

DRASIC: DISTRIBUTED RECURRENT AUTOENCODER FOR SCALABLE IMAGE COMPRESSION

Anonymous authors

Paper under double-blind review

ABSTRACT

We propose a new architecture for distributed image compression from a group of distributed data sources. The work is motivated by practical needs of data-driven codec design, low power consumption, robustness, and data privacy. The proposed architecture, which we refer to as Distributed Recurrent Autoencoder for Scalable Image Compression (DRASIC), is able to train distributed encoders and one joint decoder on correlated data sources. Its compression capability is much better than the method of training codecs separately. Meanwhile, for 10 distributed sources, our distributed system remarkably performs within 2 dB peak signal-to-noise ratio (PSNR) of that of a single codec trained with all data sources. We experiment distributed sources with different correlations and show how our methodology well matches the Slepian-Wolf Theorem in Distributed Source Coding (DSC). Our method is also shown to be robust to the lack of presence of encoded data from a number of distributed sources. Moreover, it is scalable in the sense that codes can be decoded simultaneously at more than one compression quality level. To the best of our knowledge, this is the first data-driven DSC framework for general distributed code design with deep learning.

1 INTRODUCTION

It has been shown by a variety of previous works that deep neural networks (DNN) can achieve comparable results as classical image compression techniques (Toderici et al., 2015; Ballé et al., 2016; Gregor et al., 2016; Toderici et al., 2017; Theis et al., 2017; Johnston et al., 2017; Liu et al., 2018; Li et al., 2018; Mentzer et al., 2018). Most of these methods are based on autoencoder networks and quantization of bottleneck representations. These models usually rely on entropy codec to further compress codes. Moreover, to achieve different compression rates it is unavoidable to train multiple models with different regularization parameters separately, which is often computationally intensive.

In this work, we are motivated to develop an architecture that has the following advantages. First, unlike classical distributed source coding (DSC) which requires customized code design for different scenarios (Xiong et al., 2004), a data-driven distributed compression framework can handle nontrivial distribution of image sources with arbitrary correlations. Second, the computation complexity of encoders (e.g. mobile devices) can be transferred to the decoder (e.g. a remote server). Such a system of low complexity encoders can be used in a variety of application domains, such as multi-view video coding (Girod et al., 2005), sensor networks (Xiong et al., 2004), and under-water image processing where communication bandwidth and computational power are quite restricted (Stojanovic & Preisig, 2009; Schettini & Corchs, 2010). Third, the distributed framework can be more robust against heterogeneous noises or malfunctions of encoders, and such robustness can be crucial in, e.g., unreliable sensor networks (Girod et al., 2005; Ishwar et al., 2005; Xiao et al., 2006). Last but not least, the architecture is naturally scalable in the sense that codes can be decoded at more than one compression quality level, and it allows efficient coding of correlated sources which are not physically co-located. This is especially attractive in video streaming applications (Guillemot et al., 2007; Gehrig, 2008).

It is tempting to think that splitting raw data into different encoders compromises the compression quality. It is thus natural to ask this question: Can distributed encoders perform as well as a single encoder trained with all data sources together? A positive answer from a theoretical perspective was

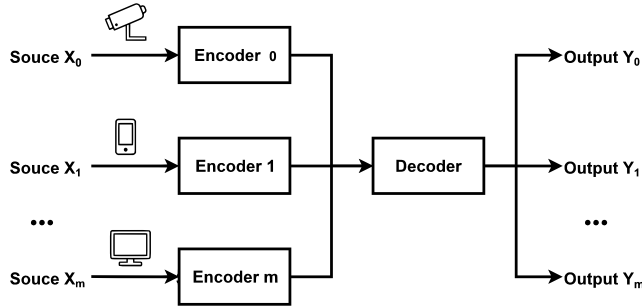


Figure 1: Illustration of Deep Distributed Source Coding.

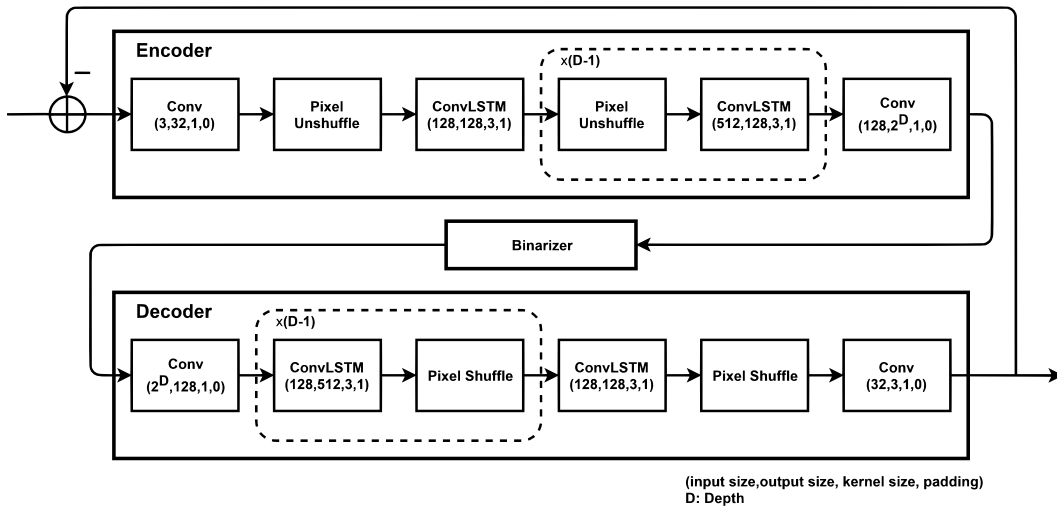
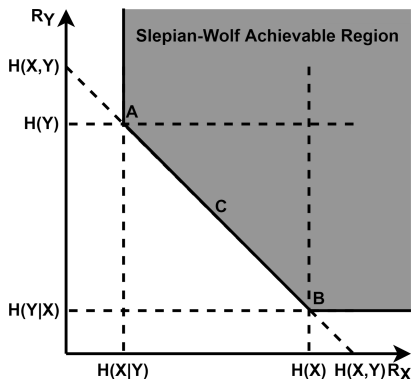


Figure 2: Illustration of Recurrent Autoencoder for Scalable Image Compression.

given in the context of information theory, where DSC is an important problem regarding the compression of multiple correlated data sources. The Slepian-Wolf Theorem shows that lossless coding of two or more correlated data sources with separate encoders and a joint decoder can compress data as efficiently as the optimal coding using a joint encoder and decoder (Slepian & Wolf, 1973; Cover, 1975). The extension to lossy compression with Gaussian data sources was proposed as Wyner-Ziv Theorem (Wyner & Ziv, 1976). Although these theorems were published in 1970s, it was after about 30 years that practical applications such as Distributed Source Coding Using Syndromes (DISCUS) emerged (Pradhan & Ramchandran, 2003). One of the main advantages of DSC is that the computation complexity of the encoder is transferred to the decoder. A system architecture with low complexity encoders can be a significant advantage in applications such as multi-view video coding and sensor networks (Girod et al., 2005; Xiong et al., 2004).

Motivated by the theoretical development of DSC, in this work we propose a DNN architecture that consists of distributed encoders and a joint decoder (illustrated in Fig. 1 and 2). We show that distributed encoders can perform as well as a single encoder trained with all data sources together. Our proposed DSC framework is data-driven by nature, and it can be applied to distributed data even with unknown correlation structure.

The paper is outlined below. We review previous related works in Section 2. We describe our proposed architecture for general image compression and its basic modules in Subsections 3.1-3.4. Then we elaborate the Deep Distributed Source Coding framework in Subsection 3.5. Experimental results are shown in Section 4, followed by conclusions in Section 5.

Figure 3: The Slepian-Wolf achievable region for two sources X and Y .

2 RELATED WORK

Though there has been a variety of research on lossy data compression in the past few decades, little attention has been paid to a systematic approach for general and practical distributed code design, especially in the presence of an arbitrary number of nontrivial data sources with arbitrary correlations (Xiong et al., 2004). A main motivation of this work is to attempt to replace the practical hand-crafted code design with data-driven approaches. To our best knowledge, what we propose is the first data-driven DSC architecture. Unlike hand-crafted quantizers, our neural network-based quantizers show that the correlations among different data sources can be exploited by the model parameters. Inspired by DSC, We empirically show that it is possible to approach the theoretical limit with our methodology.

2.1 IMAGE COMPRESSION WITH DEEP LEARNING

There exist a variety of classical codecs for lossy image compression. Although the JPEG standard (Wallace, 1992) was developed thirty years ago, it is still the most widely used image compression method. Several extensions to JPEG including JPEG2000 (Skodras et al., 2001), WebP (Google, 2010) and BPG (Bellard, 2014) have been developed. Most of these classical codecs rely on a quantization matrix applied to the coefficients of discrete cosine transform or wavelet transform.

Common deep neural network architecture for image compression are auto-encoders including non-recurrent autoencoders (Ballé et al., 2016; Theis et al., 2017; Li et al., 2018; Mentzer et al., 2018) and recurrent autoencoders (Toderici et al., 2015; 2017; Johnston et al., 2017). Non-recurrent autoencoders use entropy codec to encode quantized bottleneck representations, and recurrent models introduce incremental binarized codes at each compression quality. The generated codes of non-recurrent models is not scalable and their performance heavily relies on the conditional generative model like PixelCNN (Van den Oord et al., 2016) which arithmetic coding can take advantage of (Li et al., 2018; Mentzer et al., 2018). Recurrent autoencoders, on the other hand, can reconstruct images at lower compression qualities with the subset of high quality codes. Other notable variations include adversarial training (Rippel & Bourdev, 2017), multi-scale image compression (Nakanishi et al., 2018), and generalized divisive normalization (GDN) layers (Ballé et al., 2016).

Another challenge is to well define the derivative of quantizations of bottleneck representations. Ballé et al. (2016) replaced non-differentiable quantization step with a continuous relaxation by adding uniform noises. Toderici et al. (2015), on the other hand, used a stochastic form of binarization. discuss the performance of different quantization techniques in details.

2.2 DISTRIBUTED SOURCE CODING

Our methodology is inspired by the information-theoretic results on DSC which have been established since 1970s. The Slepian & Wolf (1973) Theorem shows that two correlated data sources encoded separately and decoded jointly can perform as well as joint encoding and decoding, and

outperform separate encoding and separate decoding. The striking result indicates that as long as the codes are jointly decoded, there can be no loss in coding efficiency even the codes are separately encoded. Cover (1975) generalizes the achievability of Slepian-Wolf coding to arbitrary number of correlated sources. Wyner & Ziv (1976) Coding gives a rate-distortion curve as an extension to lossy cases.

A classical illustration of Slepian-Wolf achievable region is shown in Fig. 3. We can achieve the performance of joint encoding and decoding of two data sources X and Y where the bit rate R is equal to the joint entropy $H(X, Y)$ with separate encoding and joint decoding. Specifically, the achievable region proved by the Slepian-Wolf Theorem is given by $R_X \geq H(X|Y)$, $R_Y \geq H(Y|X)$, and $R_X + R_Y \geq H(X, Y)$ as shown in the shaded area of Fig. 3. Here R and $H(\cdot)$ denote the bit rates and (conditional) entropies in classical Shannon theory. In practice, although some works are proposed to approach the mid-point C (Schonberg et al., 2004), the most widely used scheme is source coding with side information (syndrome bits) at the decoder (Pradhan & Ramchandran, 2003). This code design takes advantage of the corner points A and B which correspond to $H(X, Y) = H(Y) + H(X|Y)$ and $H(X, Y) = H(X) + H(Y|X)$ respectively.

Some researchers have also shown the applicability of DSC on still images (Dikici et al., 2005). In practical applications, low complexity video encoding benefits from the DSC framework which can transfer the complexity of encoder to decoder (Puri & Ramchandran, 2002; Aaron et al., 2002). Scalable Video Coding can also be incorporated with DSC (Xu & Xiong, 2006). These proposed methods indicate the feasibility of DSC in our problem setting.

3 METHODS

In this section, we first describe the recurrent autoencoder for scalable image compression used in our work. We will elaborate the basic modules including Convolutional Long short-term memory (ConvLSTM), Pixel (Un)Shuffle, and Binarizer used in our model. We will then describe how this Deep Learning architecture is used in Distributed Source Coding framework.

3.1 NETWORK ARCHITECTURE

Our compression network consists of an encoder, a binarizer, and a decoder. The activation function following each Convolutional Neural Network (CNN) module is tanh. For the first iteration of our model, the input images are initially encoded and transformed into $(-1, 1)$ by tanh activation function. Binary codes are quantized from bottleneck representations. The decoder then reconstructs images based on the received binary codes. Finally, we compute the residual difference between the original input images and the reconstructed output images. At the next iteration, the residual difference is feedback as the new input for our model. This procedure is repeated multiple iterations to gain more codes for better reconstruction performance. Therefore, the reconstructed images at each iteration are the sum of output reconstructions from previous and current iterations. The dependencies among iterations are modeled by recurrent models like ConvLSTM. We iterate 16 times to generate scalable codes. Compared to non-scalable codes which require new set of codes at each compression quality, scalable codes are able to reconstruct images at lower compression quality by using the subset of codes. This is especially attractive in video streaming applications (Guillemot et al., 2007; Gehrig, 2008).

Consider dataset $X = \{x\}^N$ consisting of N i.i.d. samples of some continuous or discrete variables x . The data generating process is unknown. Autoencoders for compression and reconstruction can be formulated in the following way. Data can be compressed with a neural network-based encoder $f(x; \theta)$ into quantized codes \tilde{z} and reconstructed with a decoder $g(\tilde{z}; \phi)$. We can binarize bottleneck representations z and control the compression quality by varying its channel sizes. The loss function $\mathcal{L}(x, \tilde{x})$ is minimized with respect to the model parameters θ and ϕ .

$$z = f(x; \theta), \tilde{z} = \text{Binarize}(z), \tilde{x} = g(\tilde{z}; \phi), \quad (1)$$

$$\text{Minimize } \mathcal{L}(x, \tilde{x}) \quad (2)$$

Deep recurrent autoencoder gradually increases compression quality by creating a correlated residual sequence from the difference between the input and output of our model. The advantage of recurrent model is that we can use a subset of generated codes to reconstruct images at lower compression qualities. Classical autoencoders, on the contrary, not only have to train multiple networks

with different penalty coefficients for rate-distortion loss but also have to generate different codes for different compression quality. Suppose T iterations are used, we can formulate the recurrent autoencoder in the following way.

$$z_t = f(x_t; \theta), \tilde{z}_t = \text{Binarize}(z_t), \quad (3)$$

$$\tilde{x}_t = g(\tilde{z}_t; \phi), x_{t+1} = x_t - \tilde{x}_t, \tilde{x}_1 = 0, \quad (4)$$

$$\text{Minimize } \frac{1}{T} \sum_{t=1}^T \mathcal{L}(x_1, \sum_{i=1}^t \tilde{x}_i). \quad (5)$$

3.2 CONVOLUTIONAL LONG SHORT-TERM MEMORY

As proposed by Xingjian et al. (2015), simply replacing the Fully Connected (FC) layer in LSTM with convolutional layer, ConvLSTM is able to capture the spatial structure in the temporal sequence. We do not add any bias term to our modules because the output of each layer is saturated by tanh. The key equations are shown below.

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1}), \quad (6)$$

$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1}), \quad (7)$$

$$c_t = f_t c_{t-1} + i_t \tanh(W_{xc} * x_t + W_{hc} * h_{t-1}), \quad (8)$$

$$o_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1}), \quad (9)$$

$$h_t = o_t \tanh(c_t). \quad (10)$$

The first and last layers of the encoder and decoder are feed-forward Convolutional Neural Network with tanh activations. The recurrent layers are all ConvLSTM networks. We do not replace CNN layer with ConvLSTM because the size of the feature maps of hidden state at shallow layers costs a mass amount of memory. From our various experimental studies, the performance does not degrade significantly because no convolutional layers are in between ConvLSTM layers.

3.3 PIXEL (UN)SHUFFLE

We resize feature maps with Pixel UnShuffle modules. Pixel Shuffle module is originally proposed by Shi et al. (2016) to tackle image and video super-resolution problem. Compared to transposed convolutional layers, Pixel Shuffle module is computationally efficient, because it is non-parametric and only requires tensor reshaping and dimension permutation (Shi et al., 2016). We note that although this method is used for upscaling, it is actually invertible and we propose to use its inversion for downscaling. Thus, the encoder and decoder can be constructed symmetrically. Our experimental results show that symmetric recurrent autoencoder architecture actually produces better results with less number of parameters, compared to the asymmetric architecture using transposed convolutional layers as proposed in Toderici et al. (2017). We describe the module with the following pseudocodes.

Algorithm 1 Pixel UnShuffle

Require: $X \sim (N, C, H, W), r_H, r_W$

Ensure: r_H, r_W are integers, divides H, W

Reshape $X \sim (N, C, H/r_H, r_H, W/r_W, r_W)$

Permute $X \sim (N, C, r_H, r_W, H/r_H, W/r_W)$

Reshape $X \sim (N, C \times r_H \times r_W, H/r_H, W/r_W)$

3.4 BINARIZER

The derivative of quantization function is only defined at the rounded integer itself. Therefore, we have to replace its derivative in the backward pass of backpropagation with a form of smooth approximation Rumelhart et al. (1988). Thanks to a thorough discussion of different alternative approaches by Theis et al. (2017), we choose to use the identity function to replace its derivatives that cannot be well defined as shown in 11. During training, we use a stochastic form of binarization proposed by Toderici et al. (2017). For bottleneck representations $z \in (-1, 1)$, the details of binarizer $\tilde{z} = \text{Binarize}(z)$ are described as follows.

Train

$$\tilde{z} = \text{Binarize}(z) = \begin{cases} 1, & \text{with probability } (z + 1)/2 \\ -1, & \text{otherwise} \end{cases}$$

$$\frac{d}{dz} \tilde{z} := \frac{d}{dz} \mathbb{E}(\tilde{z}) = \frac{d}{dz} \mathbb{E}(z) = 1 \quad (11)$$

Test

$$\text{Binarize}(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ -1, & \text{otherwise} \end{cases} \quad (12)$$

3.5 DEEP DISTRIBUTED SOURCE CODING FRAMEWORK

Fig. 1 and 2 illustrate our Distributed Recurrent Autoencoder for Scalable Image Compression (DRASIC). Similar to classical DSC framework, each data source is encoded separately and decoded jointly. In our network, each distributed encoder in Fig. 1 has the exact same structure in Fig. 2. Traditionally, researchers have to design different kind of codes for specific data sources (Schonberg et al., 2004). We propose to use data-driven approach to handle complex scenarios where the distribution of data sources is unknown and their correlations can be arbitrary. Our proposal may also shed new light on sophisticated application scenarios such as videos where data sources and correlations are time dependent.

In our neural network-based DSC, M distributed encoders encode corresponding data sources x^m that can be arbitrarily correlated. Each neural network-based encoder $f(x^m; \theta^m)$ has their own model parameters θ^m . After binarizing bottleneck representations z^m , code sources \tilde{z}^m are transmitted and concatenated batch-wisely. A single decoder $g(\tilde{z}^m; \phi)$ reconstructs images \tilde{x}^m from all sources with the same model parameters ϕ . In classical settings, the joint decoder has to process all compressed codes from each source jointly. In our data-driven setting, the joint training process optimizes the model such that the single decoder can decode from correlated sources. In this case, decoding codes from a particular data source does not depend on synchronization of codes from other sources, since the model has been optimized to adapt the correlations among all sources.

$$z_t^m = f(x_t^m; \theta^m), \tilde{z}_t^m = \text{Binarize}(z_t^m), \quad (13)$$

$$\tilde{x}_t^m = g(\tilde{z}_t^m; \phi), x_{t+1}^m = x_t^m - \tilde{x}_t^m, \tilde{x}_1^m = 0, \quad (14)$$

$$\text{Minimize } \frac{1}{MT} \sum_{t=1}^T \sum_{m=1}^M \mathcal{L}(x_1^m, \sum_{i=1}^t \tilde{x}_i^m). \quad (15)$$

Our result shows that the resulting distributed model can perform as well as encoding all data by one single encoder. However, if we encode and decode each data source separately, the performance becomes significantly worse, i.e. with $\tilde{x}_t^m = g(\tilde{z}_t^m; \phi^m)$.

4 EXPERIMENTS

To show our model is capable of compressing natural images, we train our model on CIFAR10 dataset (Stojanovic & Preisig, 2009) and evaluate the rate-distortion curve on Kodak dataset (Franzén, 2002). To show our model is capable of compressing grayscale images and demonstrate the feasibility of training encoders in a distributed manner, we train and evaluate our models with MNIST dataset LeCun et al. (1998). We observe that many non-recurrent autoencoders outperform recurrent models on rate-distortion curves (Li et al., 2018; Mentzer et al., 2018). We emphasize the distinction between the recurrent and non-recurrent autoencoders which do not have the scalability of reconstructing low quality images by using the subset of codes for high quality reconstruction. Our experiments aim to empirically demonstrate the feasibility of scalable distributed source coding in a data-driven setting. We use Adam optimizer (Kingma & Ba, 2014) with minibatch size of 100 for all experiments. We use learning rate 0.001 for a total of 200 epochs and decay every 50 epochs by a factor of 0.5. Our model uses a depth size $D = 3$ such that a 32×32 RGB image will be compressed into $8 \times 4 \times 4$ binarized codes at each iteration. We iterate our models 16 times to

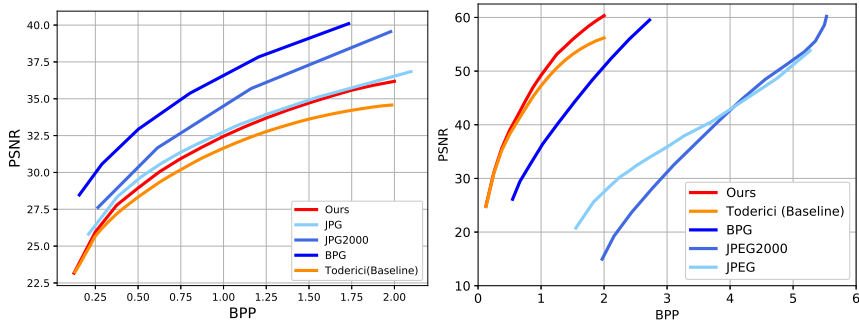


Figure 4: Our symmetric recurrent autoencoder performs comparable to classical codecs and neural network-based codecs for natural images and handwritten grayscale images.

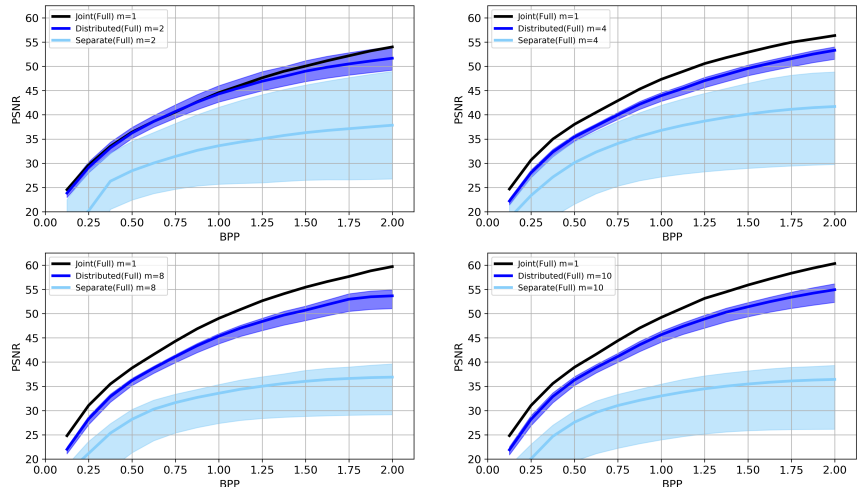


Figure 5: Rate-distortion curves for data sources distributed by class labels with $T = 16$ for all sources.

achieve the compression rates from 0.125 to 2 Bit Per Pixel (BPP). We empirically found that \mathcal{L}_1 loss performs much better than \mathcal{L}_2 and binary cross entropy loss. Fig. 4 shows that our symmetric recurrent autoencoder performs comparable to classical codecs and neural network-based codecs on compressing natural images, and performs significantly better on compressing handwritten grayscale images.

To demonstrate the feasibility of compressing distributed data sources, we split our data into correlated subsets to emulate the case where encoders only have access to distributed correlated data sources. We conduct our experiments with (2, 4, 8, 10) number of distributed sources. For the MNIST dataset, the correlated data sources are from images separated by class labels. Each data source only contains the images of the same digit. First, we compare our result, labeled as *Distributed*, to the case where all data are trained with one encoder and one decoder jointly, labeled as *Joint*. The *Joint* curve is approximated as the theoretical upper bound of performance. Second, we compare our result to the case where each data source is trained with a separate pair of encoder and decoder, labeled as *Separate*.

In our data-driven setting, decoding codes from a particular data source does not require synchronization of codes from all sources as in classical settings. In a classical DSC code design, if the syndrome bits $H(X|Y)$ are used and the data source Y is accidentally blocked, we will not be able to decode the data source X . We test each pair of distributed encoder and joint decoder separately to demonstrate the robustness of our framework. The solid curves are the average performance across all encoders. At each iteration, the confidence band is determined by the best and worst performance of all encoders. Fig. 5 and Fig. 6 show that distributed encoders not only dominate separately trained codecs but also have narrower confidence bands. As the number of encoders increases, the confi-

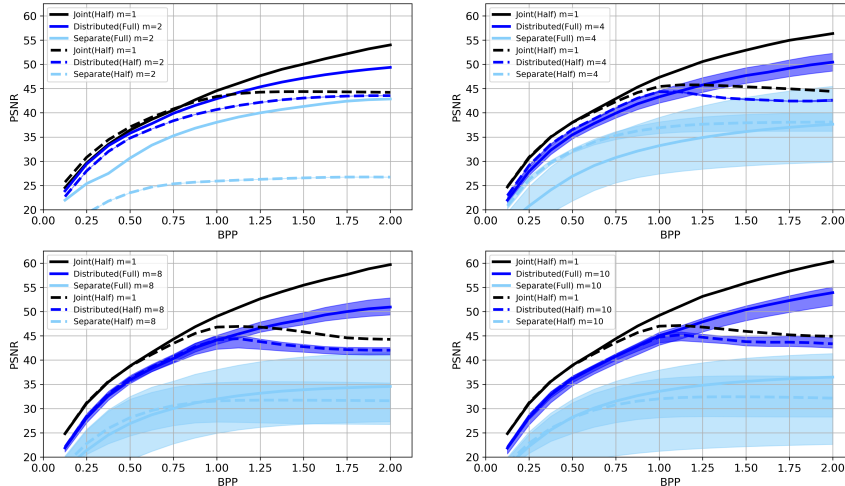


Figure 6: Rate-distortion curves for data sources distributed by class labels with $T = 16$ for the first half of sources and $T = 8$ for the second half of sources.

dence bands of separately trained codecs become wider because each separate codec can only access very limited amount of data and thus suffer from overfitting. As the number of iterations increases, the confidence bands also become wider. This is because the residual differences at a later iteration become less correlated.

We also demonstrate the performance of low complexity encoders which are trained with a smaller number of iterations. In Fig. 6, the first half of encoders is trained with 16 iterations, labeled as *Full* in solid lines, and the second half of encoders is only trained with 8 iterations, labeled as *Half* in dashed lines. For example, for $m = 8$, encoders 1 to 4 are trained with 16 iterations while encoders 5 to 8 are trained with 8 iterations. The theoretical limits (in black lines) are trained with all available data as in Fig. 5. Half complexity encoders perform as well as full complexity encoders in the first 8 iterations, because their dependencies of the first eight iterations are trained properly with the model parameters. After the eighth iteration, both full and half complexity encoders can still approach their theoretical limits. However, without specifically training dependencies after the eighth iterations, the performance of half complexity encoders slightly decreases.

5 CONCLUSION

We introduced a data-driven Distributed Source Coding framework based on Distributed Recurrent Autoencoder for Scalable Image Compression (DRASIC). Compared to classical code design, our method has the following advantages. First, instead of explicitly estimating the correlations among data sources in advance, we use data-driven approach to learn the dependencies with the neural network parameters. Given enough training data, our method can handle an arbitrary number of sources with arbitrary correlations. Second, we showed the robustness of our framework. Unlike classical code design which may require careful data source synchronization, each distributed encoder of our model, once trained and deployed, can be used independently of others because the dependencies are already learned by the model parameters. Third, as one of the most important applications of Distributed Source Coding, low complexity encoders were shown to be feasible based on our experimental results. Data sources trained with less data and fewer number of iterations can still approach the theoretical limit obtained by pulling all the data. Last but not least, our recurrent model can reconstruct images efficiently even at low compression quality.

We point out two interesting directions of future work. First, the compression quality of the proposed architecture may be improved by introducing spatially adaptive weights over different iterations, e.g. by using context models for adaptive arithmetic coding. Second, the network architecture may be further extended to handle time-dependent data sources.

REFERENCES

- Anne Aaron, Rui Zhang, and Bernd Girod. Wyner-Ziv coding of motion video. In *Signals, Systems and Computers, 2002. Conference Record of the Thirty-Sixth Asilomar Conference on*, volume 1, pp. 240–244. IEEE, 2002.
- Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.
- Fabrice Bellard. BPG image format, 2014. URL <https://bellard.org/bpg/>.
- T Cover. A proof of the data compression theorem of slepian and wolf for ergodic sources (corresp.). *IEEE Transactions on Information Theory*, 21(2):226–228, 1975.
- Cagatay Dikici, Radhouane Guermazi, Khalid Idrissi, and Atilla Baskurt. Distributed source coding of still images. In *Signal Processing Conference, 2005 13th European*, pp. 1–5. IEEE, 2005.
- Robert Franzén. Kodak lossless true color image suite: Photocd pcd0992. 2002.
- Nicolas Gehrig. *Distributed source coding of multi-view images*. PhD thesis, University of London, 2008.
- Bernd Girod, Anne Margot Aaron, Shantanu Rane, and David Rebollo-Monedero. Distributed video coding. *Proceedings of the IEEE*, 93(1):71–83, 2005.
- Google. WebP: Compression techniques, 2010. URL <https://developers.google.com/speed/webp/docs/compression>.
- Karol Gregor, Frederic Besse, Danilo Jimenez Rezende, Ivo Danihelka, and Daan Wierstra. Towards conceptual compression. In *Advances In Neural Information Processing Systems*, pp. 3549–3557, 2016.
- Christine Guillemot, Fernando Pereira, Luis Torres, Touradj Ebrahimi, Riccardo Leonardi, and Joern Ostermann. Distributed monoview and multiview video coding. *IEEE Signal Processing Magazine*, 24(5):67–76, 2007.
- Prakash Ishwar, Rohit Puri, Kannan Ramchandran, and S Sandeep Pradhan. On rate-constrained distributed estimation in unreliable sensor networks. *IEEE Journal on Selected Areas in Communications*, 23(4):765–775, 2005.
- Nick Johnston, Damien Vincent, David Minnen, Michele Covell, Saurabh Singh, Troy Chinen, Sung Jin Hwang, Joel Shor, and George Toderici. Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. *structure*, 10:23, 2017.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Mu Li, Wangmeng Zuo, Shuhang Gu, Debin Zhao, and David Zhang. Learning convolutional networks for content-weighted image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3214–3223, 2018.
- Dong Liu, Haichuan Ma, Zhiwei Xiong, and Feng Wu. CNN-based DCT-like transform for image compression. In *International Conference on Multimedia Modeling*, pp. 61–72. Springer, 2018.
- Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4394–4402, 2018.
- Ken M Nakanishi, Shin-ichi Maeda, Takeru Miyato, and Daisuke Okanohara. Neural multi-scale image compression. In *Asian Conference on Computer Vision*, pp. 718–732. Springer, 2018.

- S Sandeep Pradhan and Kannan Ramchandran. Distributed source coding using syndromes (DISCUS): Design and construction. *IEEE transactions on information theory*, 49(3):626–643, 2003.
- Rohit Puri and Kannan Ramchandran. PRISM: A new robust video coding architecture based on distributed compression principles. In *Proceedings of the annual allerton conference on communication control and computing*. Citeseer, 2002.
- Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2922–2930. JMLR. org, 2017.
- David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.
- Raimondo Schettini and Silvia Corchs. Underwater image processing: state of the art of restoration and image enhancement methods. *EURASIP Journal on Advances in Signal Processing*, 2010: 14, 2010.
- D Schonberg, Kannan Ramchandran, and S Sandeep Pradhan. Distributed code constructions for the entire slepian-wolf rate region for arbitrarily correlated sources. In *Data Compression Conference, 2004. Proceedings. DCC 2004*, pp. 292–301. IEEE, 2004.
- Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1874–1883, 2016.
- Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi. The JPEG 2000 still image compression standard. *IEEE Signal processing magazine*, 18(5):36–58, 2001.
- David Slepian and Jack Wolf. Noiseless coding of correlated information sources. *IEEE Transactions on information Theory*, 19(4):471–480, 1973.
- Milica Stojanovic and James Preisig. Underwater acoustic communication channels: Propagation models and statistical characterization. *IEEE communications magazine*, 47(1):84–89, 2009.
- Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017.
- George Toderici, Sean M O’Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. Variable rate image compression with recurrent neural networks. *arXiv preprint arXiv:1511.06085*, 2015.
- George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5306–5314, 2017.
- Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pp. 4790–4798, 2016.
- Gregory K Wallace. The JPEG still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.
- Aaron Wyner and Jacob Ziv. The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on information Theory*, 22(1):1–10, 1976.
- Jin-Jun Xiao, Alejandro Ribeiro, Zhi-Quan Luo, and Georgios B Giannakis. Distributed compression-estimation using wireless sensor networks. *IEEE Signal Processing Magazine*, 23(4):27–41, 2006.
- SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in neural information processing systems*, pp. 802–810, 2015.

Zixiang Xiong, Angelos D Liveris, and Samuel Cheng. Distributed source coding for sensor networks. *IEEE signal processing magazine*, 21(5):80–94, 2004.

Qian Xu and Zixiang Xiong. Layered Wyner-Ziv video coding. *IEEE Transactions on Image Processing*, 15(12):3791–3803, 2006.