DeepJSCC-Q: Channel Input 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
{tze-yang.tung14, d.kurka, mikolaj.jankowski17, d.gunduz}@imperial.ac.uk

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9838671

Abstract—**Recent works have shown that the task of wireless transmission of images can be learned with the use of machine learning techniques. Very promising results in end-to-end image quality, superior to popular digital schemes that utilize source and channel coding separation, have been demonstrated through the training of an autoencoder, with a non-trainable channel layer in the middle. However, these methods assume that any complex value can be transmitted over the channel, which can prevent the application of the algorithm in scenarios where the hardware or protocol can only admit certain sets of channel inputs, such as the use of a digital constellation. Herein, we propose *DeepJSCC-Q*, an end-to-end optimized joint source-channel coding scheme for wireless image transmission, which is able to operate with a fixed channel input alphabet. We show that *DeepJSCC-Q* can achieve similar performance to models that use continuous-valued channel input. Importantly, it preserves the graceful degradation of image quality observed in prior work when channel conditions worsen, making *DeepJSCC-Q* much more attractive for deployment in practical systems.**

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

I. Introduction

Source coding and channel coding are two essential steps in modern data transmission. The former reduces the redundancy within source data, preserving only the essential information needed to achieve a prescribed reconstruction fidelity. For example, in image transmission, commonly used source coding schemes are JPEG2000 and BPG. Channel coding, on the other hand, introduces structured redundancy within the data to allow reliable decoding under the presence of channel noises. It was proven by Shannon that, the separation of source and channel coding is without loss of optimality when the blocklength goes to infinity and if the source is discrete memoryless [1]. Nevertheless, in practical applications we are limited to finite blocklengths, and it has been known that such communication systems can benefit from combining the two coding steps, motivating the design of joint source-channel coding (JSCC) schemes. Despite their potential advantages, JSCC schemes have found limited use in practice due to their lack of modularity and difficulty in designing such codes. Recently, a method called DeepJSCC, which employs deep neural networks (DNNs) to perform a mapping from the input image space directly to channel input symbols in a joint manner was introduced in [2]. It shows appealing properties, such as better or similar performance compared with state-of-the-art digital compression schemes [3], flexibility to adapt to different source or channel models [2]–[4], ability to exploit channel feedback [5], and capability to produce adaptive-bandwidth transmission schemes [6]. 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, where the image becomes un-decodable due to the channel quality falling below what the channel code anticipates.

One of the strengths of DeepJSCC is the fact that it learns a communication scheme from scratch, defining all transformations in a data-driven manner. This simplifies the code design procedure, and allows adaptation to any particular source or channel domain. We note that the DeepJSCC approach not only combines source and channel coding into one single mapping, but it also removes the constellation diagrams used in digital schemes. Instead, in DeepJSCC, the encoder can transmit arbitrary complex-valued channel inputs (within the power constraint). This can hinder the suitability of using DeepJSCC in current commercial hardware and standardized protocols, which are constrained to produce fixed sets of symbols.

In this work, we investigate the effects of constraining the channel input alphabet 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, limiting the output space available for the encoder, or even in incorporating such techniques into established standards such as cellular communications. Therefore, in this paper, we introduce a new strategy for JSCC of images, *DeepJSCC-Q*, which allows for the transmission of the content through fixed pre-defined constellations. A series of experiments demonstrate that *DeepJSCC-Q* is able to:

- Achieve similar performance results to unconstrained DeepJSCC, even when using a highly constrained channel input representation.
- Achieve superior performance when compared to

This work was supported by the European Research Council (ERC) through project BEACON (No. 677854).

capacity-achieving low density parity check (LDPC) codes [7] with BPG compression [8] under the same channel input constraint.
- Create coherent mapping on neighboring constellation points, thus avoiding the *cliff-effect* present in separation-based schemes.

II. Related Works

JSCC for wireless image transmission using DNNs was first demonstrated in [2], and dubbed *DeepJSCC*. The authors were able to show that by setting up the encoder and decoder in an autoencoder configuration, with a non-trainable channel layer in between, the DNN encoder was able to learn a function which maps images to continuous channel inputs, and vice versa at the decoder. They demonstrated that the resultant JSCC encoder and decoder was able to surpass the performance of JPEG2000 [9] compression followed by LDPC codes [7] for channel coding. Importantly, they 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, 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 [5] and adapt to various bandwidth requirements without retraining [6]. In [4], the viability of DeepJSCC in an orthogonal frequency division multiplexing (OFDM) system is shown and in [10] it is shown that it can be extended to multi-user scenarios with the same decoder.

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 feasible as many commercially available hardware may have hard-coded standard protocols, making these methods less viable for real world deployment. In [11], image transmission problem is considered over a discrete input channel, where the input representation is learned using a variational autoencoder (VAE) assuming a Bernoulli prior. They consider the transmission of MNIST images [12] 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. In [13], the channel coding problem with a fixed channel input alphabet is considered, and it is shown 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). In contrast to these works, we investigate the transmission of natural images over an additive white Gaussian noise (AWGN) channel with a finite channel input alphabet.

III. Problem Statement

Herein, we consider the problem of wireless transmission of images over an additive white Gaussian noise (AWGN) channel, in which communication should be performed using a finite set of symbols. An input image $\mathbf{x} \in \mathbb{R}^{H\times W\times C}$ (where H, W and C represent the image's height, width and color channels, respectively) is mapped with a non-linear encoder function $f_{\boldsymbol{\theta}} : \mathbb{R}^{H\times W\times C} \mapsto \mathbb{C}^k$ into a latent vector $\mathbf{z} = f_{\boldsymbol{\theta}}(\mathbf{x}, \hat{\sigma}^2)$, where $\hat{\sigma}^2$ is the estimated channel noise power. The latent vector $\mathbf{z}$ is then mapped to a channel input alphabet $\mathcal{C}$ via the modulator $q_{\mathcal{C}} : \mathbb{C}^k \mapsto \mathbb{C}^k$, which we will define in the sequel. There are $M = |\mathcal{C}|$ possible symbols, and we refer to $\mathcal{C}$ as the constellation, denoted by $\mathcal{C} = \{c_1, ..., c_M\}$. That is, $\bar{\mathbf{z}} = q_{\mathcal{C}}(\mathbf{z})$, $\bar{z}_i \in \mathcal{C}$, where $\bar{z}_i$ is the ith element of the channel input vector $\bar{\mathbf{z}}$. We define the average power of the constellation assuming equally likely inputs as

$$P = \frac{1}{M}\sum_{j=1}^{M} |c_j|^2. \tag{1}$$

The channel input $\bar{\mathbf{z}}$ is transmitted through an AWGN channel, producing the channel output $\mathbf{y} = \bar{\mathbf{z}} + \boldsymbol{\eta}$, where $\boldsymbol{\eta} \sim CN(0, \sigma^2 I_{k\times k})$ is a complex Gaussian vector with covariance matrix $\sigma^2 I_{k\times k}$, and $I_{k\times k}$ is the $k \times k$ identity matrix. Finally, a receiver passes the channel output through a non-linear decoder function $g_{\boldsymbol{\phi}} : \mathbb{C}^k \mapsto \mathbb{R}^{H\times W\times C}$ to produce a reconstruction of the input $\hat{\mathbf{x}} = g_{\boldsymbol{\phi}}(\mathbf{y}, \hat{\sigma}^2)$.

We assume that both the transmitter and the receiver can estimate the channel noise power, denoted by $\hat{\sigma}^2$, in order to select the best resource allocation strategy. We define the channel SNR as

$$\text{SNR} = 10\log_{10}\left(\frac{P}{\sigma^2}\right) \text{ dB}, \tag{2}$$

and the SNR estimated by the encoder and decoder as

$$\text{SNR}_{\text{Est}} = 10\log_{10}\left(\frac{P}{\hat{\sigma}^2}\right) \text{ dB}. \tag{3}$$

We also define the *bandwidth compression ratio* as

$$\rho = \frac{k}{H\times W\times C} \text{ channel symbols/pixel}, \tag{4}$$

where a smaller number reflects more compression.

To this end, we propose *DeepJSCC-Q*, a DNN model and an end-to-end training strategy, which learns the encoder and decoder parameters $\boldsymbol{\theta}, \boldsymbol{\phi}$, and effectively utilize the available symbols in order to recover the transmitted images with satisfactory quality. We will consider the use of quantization to represent our modulator $q_{\mathcal{C}}$, such that each quantization level represents a point in the constellation. As in previous works [2], [3], we utilize an autoencoder architecture to jointly train the encoder and the decoder. However, one of the drawbacks of the previous works was the need to train multiple networks, one for each channel condition. To address this issue, [14] proposed an attention feature (AF) module, motivated by resource assignment strategies in traditional JSCC schemes [15], which allows the network to learn to assign different weights to different features for a given SNR. This is done by deliberately randomizing the channel SNR

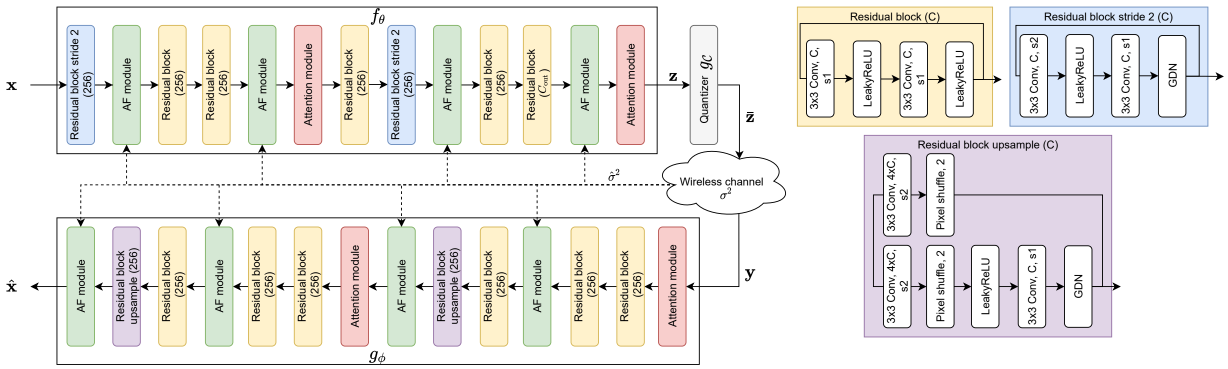


Fig. 1. Architecture of the proposed encoder and decoder models.

during training and providing the AF modules the current SNR, such that $\hat{\sigma}^2 = \sigma^2$. By doing so, the results in [14] show that the single model performs at least as well as the models trained for each SNR individually. We adopt the AF module proposed by [14] in *DeepJSCC-Q* to obtain a single model that can work over a range of SNRs given an estimate of the channel condition.

We propose a fully convolutional encoder and decoder architecture as shown in Fig. 1. 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 dimensions by reshaping the input tensor, such that the channel dimension is reduced while the height and width dimensions are increased. This was first proposed in [16] as a less computationally expensive method for increasing the CNN tensor dimensions without requiring a large number of parameters, like transpose convolutional layers. The GDN layer refers to generalized divisive normalization, initially proposed in [17], and shown to be effective in density modeling and compression of images. The Attention layer refers to the simplified attention module proposed in [18], which reduces the computation cost of the attention module originally proposed in [19]. While the attention layer has been used in [18] and [19] to improve the compression efficiency by learning to focus on image regions that require higher bit rate, in our model it is used to allow adaptive allocation of channel bandwidth and power resources.

A. Quantization

In order to produce an encoder that outputs a fixed constellation, we perform quantization of the latent vector generated by the encoder, $\bar{\mathbf{z}} = g_{\mathcal{C}}(\mathbf{z})$. Given the encoder output $\mathbf{z}$, we first obtain 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 [20]. 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,

$$\tilde{z}_i = \sum_{j=1}^{M} \frac{e^{-\sigma_q d_{ij}}}{\sum_{n=1}^{M} e^{-\sigma_q d_{in}}} c_j, \tag{5}$$

where σ_q is a parameter controlling the “hardness” of the assignment, and $d_{ij} = ||z_i - c_j||_2^2$, is 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,

$$\frac{\partial \bar{\mathbf{z}}}{\partial \mathbf{z}} = \frac{\partial \tilde{\mathbf{z}}}{\partial \mathbf{z}}. \tag{6}$$

A diagram illustrating the soft-to-hard quantizer for 4-QAM, also known as the quadrature phase shift keying (QPSK), is shown in Fig. 2.

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 $A_{max} = \frac{(M-1)}{2}\sqrt{\frac{12P}{(M^2-1)}}$, and inter symbol distance $d_{sym} = \sqrt{\frac{12P}{(M^2-1)}}$, where P is the average power of the constellation under uniform distribution, i.e., $\mathbb{E}[\mathcal{C}^2] = \frac{1}{M}\sum_{i=1}^{M} |c_i|^2 = P$.

B. Training Strategy

Given that our model is end-to-end optimized, we are free to choose any distortion metric. Two common distortion metrics for images are the mean squared error (MSE),

$$\mathrm{MSE}(\mathbf{x}, \hat{\mathbf{x}}) = ||\mathbf{x} - \hat{\mathbf{x}}||_2^2, \tag{7}$$

and the structural similarity index measure (SSIM),

$$\mathrm{SSIM}(\mathbf{x}, \hat{\mathbf{x}}) = \left(\frac{2\mu_{\mathbf{x}}\mu_{\hat{\mathbf{x}}} + v_1}{\mu_{\mathbf{x}}^2 + \mu_{\hat{\mathbf{x}}}^2 + v_1} \right) \left(\frac{2\sigma_{\mathbf{x}}\sigma_{\hat{\mathbf{x}}} + v_2}{\sigma_{\mathbf{x}}^2 + \sigma_{\hat{\mathbf{x}}}^2 + v_2} \right), \tag{8}$$

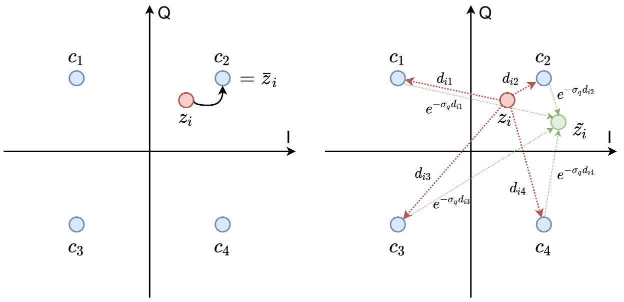


Fig. 2. Illustration of the soft-to-hard quantization procedure for a single value z_i using a QPSK constellation. The hard quatized value $\bar{z}_i$ (left) simply maps the latent value z_i to the nearest point in the constellation, while the soft quantized value $\tilde{z}_i$ (right) is the softmax weighted sum of the constellation points according to l_2 distance.

where $\mu_{\mathbf{x}}$, $\sigma^2_{\mathbf{x}}$, $\sigma^2_{\mathbf{x}\hat{\mathbf{x}}}$ are the mean and variance of $\mathbf{x}$, and the covariance between $\mathbf{x}$ and $\hat{\mathbf{x}}$, respectively, and v_1, v_2 are coefficients for numeric stability. When used on an RGB image, SSIM is computed for all three color channels and then averaged. Note that a higher SSIM value indicates a lower distortion, with a maximum of 1, therefore when used as a training loss, $1 - \text{SSIM}$ is used.

A common complimentary metric to the MSE distortion to measure image reconstruction quality is the peak signal-to-noise ratio (PSNR) defined as

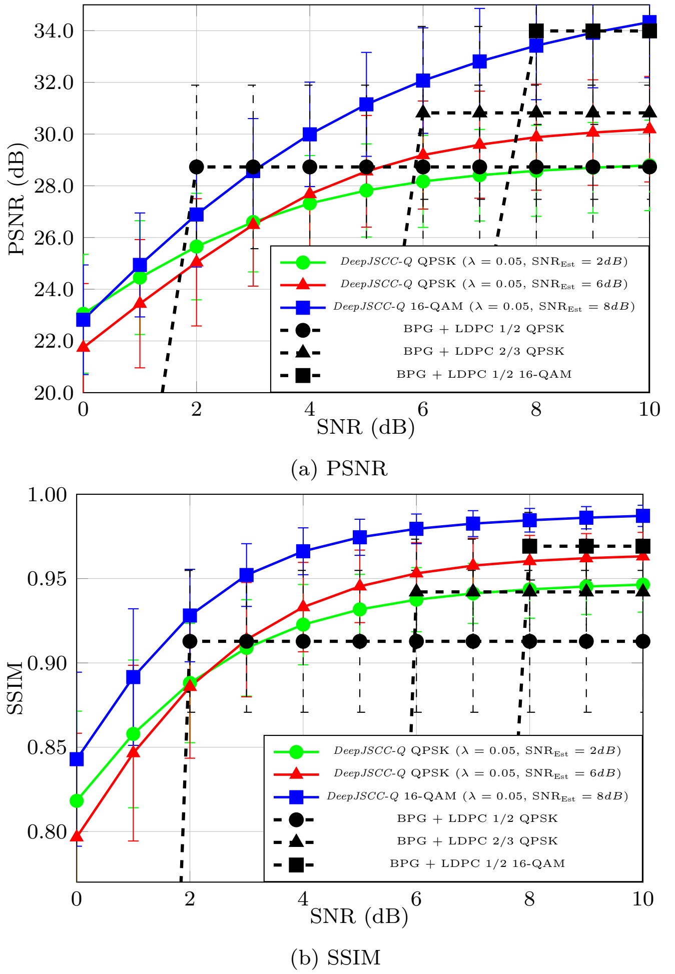
$$\text{PSNR}(\mathbf{x}, \hat{\mathbf{x}}) = \log_{10}\left(\frac{A^2}{\text{MSE}(\mathbf{x}, \hat{\mathbf{x}})}\right) \text{ dB}, \tag{9}$$

where A is the maximum possible value for a given pixel. For a 24 bit RGB pixel, $A = 255$.

In order to utilize the available constellation points in the most optimal way and encourage the transmitted signal to have an average power close to 1, we introduce a regularization term based on the Kullback-Leibler (KL) divergence between the distribution $P(\mathcal{C}|\mathbf{z})$ and a uniform distribution over the constellation set $\mathcal{U}(\mathcal{C})$. The KL divergence between distributions P_W and P_V is defined as

$$D_{\text{KL}}(P_W \mid\mid P_V) = \mathbb{E}\left[\log\left(\frac{P_W}{P_V}\right)\right], \tag{10}$$

and it measures how different the two distributions are, with $D_{\text{KL}}(P_W \mid\mid P_V) = 0 \iff P_W = P_V$. The distribution $P(\mathcal{C}|\mathbf{z})$ represents the probability of selecting a point in the constellation set $\mathcal{C}$ given the encoded latent vector $\mathbf{z}$. By regularizing the distortion loss with the KL divergence $D_{\text{KL}}(P(\mathcal{C}|\mathbf{z}) \mid\mid \mathcal{U}(\mathcal{C}))$, we encourage the quantizer $q_{\mathcal{C}}$ to explore the available constellation points and thus force the transmitted signal power to be as close to 1 as possible. To model the distribution $P(\mathcal{C}|\mathbf{z})$, we utilize the softmax weights used in the soft assignment in Eq. (5). 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 $P(\mathcal{C}|\mathbf{z})$. That is, the probability of selecting a constellation point c_j given a batch of latent vectors $\{\mathbf{z}^v\}_{v=1}^{|\mathcal{B}|}$, where $\mathcal{B}$ is a batch of encoded latent vectors, can be estimated as



(a) PSNR

(b) SSIM

Fig. 3. Effect of channel estimation error. Here, we compare *DeepJSCC-Q* using different SNR_{Est} and distortion metrics to BPG using LDPC codes.

$$\hat{P}(c_j|\mathbf{z}) = \frac{1}{|\mathcal{B}|k}\sum_{v=1}^{|\mathcal{B}|}\sum_{i=1}^{k}\frac{e^{-\sigma_q d_{ij}^v}}{\sum_{n=1}^{M} e^{-\sigma_q d_{in}^v}}, \tag{11}$$

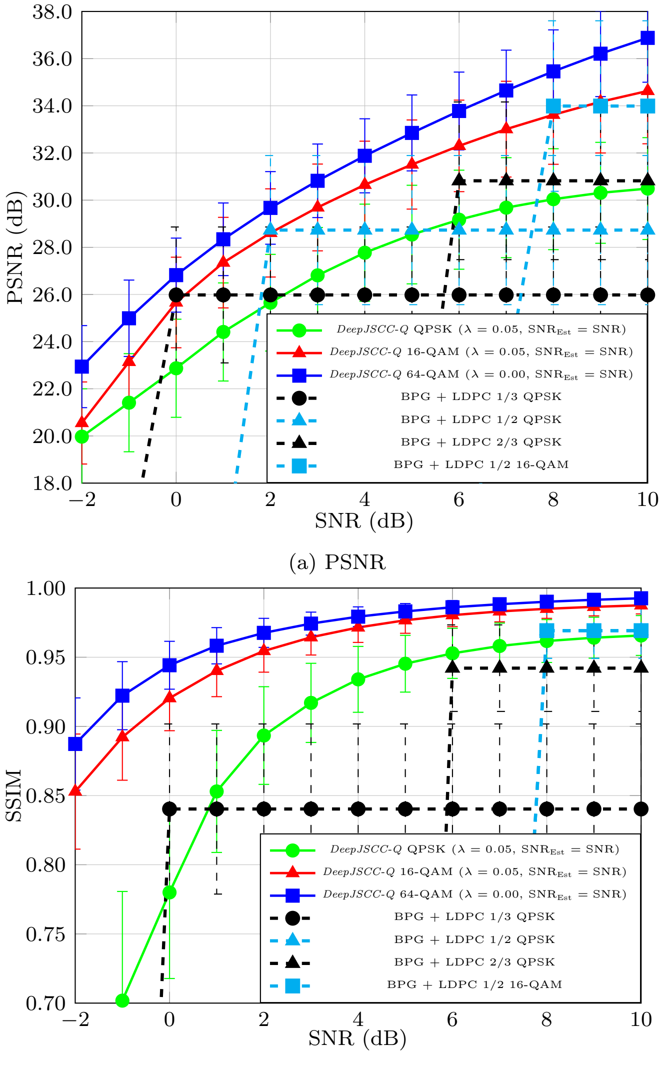
where $d_{ij}^v = ||z_i^v - c_j||_2^2$ is the l_2 distance between the constellation point c_j and the ith element in the vth latent vector in the batch. Therefore, the final loss function we use for training is:

$$l(\mathbf{x}, \hat{\mathbf{x}}) = d(\mathbf{x}, \hat{\mathbf{x}}) + \lambda D_{\text{KL}}(\hat{P}(\mathcal{C}|\mathbf{z}) \mid\mid \mathcal{U}(\mathcal{C})), \tag{12}$$

where $d(\cdot, \cdot)$ is the distortion metric (either MSE or $1 -$ SSIM) and λ is the weighting parameter to control the amount of regularization.

IV. Experimental Results

In this section we perform a series of experiments to demonstrate the performance of *DeepJSCC-Q*. Herein, we consider the CIFAR10 dataset [21] which consists of 60000 RGB images of 32×32 resolution. We split the dataset into $5 : 1$ for training and testing, respectively. We use the Pytorch [22] library and Adam [23] optimizer with learning rate 0.0001, $\beta_1 = 0.9$, and $\beta_2 = 0.999$ to train our encoder



(a) PSNR

(b) SSIM

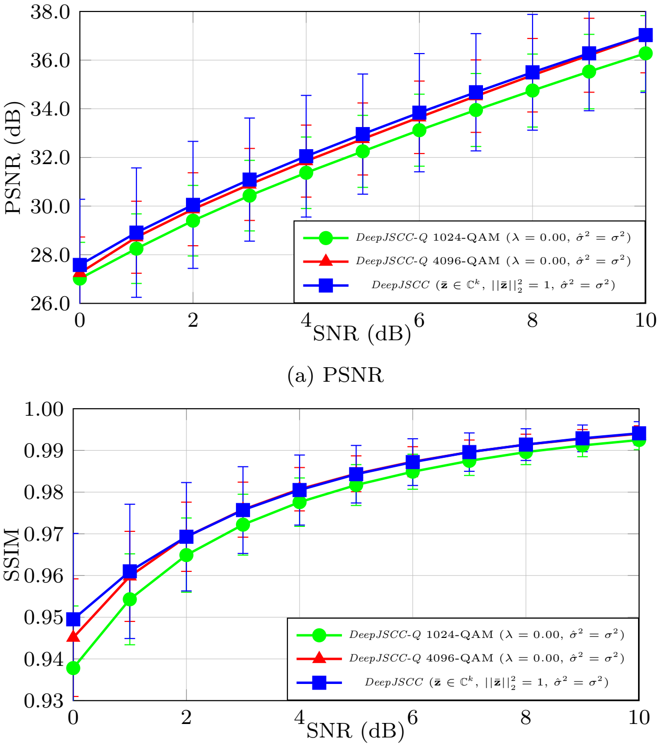
Fig. 4. Comparison of *DeepJSCC-Q* for $\text{SNR}_{\text{Est}} = \text{SNR}$ to BPG using LDPC codes.

and decoder networks. 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 in a row. We use an average constellation power $P = 1$ and change the channel noise power σ^2 accordingly to obtain any given SNR. During training, the channel SNR is drawn uniformly randomly $\text{SNR} \sim \mathcal{U}(0, 10)$. The softmax hardness assignment parameter is chosen as $\sigma_q = 100$. We set the weighting for the regularizer $\lambda = 0.05$ when the size of the constellation M is less than 64, as we experimentally found it to be helpful to encourage the channel input to be more uniformly distributed across the constellation set, while for larger constellations, $\lambda = 0$ 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. We use $C_{\text{out}} = 40$ for all models, which corresponds to $\rho = 0.4166$.

To compare the performance of our solution, we consider separation-based schemes, in which images are first compressed using the BPG [8] codec, before an LDPC code [7] is used as the channel code. We compare the average image quality over the test dataset, with error bars showing the standard deviation of the image quality metric. Fig. 3 shows the effects of channel estimation error on the performance of *DeepJSCC-Q*. The constellation and SNR_{Est} values used for *DeepJSCC-Q* are chosen to coincide with the SNR values at which the separation schemes would fail, in order to highlight the behavior of *DeepJSCC-Q* around the cliff edges of the separation-based schemes. Note that this is done by fixing the channel noise estimate $\hat{\sigma}^2$, while varying the actual channel noise power σ^2. For all M-QAM modulation orders shown here, *DeepJSCC-Q* exhibited graceful degradation of image quality with decreasing channel quality. This is similar to the *DeepJSCC* result from [2], but we are able to obtain the same behavior despite being constrained to a finite digital constellation. Moreover, when compared to the separation-based results, the *DeepJSCC-Q* 16-QAM model performed close to the envelope of all the separation-based schemes, with the model trained using the SSIM distortion performing considerably above the envelope. This shows that the end-to-end optimized *DeepJSCC-Q* is fundamentally different from separation-based schemes, as it is able to maintain a performance on par with separation-based schemes without the need to adapt its code rate.

The reason *DeepJSCC-Q* performs this way is related to the quantization error. A higher order modulation essentially corresponds to greater number of quantization levels of the encoder output $\mathbf{z}$, and thus lower quantization error of $\bar{\mathbf{z}}$ in representing $\mathbf{z}$. Since DeepJSCC has already been shown to surpass the performance of BPG and LDPC codes by [3], naturally as we increase the constellation order M, the performance of *DeepJSCC-Q* will approach that of *DeepJSCC*. This is also why *DeepJSCC-Q* with 16-QAM constellation performs better than both QPSK models in Fig. 3, except at the low SNR regime, although that is due to the SNR_{Est} used for the 16-QAM model being much greater than the channel SNR in that region. In Fig. 4, where the $\text{SNR}_{\text{Est}} = \text{SNR}$, we can see that not only does the 16-QAM *DeepJSCC-Q* model perform better than the QPSK model, the 64-QAM model also performs better than either of those models, for all channel SNRs tested. As further evidence that increasing the modulation order of *DeepJSCC-Q* leads to the performance of *DeepJSCC-Q* approaching that of *DeepJSCC* [2], we investigate very high order modulations $M = 1024, 4096$, as shown in Fig. 5. 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

$$\bar{\mathbf{z}} = \frac{\sqrt{kP}}{||\mathbf{z}||_2}\mathbf{z}. \tag{13}$$



(a) PSNR

(b) SSIM

Fig. 5. Comparison of *DeepJSCC-Q* using high modulation orders to *DeepJSCC* with a continuous channel input, as in [2].

From Fig. 5, the performance of *DeepJSCC-Q* approaches *DeepJSCC* as the modulation order increases, with $M = 4096$ the two models performing almost the same.

V. Conclusions

In this paper, we have proposed *DeepJSCC-Q*, an end-to-end optimized JSCC scheme for image transmission that is able to utilize a fixed modulation constellation and achieve similar performance to unquantized *DeepJSCC*, as previously proposed by [2]. Even with such a constraint, 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. 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*. This makes the viability of *DeepJSCC-Q* in existing commercial hardware with standardized protocols much more attractive.

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] 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.

[3] 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.

[4] 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. arXiv: 2101.03909.

[5] 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.

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] K. Choi, K. Tatwawadi, T. Weissman, and S. Ermon, “NECST: Neural Joint Source-Channel Coding,” *International Conference on Machine Learning (ICML)*, 2019.

[12] 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. Conference Name: IEEE Signal Processing Magazine.

[13] M. Stark, F. A. Aoudia, and J. Hoydis, “Joint learning of geometric and probabilistic constellation shaping,” *IEEE Globecom Workshops*, Dec. 2019.

[14] J. Xu, B. Ai, W. Chen, A. Yang, P. Sun, and M. Rodrigues, “Wireless image transmission using deep source channel coding with attention modules,” *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2021.

[15] K. Sayood, H. H. Otu, and N. Demir, “Joint source/channel coding for variable length codes,” *IEEE Transactions on Communications*, vol. 48, pp. 787–794, May 2000. Conference Name: IEEE Transactions on Communications.

[16] 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,” pp. 1874–1883, June 2016. ISSN: 1063-6919.

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

[18] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, “Learned image compression with discretized Gaussian mixture likelihoods and attention modules,” *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7939–7948, 2020.

[19] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7794–7803, June 2018. ISSN: 2575-7075.

[20] E. Agustsson *et al.*, “Soft-to-hard vector quantization for end-to-end learning compressible representations,” in *Advances in Neural Information Processing Systems 30*, pp. 1141–1151, Curran Associates, Inc., 2017.

[21] A. Krizhevsky, “Learning multiple layers of features from tiny images,” 2009.

[22] A. Paszke *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.

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