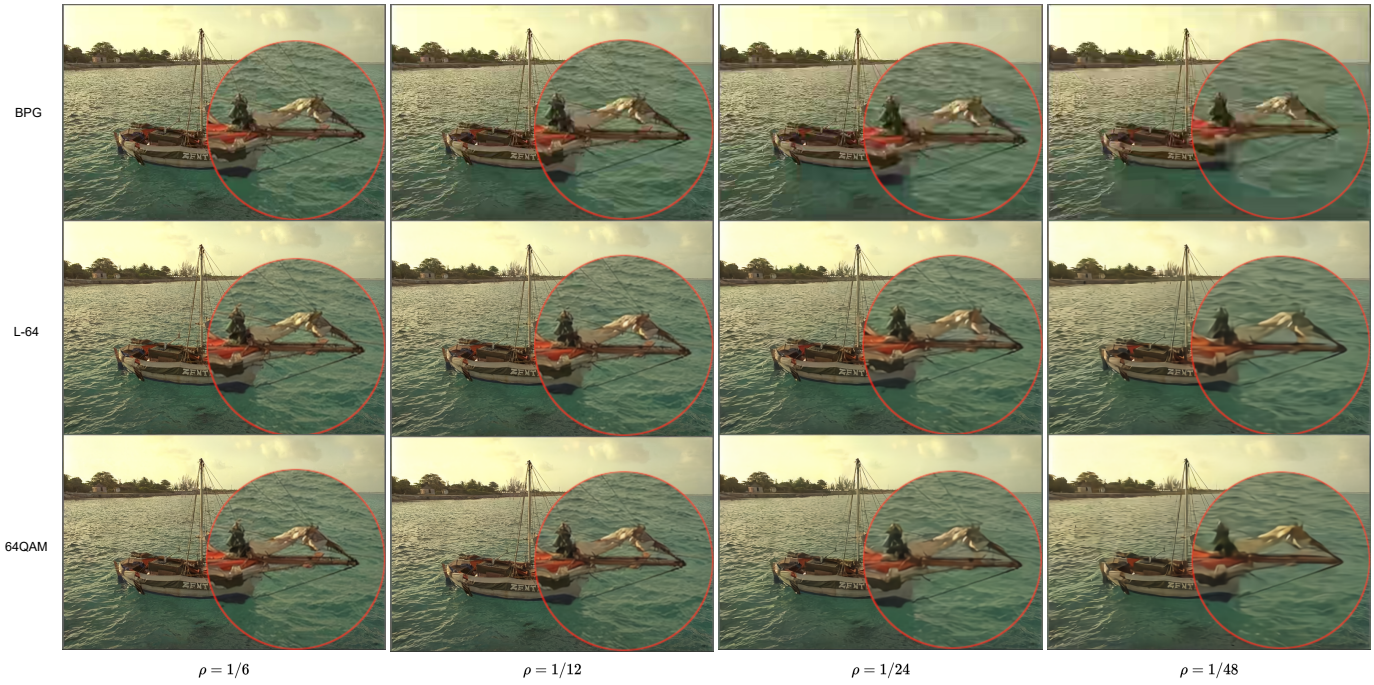


Fig. 13. Visualization of image samples for different bandwidth compression ratios transmitted over the AWGN channel.

when compared with the L-16 constellation learned under the AWGN channel, Fig. 11b shows a much more centered constellation, when compared to Fig. 10d. This is similar to the results from [35], but here we are showing these properties in a JSCC context.

Lastly, we investigate the relationship between the *bandwidth compression ratio* ρ and the reconstruction distortion. Fig. 12 compares the performance of *DeepJSCC-Q* using $M = 64$ with separate source and channel coding for SNR = 10dB for the AWGN channel case. This is akin to plotting the end-to-end rate distortion performance for each of the schemes considered herein. We see that in all but one instance, *DeepJSCC-Q* performs better than separation using BPG for source coding and an LDPC code for channel coding, demonstrating the superiority of JSCC in the finite block length regime. Visual examples of the schemes considered herein given different bandwidth compression ratios ρ are presented in Fig. 13, where the last column of images clearly show that BPG produces more blurry images when compared to *DeepJSCC-Q* using both L-64 and 64QAM constellations, under $\rho = 1/48$. The only instance where *DeepJSCC-Q* performed worse than separation is when the constellation is restricted to 64-QAM and the bandwidth compression ratio is $\rho = 1/6$. This is likely because, as the bandwidth utilization increases, the encoder has greater degrees of freedom, allowing for more expressive channel input features that benefit from non-uniform quantization.

VI. Conclusions

In this paper, we have proposed *DeepJSCC-Q*, an end-to-end optimized joint source-channel coding scheme for wireless image transmission that is able to utilize a fixed channel input constellation and achieve similar performance to unquantized DeepJSCC, as previously proposed in [8]. Even with a constrained constellation, we are able to achieve superior performance to separation-based schemes using BPG for source coding and LDPC for channel coding, all the while avoiding the *cliff-effect* that plagues the separation-based schemes. This makes the viability of *DeepJSCC-Q* in existing commercial hardware with standardized protocols much more attractive. We also show that with sufficiently high modulation order, *DeepJSCC-Q* can approach the performance of DeepJSCC, which does not have a fixed channel input constellation. As such, if such constellations are available on the hardware, *DeepJSCC-Q* can perform nearly as well as DeepJSCC. Finally, we demonstrate that it is possible to learn a finite channel input alphabet that further improves the performance of *DeepJSCC-Q*, with the resultant constellation showing highly non-trivial geometry and distribution. These promising results can bring DNN-based JSCC schemes closer to real world deployment.

References

[1] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, pp. 379–423 and 623–656, July and October 1948.

[2] R. G. Gallager, *Information Theory and Reliable Communication*. USA: John Wiley & Sons, Inc., 1968.

[3] V. Kostina and S. Verdú, "Lossy joint source-channel coding in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 59, pp. 2545–2575, May 2013.

[4] M. Gastpar, B. Rimoldi, and M. Vetterli, "To code, or not to code: Lossy source-channel communication revisited," *IEEE Transactions on Information Theory*, vol. 49, pp. 1147–1158, May 2003.

[5] W. Yu, Z. Sahinoglu, and A. Vetro, "Energy efficient JPEG 2000 image transmission over wireless sensor networks," in *IEEE Global Telecommunications Conference, 2004. GLOBECOM '04.*, vol. 5, pp. 2738–2743 Vol.5, Nov. 2004.

[6] S. Appadwedula, D. Jones, K. Ramchandran, and L. Qian, "Joint source channel matching for a wireless image transmission," in *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No.98CB36269)*, vol. 2, pp. 137–141 vol.2, Oct. 1998.

[7] J. Cai and C. W. Chen, "Robust joint source-channel coding for image transmission over wireless channels," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, pp. 962–966, Sept. 2000.

[8] E. Bourtsoulatze, D. Burth Kurka, and D. Gündüz, "Deep joint source-channel coding for wireless image transmission," *IEEE Transactions on Cognitive Communications and Networking*, vol. 5, pp. 567–579, Sep. 2019.

[9] D. Burth Kurka and D. Gündüz, "Joint source-channel coding of images with (not very) deep learning," in *International Zurich Seminar on Information and Communication (IZS 2020). Proceedings*, pp. 90–94, ETH Zurich, 2020.

[10] J. Dai, S. Wang, K. Tan, Z. Si, X. Qin, K. Niu, and P. Zhang, "Nonlinear transform source-channel coding for semantic communications," *IEEE Journal on Selected Areas in Communications*, vol. 40, pp. 2300–2316, Aug. 2022.

[11] Y. M. Saidutta, A. Abdi, and F. Fekri, "Joint source-channel coding over additive noise analog channels using mixture of variational autoencoders," *IEEE Journal on Selected Areas in Communications*, vol. 39, pp. 2000–2013, July 2021.

[12] D. B. Kurka and D. Gündüz, "Deepjscc-f: Deep joint source-channel coding of images with feedback," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 178–193, 2020.

[13] D. B. Kurka and D. Gündüz, "Bandwidth-agile image transmission with deep joint source-channel coding," *IEEE Transactions on Wireless Communications*, pp. 1–1, 2021.

[14] D. H. Ballard, "Modular learning in neural networks," in *Proceedings of the Sixth National Conference on Artificial Intelligence - Volume 1*, AAAI'87, p. 279–284, AAAI Press, 1987.

[15] "IEEE standard for information technology–Telecommunications and information exchange between systems–Local and metropolitan area networks–Specific requirements-Part 11: Wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment 3: Enhancements for very high throughput in the 60 GHz band," *IEEE Std 802.11ad-2012 (Amendment to IEEE Std 802.11-2012, as amended by IEEE Std 802.11ae-2012 and IEEE Std 802.11aa-2012)*, pp. 1–628, Dec. 2012.

[16] "IEEE Standard for Low-Rate Wireless Networks," *IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15.4-2015)*, pp. 1–800, July 2020.

[17] F. Bellard, *Better Portable Graphics*, 2014 (accessed March 13, 2020). https://bellard.org/bpg/.

[18] R. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, pp. 21–28, Jan. 1962. Conference Name: IRE Transactions on Information Theory.

[19] N. Thomos, N. Boulgouris, and M. Strintzis, "Wireless image transmission using turbo codes and optimal unequal error protection," *IEEE Transactions on Image Processing*, vol. 14, pp. 1890–1901, Nov. 2005.

[20] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Transactions on Communications*, vol. 44, pp. 1261–1271, Oct. 1996.

[21] S. Lin and D. J. Costello, *Error control coding: fundamentals and applications*. Upper Saddle River, NJ: Pearson/Prentice Hall, 2004.

[22] S. S. Arslan, P. C. Cosman, and L. B. Milstein, "Coded hierarchical modulation for wireless progressive image transmission," *IEEE Transactions on Vehicular Technology*, vol. 60, pp. 4299–4313, Nov. 2011.

[23] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: an overview," *IEEE Transactions on Consumer Electronics*, vol. 46, pp. 1103–1127, Nov. 2000.

[24] M. Yang, C. Bian, and H.-S. Kim, "Deep joint source channel coding for wireless image transmission with OFDM," *arXiv:2101.03909 [cs, eess, math]*, May 2021.

[25] M. Ding, J. Li, M. Ma, and X. Fan, "SNR-adaptive deep joint source-channel coding for wireless image transmission," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1555–1559, June 2021. ISSN: 2379-190X.

[26] T.-Y. Tung and D. Gündüz, "DeepWiVe: Deep-learning-aided wireless video transmission," *arXiv:2111.13034 [cs, eess]*, Nov. 2021.

[27] K. Choi, K. Tatwawadi, T. Weissman, and S. Ermon, "NECST: Neural joint source-channel coding," Sept. 2018.

[28] L. Deng, "The MNIST database of handwritten digit images for machine learning research [Best of the Web]," *IEEE Signal Processing Magazine*, vol. 29, pp. 141–142, Nov. 2012.

[29] Y. Song, M. Xu, L. Yu, H. Zhou, S. Shao, and Y. Yu, "Infomax neural joint source-channel coding via adversarial bit flip," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 5834–5841, Apr. 2020.

[30] F. Kayhan and G. Montorsi, "Constellation design for transmission over nonlinear satellite channels," in *2012 IEEE Global Communications Conference (GLOBECOM)*, pp. 3401–3406, Dec. 2012.

[31] F. Kayhan and G. Montorsi, "Constellation design for channels affected by phase noise," in *2013 IEEE International Conference on Communications (ICC)*, pp. 3154–3158, June 2013.

[32] G. Foschini, R. Gitlin, and S. Weinstein, "Optimization of two-dimensional signal constellations in the presence of Gaussian noise," *IEEE Transactions on Communications*, vol. 22, pp. 28–38, Jan. 1974.

[33] G. J. Foschini, R. D. Gitlin, and S. B. Weinstein, "On the selection of a two-dimensional signal constellation in the presence of phase jitter and Gaussian noise," *Bell System Technical Journal*, vol. 52, no. 6, pp. 927–965, 1973.

[34] A. J. Kearsley, "Global and local optimization algorithms for optimal signal set design," *Journal of Research of the National Institute of Standards and Technology*, vol. 106, no. 2, pp. 441–454, 2001.

[35] F. A. Aoudia and J. Hoydis, "Joint learning of probabilistic and geometric shaping for coded modulation systems," in *2020 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Dec. 2020.

[36] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, vol. 2, pp. 1398–1402 Vol.2, Nov. 2003.

[37] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1874–1883, June 2016.

[38] J. Ballé, V. Laparra, and E. P. Simoncelli, "Density modeling of images using a generalized normalization transformation," *arXiv preprint arXiv:1511.06281*, 2015.

[39] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *2020 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7936–7945, June 2020.

[40] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-Local Neural Networks," in *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7794–7803, June 2018.

[41] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool, "Soft-to-hard vector quantization for end-to-end learning compressible representations," in *2017 Advances in Neural Information Processing Systems (NIPS)*, pp. 1141–1151, Dec. 2017.

[42] A. Razavi, A. van den Oord, and O. Vinyals, "Generating diverse high-fidelity images with VQ-VAE-2," June 2019.

[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, June 2009.

[44] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," Oct. 2017.

[45] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980 [cs]*, Jan. 2017. arXiv: 1412.6980.