Joint Source-Channel Coding with Neural Networks for Analog Data Compression and Storage

Ryan Zarcone[1], Dylan Paiton[2], Alex Anderson[3], Jesse Engel[4]
H.S. Philip Wong[4], and Bruno Olshausen[2]

[1]Biophysics, [2]Vision Science, [3]Physics
University of California, Berkeley
zarcone@berkeley.edu

[4]Electrical Engineering
Stanford University

Abstract

We provide an encoding and decoding strategy for efficient storage of analog data onto an array of Phase-Change Memory (PCM) devices. The PCM array is treated as an analog channel, with the stochastic relationship between write voltage and read resistance for each device determining its theoretical capacity. The encoder and decoder are implemented as neural networks with parameters that are trained end-to-end to minimize distortion for a fixed number of devices. To minimize distortion, the encoder and decoder must adapt jointly to the statistics of images and the statistics of the channel. Similar to Balle et al. (2017), we find that incorporating divisive normalization in the encoder, paired with de-normalization in the decoder, improves model performance. We show that the autoencoder achieves a rate-distortion performance above that achieved by a separate JPEG source coding and binary channel coding scheme. These results demonstrate the feasibility of exploiting the full analog dynamic range of PCM or other emerging memory devices for efficient storage of analog image data.

Introduction

With the rapid increase of internet-connected devices has come an increase in research aimed to design and characterize smaller and more power-efficient devices for information storage [1–3]. In particular, Phase Change Memory (PCM) is a type of memristive, non-volatile storage device that has gained considerable interest for both research and application in the last decade [4, 5]. These are 2-terminal, nanoscale circuit elements whose resistance can be changed in a continuous manner. PCMs have many desirable properties of storage devices: non-volatility, long endurance, low power consumption, and fast read/right.

We can characterize the theoretical storage capacity of a PCM device by treating it as an information channel. The channel capacity can be determined from the stochastic relationship between write voltage V and the resulting read resistance R, expressed in the conditional distribution $P(R|V)$. For the particular device used in this study the dependence of resistance on voltage is non-linear, as shown in Figure 1. (Complete details of the device characterization are available in [3].) In previous work [3], we estimated the capacity of the device to be 2.68 bits. Importantly, however, information theory does not provide a prescription for constructing a channel code that can achieve this hypothetical capacity limit. Furthermore, such a code would only be optimal in the limit of asymptotically long blocklengths, thus requiring potentially long delay and complexity.

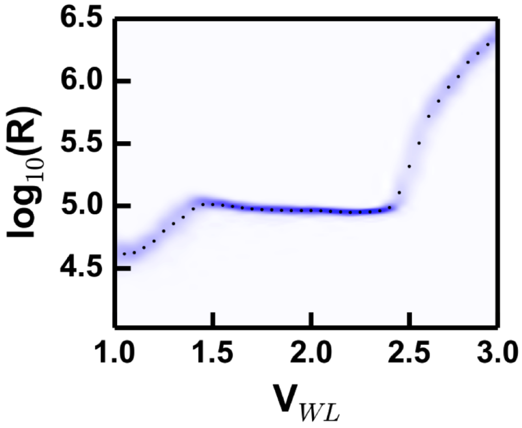


Figure 1: Conditional density, $P(R|V)$, for the PCM device, reproduced with permission from [3]. Darker color indicates higher probability density at that location. V is the voltage applied across the device, R is the resulting resistance.

For the purpose of this study, we utilize an idealized model of the device in that it has a single, fixed probabilistic mapping from voltage to resistance. Real devices exhibit other effects such as drift, cross talk, and device-to-device variation [6, 7]. While some solutions currently exist to mitigate these effects [8–10], we do not address them here. It should be noted, however, that the adaptive joint coding framework we describe has the potential to be robust to these effects by adaptively compensating for them.

Source-Channel Coding

Shannon's source separation theorem provides a proof showing the optimality of separate source and channel coding in the limit of infinite blocklength codes [11]. The process of transmission is split into two parts: (i) source coding, which removes the redundancy from a source, and (ii) channel coding, which adds specific redundancies to correct for errors introduced by the channel.

However, as the proof is non-constructive, there is no prescription for achieving the optimal source and channel codes. Importantly, source coding is often very difficult if the signal lives in a high-dimensional, non-linear space. A closely related open research question involving image compression is characterizing the high dimensional, non-convex space on which natural images lie [12]. The modern advent of large-scale deep learning models has pushed the image source coding field further by improving results without the need to understand directly the complex structure of the images [13–15], but this method has not been applied to joint source-channel coding. Instead, source and channel codes are developed separately and channel codes are typically tailored to particular channels, which can be difficult if the conditional distribution describing the channel is complicated.

The Shannon guarantee of coding optimality requires asymptotically long blocklengths (and subsequently potentially asymptotically long delays and complexity in

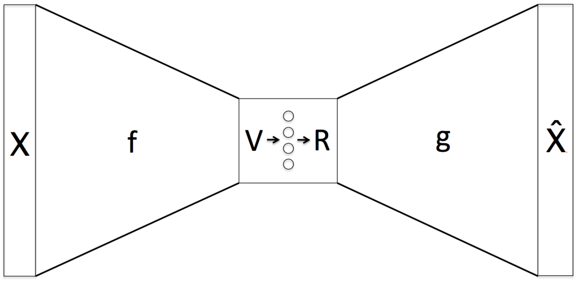


Figure 2: Diagram of the autoencoder architecture. X is the input to the network (an image). $f(\bullet)$ and $g(\bullet)$ are the encoding and decoding networks, respectively. The encoder network outputs a set of write voltages, V. This results in a set of noisy resistances R according to $P(R|V)$ (Figure 1). The decoder network then reconstructs the image $\hat{X}$ from the resistances R.

order to read-out/decode these block-lengths). In the regime of finite block-lengths, a joint source-channel coding strategy has been shown to be superior over separate coding [16,17]. For example, if the probability distributions for the source and channel are well-matched (often in terms of relative entropy), optimal transmission can be achieved with a joint scheme that involves little to no coding. The key here is that the data statistics, channel, and distortion function are considered simultaneously. This allows the coder to match noise characteristics between compression and the channel in such a way that the noise has a minimal effect on the distortion metric.

Our interest focuses on practical image storage on an array of analog memory devices. We assess performance at this task by looking at the rate-distortion trade-off. As a simple case, we could start by looking at a single PCM channel and determining what the optimal transformation and corresponding distortion for a univariate Gaussian signal would be. This was calculated previously by Engel et al. [18], who found that a joint coding, symbol-by-symbol transformation (i.e. a lookup table) is able to achieve the same performance as a realistic (non-asymptotic) separate coding system with optimal 2 bit scalar quantization and 2 bit channel coding (achievable for this particular channel with 8 read and write states). Thus, with reduced complexity and latency of the control circuitry, this joint scheme was able to perform on par with separate coding schemes. This result is encouraging, as it suggests that a proper joint scheme can perform well with minimal delay and complexity. The difficulty then lies in finding a 'proper' scheme. In the case described above, the problem was simple enough to be solved by exhaustive search. If, however, we desire to store something more complicated, such as an image, a different method is required.

Autoencoder Framework

An autoencoder neural network is typically trained to reconstruct an input pattern after passing through an information bottleneck. The bottleneck is often reduced in dimensionality compared to the input and output. For our case, the bottleneck is a fixed number of noisy information channels (PCMs), as illustrated in Figure 2.

Traditional image compression schemes such as JPEG and JPEG2000 can also be thought of as two-layer autoencoders. In this case the encoder and decoder are hand-designed to be the discrete cosine transform and its inverse (JPEG) or a wavelet transform and its inverse (JPEG2000). These source coders also have a non-trivial additional step where the latent representation (i.e. output of $f(\bullet)$ in Figure 2) is quantized and entropy coded. Following the standard pipeline, once an image is compressed with one of these source coders, the latent representation can be transformed via a channel coder to be passed through a particular channel. The signal received on the other side of the channel would then be decoded by a channel decoder and then a source decoder (i.e. $g(\bullet)$ in Figure 2) to be transformed back into a reconstruction of the input image.

Here we describe an end-to-end optimized solution for learning the encoder and decoder best suited for storing images on an array of PCM devices using a multi-layered autoencoder network. The objective of the network is to transform the input image via a series of linear/non-linear operations into a set of optimal write voltages for the PCM devices, and to transform the resulting read resistances to reconstruct the image with minimal distortion, measured as MSE.

To characterize the PCM channel, we constructed a set of conditional histograms using data obtained from [3] and modeled each one as a Gaussian distribution with mean and variance determined from the data. This provided a set of 40 discrete samples of $P\left(R|V\right)$, consisting of voltages v and the corresponding mean resistance $\mu\left(v\right)$ and its standard deviation $\sigma\left(v\right)$. To then estimate values in between those measured, we performed Gaussian interpolation (convolving the measured data with Gaussian kernels). Importantly, this non-parametric channel model is not device-specific - i.e. it can be adapted to any two-terminal device as long as $P\left(R|V=v\right)$ is approximately Gaussian $\forall$ v, as is the case for this device.

The weights of the autoencoder network were learned by backpropagating the gradient of the objective function (see below) through the network. The write voltages were given by the output of the encoder, $v = f(X)$. Each use of the channel can be thought of as drawing a sample from $P\left(R|V=v\right)$. However, to minimize the objective, we needed R to be a differentiable function of v. Thus, as our model for the channel was $P\left(R|V=v\right) = \mathcal{N}(\mu(v), \sigma(v))$, we reparameterized R so that it was differentiable: $R\left(v\right) = \mu\left(v\right) + \sigma\left(v\right) \cdot \epsilon$, $\epsilon \sim \mathcal{N}\left(0, 1\right)$. Though this may look like a model of a Gaussian channel, it is more general as both μ and σ are nonlinear functions of v.

Note that our usage is different than what is typical for a variational autoencoder [19], as we are not interested in unconditional sampling and we are not interpreting the distribution as a prior. Instead, our model is more analogous to a denoising autoencoder [20], where noise is added to the hidden representation.

Evaluation on Image Data

MNIST

We first evaluated the network on images of hand-written digits (MNIST) using a three-layer encoder and three-layer decoder. Both networks were feed-forward, fully-connected with rectified (ReLU [21]) activation functions. The objective to be minimized was a combination of reconstruction error and a regularization penalty on the activations of the latent space:

$$C = \left\langle \left\| X - \hat{X} \right\|_2^2 + \lambda \left[\max\left(0, v - V_{max}\right) + \max\left(0, V_{min} - v\right) \right] \right\rangle \tag{1}$$

Here, $\max\left(\bullet\right)$ puts a constraint on the latent space activity (for going outside V_{max} or V_{min}), λ establishes the relative importance of keeping the write voltages within the range of the device, and v (voltages) are the activation values of the final encoding layer (i.e. outputs of $f\left(\bullet\right)$). The full network was trained using ADAM [22] on this cost function.

Natural Images

We next turned to the problem of evaluating performance on natural images. Since the structure in natural images is significantly more complex than MNIST digits, it stands to reason that a different encoding and decoding strategy will be needed. Indeed, our initial attempts using the above network architecture for a set of natural images yielded poor results. Concurrently however, Balle et al. [13] published work employing a similar autoencoder framework, but instead of ReLUs they incorporated a divisive normalization non-linearity into each layer of the encoder, along with a denormalization non-linearity in each layer of the decoder.

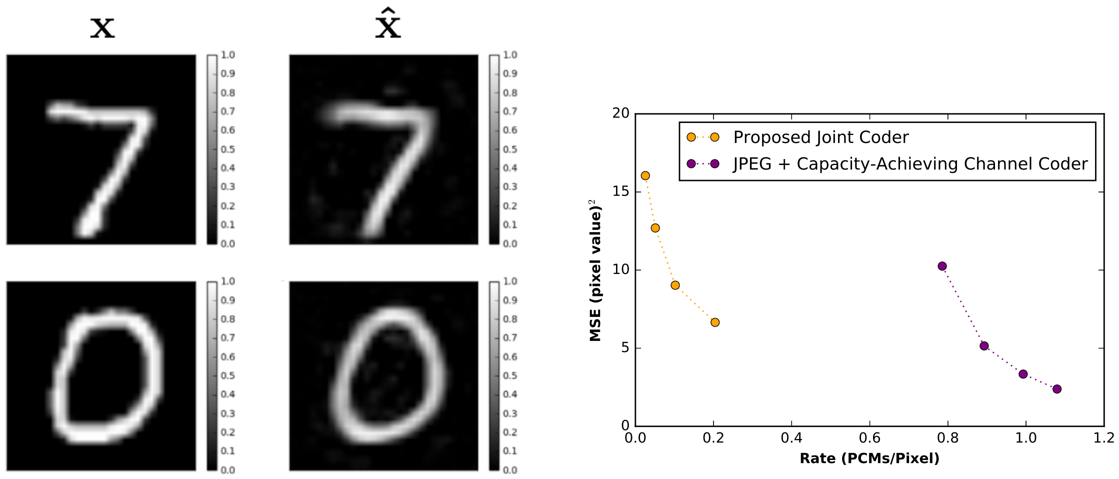
The nonlinearities in standard neural networks are typically pointwise ReLU, i.e. $a_i' = \max\left(0, a_i\right)$, where $a_i = \sum_j x_j \cdot w_{ij}$, with x indicating the layer's input and w indicating the weights. In contrast, divisive normalization is a population nonlinearity, whose functional form is given by

$$a_i' = \frac{a_i}{\sqrt{\beta_i^2 + \sum_k \gamma_{ik} a_k^2}} \tag{2}$$

where β and γ are learned parameters. Divisive normalization implements a local form of gain control that can reduce nonlinear dependencies [23].

The encoder and decoder weights in [13] are parameterized as filter convolutions at each layer, allowing the network to be scaled up to large images. Their network demonstrates impressive performance for compression over a binary channel.

The architecture we used was similar to that described in [13]. To adjust the compression rate of the network, we varied the number of units in the last layer of the encoder/first layer of the decoder (128 for the low compression, 30 for the high compression), while keeping the number of units in all other layers and the number of layers the same as in [13]. The output of the encoder is passed through a set of model PCM devices and their outputs are passed through the decoder. Thus, in contrast to [13], we are asking the network to perform both source and channel coding. Since it is presumably desirable to fill the full dynamic range of the PCM device, we also omitted the entropy cost in Balle et al.'s objective and utilized only the MSE and clipping constraints in Eq. 1. The training set consisted of $\sim 120,000$ images from the 2016 ImageNet test set [24] and the Flickr Creative Commons image set [25].



(a)

(b)

Figure 3: Results from storing 28×28 MNIST digits onto 40 PCM devices (giving a rate of 0.05 PCMs/Pixel) using the fully-connected network with ReLU point-wise non-linearities. (a) The images in the left column show two original images and in the right column their reconstructions. (b) Rate-distortion curve for the fully connected network. In purple, results are shown for the JPEG codec combined with a hypothetical channel coder that achieves a transmission rate of 2.68 bits (the channel capacity) across each PCM device. Dots indicate an average over a set of 10,000 images in the test set, dashes indicate interpolation.

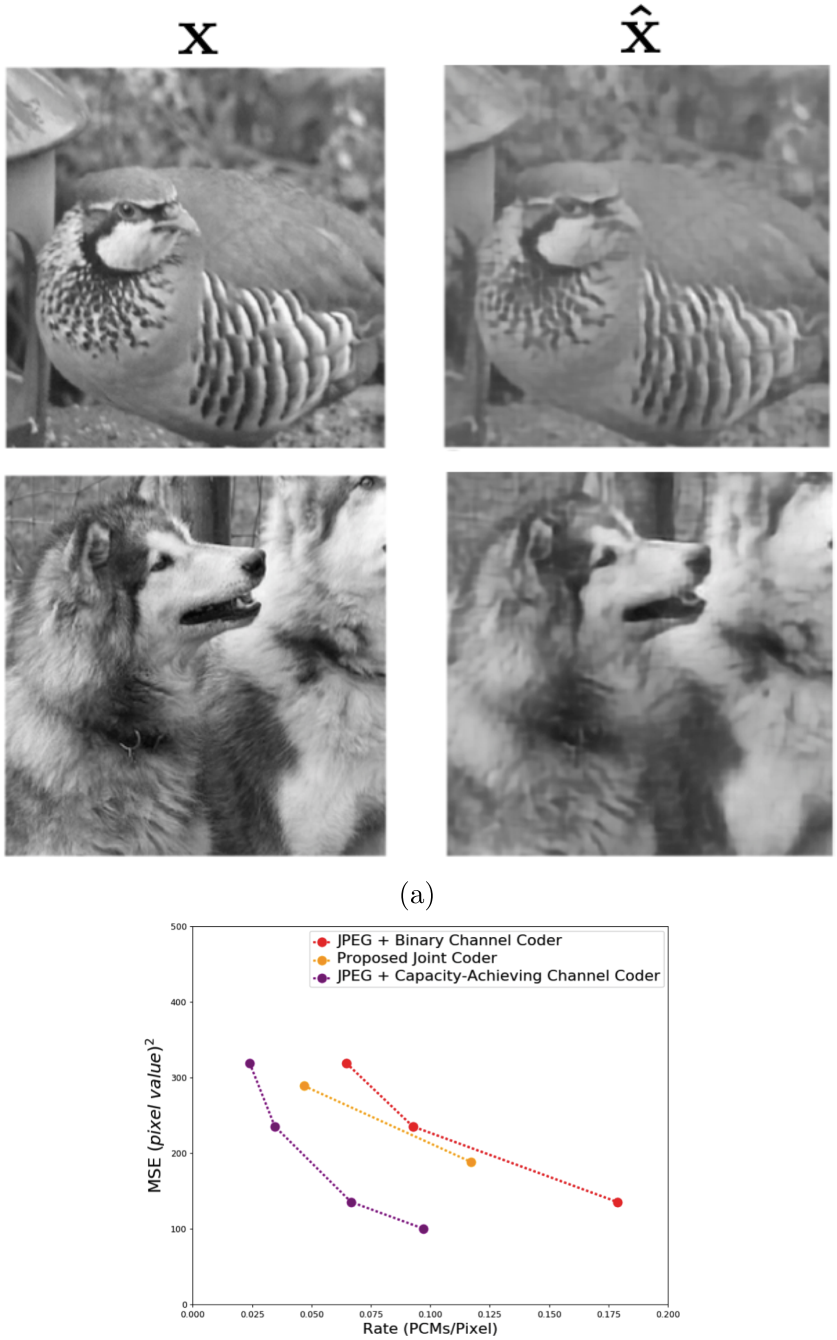
Results

MNIST

We compare performance of our method against separate source and channel coding by combining an existing source coder with hypothetical channel coders, as there currently does not exist a channel coder for this channel. We chose to use the JPEG codec for the source coder because it is the most commonly used source coder for natural images. For the channel coder, we used two different hypothetical coders: one that was able to achieve transmission rates of 1 bit across a PCM device – as is currently done with commercial versions of these devices (MNIST data not shown) – and one that was able to achieve transmission rates equal to the capacity, 2.68 bits. In order to produce the hypothetical rates, we first code each image using the JPEG codec, and then divide the number of bits by 1 or 2.68 for the binary and capacity-achieving channel coders, respectively. Using the fully trained network, we were able to store 28×28 pixel MNIST images more effectively than the JPEG source coder (which was not designed for MNIST-like images) combined with a theoretically optimal channel-coder (see Fig. 3). This is encouraging as it demonstrates that our method adapts well to data statistics.

Natural Images

To assess model performance on images of natural scenes, we used 24 gray-scale images from the Kodak dataset [26]. We measured two compression levels using our



(a)

(b)

Figure 4: Example storing 256 × 256 pixel natural images onto 7,680 PCM devices. (a) Original images (left) and their reconstructions (right). The distortion achieved was an MSE of ~ 200 (corresponding to PSNR of ~ 25 dB). (b) Rate-distortion curve for three different storage methods: In red and purple are results for the JPEG codec combined with two hypothetical channel coders that achieve transmission rates of (respectively) 1 bit and 2.68 bits across each PCM device. The proposed joint coder is shown in yellow. Dots indicate an average MSE achieved for 24 gray-scale images from the Kodak dataset.

proposed autoencoder framework. The higher capacity network (with 30 filters in the last encoding layer) used 7,680 model PCM devices to store each image, while the lower capacity network (with 12 filters in the last encoding layer) used 3,072 PCM devices. Results from this test are illustrated in Figure 4.

We found that the convolutional autoencoder was able to outperform the JPEG + binary channel coder method for both rates. Using the k-nearest neighbors algorithm introduced in [27], we estimated the achieved marginal information transmission rate across a subset of the PCM channels was ~ 1.5 bits/channel. This indicates that it would take more than 1.5 times as many PCM devices to store an image if the binary channel coder were used (as is the case in commercially deployed 3D-Xpoint[28]). For our network to achieve the same rate-distortion performance as the JPEG + capacity-achieving channel coder it would have either had to achieve a higher information transmission rate, or a higher compression rate, which we will discuss in the following section.

Discussion

The traditional paradigm of image compression focuses on designing source coders that process images into a string of bits. These bits are then taken by a channel coder that either transmits or stores them on a binary channel. In this work, we are addressing a novel setting for image compression in which the data is to be stored on an analog device. We are interested in these devices because of their benefits over traditional binary storage devices, chiefly: power-consumption, speed, and endurance. The questions we address are how to optimally store image data an on analog medium, and, more generally, how to optimally perform compression and error correction in this setting. The noise characteristics of PCMs and other analog storage devices are significantly different than traditional devices. Additionally, to optimally utilize PCM devices at their full capacity, it is necessary to develop an analog channel coding scheme [3, 18]. Instead of hand-designing analog channel coders for storing images on these devices – often an arduous task – we propose an adaptive autoencoder framework that accomplishes joint source-channel coding.

We find that our proposed joint source-channel coding scheme is able to achieve a rate-distortion performance that is superior to that achieved by JPEG combined with a binary channel coder. We simulated the input-output behavior of PCM using measured data from the devices. From this, our proposed network learned about the characteristics of PCMs and adapted the encoder and decoder to effectively use these devices for storing image data. Thus, in principle, our method can adapt to the statistics of a broad range of data types and memory devices, potentially even adapting to changes in device properties over time.

We believe there are several ways one could improve upon the proposed method. For the convolutional autoencoder, we noticed that there were substantial linear correlations between the outputs of the filters after the first layer. Thus, constructing a simple linear transformation to decorrelate these outputs (e.g. multiplying them by a matrix that diagonalizes their correlations) or incorporating a sparsifying nonlinearity [29] could remove additional redundancy, which should help with compression.

Additional challenges will arise when attempting to code images onto a physical PCM array. Specifically, the PCM devices in a fabricated array will likely not be completely independent, as cross-talk in the read/write process can modify their statistics. We simulated a set of uniformly behaving devices, while in reality each storage device would have somewhat different input-output characteristics. Finally, resistance drift will slowly change the properties of the PCM devices over time. Though the adaptive approach we have outlined can theoretically cope with these effects, it would require a considerable amount of retraining. More realistic extensions of the approach are possible for future work. For example, coupling the encoder/decoder to the PCM control circuitry that can adapt to chip specific statistics would likely alleviate some of the aforementioned concerns.

Acknowledgments

The authors would like to thank Yubei Chen, Charles Frye, and Daniel Rua for their valuable discussions and feedback.

References

[1] D. Evans, "The internet of things: how the next evolution of the internet is changing everything." *Cisco White Paper*, 2011.

[2] J. M. Rabaey and S. Malik, "Challenges and solutions for late-and post-silicon design," *IEEE Design & Test of Computers*, vol. 25, no. 4, 2008.

[3] J. H. Engel, S. B. Eryilmaz, S. Kim, M. BrightSky, C. Lam, H.-L. Lung, B. Olshausen, and H.-S. P. Wong, "Capacity optimization of emerging memory systems: A shannon-inspired approach to device characterization," in *Electron Devices Meeting (IEDM), 2014 IEEE International*. IEEE, 2014, pp. 29–4.

[4] H.-S. P. Wong, S. Raoux, S. Kim, J. Liang, J. P. Reifenberg, B. Rajendran, M. Asheghi, and K. E. Goodson, "Phase change memory," *Proceedings of the IEEE*, vol. 98, no. 12, pp. 2201–2227, 2010.

[5] S. W. Fong, C. M. Neumann, and H.-S. P. Wong, "Phase-change memory–towards a storage-class memory," *IEEE Transactions on Electron Devices*, 2017.

[6] A. Pirovano, A. L. Lacaita, F. Pellizzer, S. A. Kostylev, A. Benvenuti, and R. Bez, "Low-field amorphous state resistance and threshold voltage drift in chalcogenide materials," *IEEE Transactions on Electron Devices*, vol. 51, no. 5, pp. 714–719, 2004.

[7] I. Karpov, M. Mitra, D. Kau, G. Spadini, Y. Kryukov, and V. Karpov, "Fundamental drift of parameters in chalcogenide phase change memory," *Journal of Applied Physics*, vol. 102, no. 12, p. 124503, 2007.

[8] N. Papandreou, H. Pozidis, T. Mittelholzer, G. Close, M. Breitwisch, C. Lam, and E. Eleftheriou, "Drift-tolerant multilevel phase-change memory," in *Memory Workshop (IMW), 2011 3rd IEEE International*. IEEE, 2011, pp. 1–4.

[9] M. Stanisavljevic, A. Athmanathan, N. Papandreou, H. Pozidis, and E. Eleftheriou, "Phase-change memory: Feasibility of reliable multilevel-cell storage and retention at elevated temperatures," in *Reliability Physics Symposium (IRPS), 2015 IEEE International*. IEEE, 2015, pp. 5B–6.

[10] A. Athmanathan, M. Stanisavljevic, N. Papandreou, H. Pozidis, and E. Eleftheriou, "Multilevel-cell phase-change memory: A viable technology," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 6, no. 1, pp. 87–100, 2016.

[11] C. E. Shannon and W. Weaver, *The mathematical theory of communication.* University of Illinois press, 1998.

[12] B. Culpepper and B. A. Olshausen, "Learning transport operators for image manifolds," in *Advances in neural information processing systems*, 2009, pp. 423–431.

[13] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," *arXiv preprint arXiv:1611.01704*, 2016.

[14] O. Rippel and L. Bourdev, "Real-time adaptive image compression," *arXiv preprint arXiv:1705.05823*, 2017.

[15] F. Jiang, W. Tao, S. Liu, J. Ren, X. Guo, and D. Zhao, "An end-to-end compression framework based on convolutional neural networks," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[16] 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, no. 5, pp. 1147–1158, 2003.

[17] V. Kostina and S. Verdú, "Lossy joint source-channel coding in the finite blocklength regime," *IEEE Transactions on Information Theory*, vol. 59, no. 5, pp. 2545–2575, 2013.

[18] J. H. Engel, S. B. Eryilmaz, S. Kim, M. BrightSky, C. Lam, H.-L. Lung, B. A. Olshausen, and H.-S. P. Wong, "Opportunities for analog coding in emerging memory systems," *arXiv preprint arXiv:1701.06063*, 2017.

[19] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[20] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. Dec, pp. 3371–3408, 2010.

[21] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.

[22] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[23] E. P. Simoncelli and B. A. Olshausen, "Natural image statistics and neural representation," *Annual review of neuroscience*, vol. 24, no. 1, pp. 1193–1216, 2001.

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

[25] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "Yfcc100m: The new data in multimedia research," *Communications of the ACM*, vol. 59, no. 2, pp. 64–73, 2016.

[26] "Rpi kodak image dataset," http://www.cipr.rpi.edu/resource/stills/kodak.html.

[27] A. Kraskov, H. Stögbauer, and P. Grassberger, "Estimating mutual information," *Physical review E*, vol. 69, no. 6, p. 066138, 2004.

[28] F. T. Hady, A. Foong, B. Veal, and D. Williams, "Platform storage performance with 3d xpoint technology," *Proceedings of the IEEE*, vol. 105, no. 9, pp. 1822–1833, 2017.

[29] C. J. Rozell, D. H. Johnson, R. G. Baraniuk, and B. A. Olshausen, "Sparse coding via thresholding and local competition in neural circuits," *Neural computation*, vol. 20, no. 10, pp. 2526–2563, 2008.