Generative Compression

Shibani Santurkar 1  David Budden 1  Nir Shavit 1

Abstract

Traditional image and video compression algorithms rely on hand-crafted encoder/decoder pairs (codecs) that lack adaptability and are agnostic to the data being compressed. Here we describe the concept of generative compression, the compression of data using generative models, and show its potential to produce more accurate and visually pleasing reconstructions at much deeper compression levels for both image and video data. We also demonstrate that generative compression is orders-of-magnitude more resilient to bit error rates (e.g. from noisy wireless channels) than traditional variable-length entropy coding schemes.

1. Introduction

Graceful degradation is a quality-of-service term used to capture the idea that, as bandwidth drops or transmission errors occur, user experience deteriorates but continues to be meaningful. Considering the example of a streaming tennis match, this might mean preserving the quality of the players and court while losing distinguishing features of the crowd. Traditional compression techniques, such as JPEG and MPEG, are agnostic to the data being compressed and do not degrade gracefully. This is shown in Figure 1, which compares the original image of a face (a) to its JPEG2000-compressed representation (b). Building upon the ideas of (Gregor et al., 2016) and the recent promise of deep generative models (Goodfellow et al., 2014), this paper presents a framework for generative compression of image and video data. As seen in Figure 1(c), this direction shows great potential for compressing data so as to provide graceful degradation, and to do so at bandwidths far beyond those reachable by traditional techniques.

There are two main categories of data compression, descriptively named lossless and lossy. The former problem traditionally involved deriving codes for discrete data given knowledge of their underlying distribution, the entropy of

which imposes a bound on achievable compression (Shannon, 1948). To deliver graceful degradation, we focus on the relaxed problem of lossy compression, where we believe there is potential for orders-of-magnitude improvement using generative compression compared to existing algorithms. We propose two arguments in support of this claim. First, from a bottom-up perspective, consider the string \( s = \text{grass tennis court} \). This string contains just a few bytes of information, and yet the detail and vividness of your mental reconstruction is astounding. (Defining a meaningful perceptual loss of your tennis court with respect to mine is a difficult problem (Sønderby et al., 2016) and an area of active research (Laparra et al., 2016)).

Second, from a top-down perspective, an MNIST-style 28x28 grayscale image can represent many more unique images than there are atoms in the universe. How small of a region of this space is spanned by plausible MNIST samples? If we could translate even a small fraction of this perceptual redundancy into a reduction in code verbosity, we would likely have solved the world’s data storage and transmission problems into the foreseeable future.

Lossy compression has traditionally been formulated as a rate-distortion optimization problem. In this framework, an analysis transform, \( f : \mathbb{R}^N \rightarrow \mathbb{R}^M \), maps input data \( \mathbf{x} \) (e.g. a vector of pixel intensities) to a vector \( \mathbf{z} \) in latent code space, and a synthesis transform, \( g : \mathbb{R}^M \rightarrow \mathbb{R}^N \), maps \( \mathbf{z} \) back into the original space. Compression is achieved by (lossy) quantization of \( \mathbf{z} \) followed by lossless compression using an entropy coding scheme (e.g. (Huffman, 1952)). Compression in this form seeks to minimize both the rate of the latent code, lower-bounded by the entropy of its distribution, and the \( \beta \)-prioritized distortion of the output,
\[ \hat{x} = g([f(x)]), \] with respect to the input, \( x \):

\[ L_{RD}(x, \hat{x}) = \beta \left( L(x, \hat{x}) - \log Q([f(x)]) \right), \tag{1} \]

where \( Q : \mathbb{Z}^M \rightarrow [0, 1] \) assigns bit-strings to quantized representations based on frequency (entropy coding) and \([\cdot]\) indicates the quantization operation (Theis et al., 2017). The distortion loss, \( L \), is typically a signal-to-noise ratio or structural similarity (SSIM) metric (Wang et al., 2004).

Joint optimization over rate and distortion has long been considered an intractable problem for images and other high-dimensional spaces (Gersho & Gray, 1992). Attention has instead been focused on hand-crafting encoder/decoder pairs (codecs) that apply linear analysis and synthesis transforms, e.g. discrete cosine transforms (JPEG) and multi-scale orthogonal wavelet decomposition (JPEG 2000). There are several limitations to this approach. There is no reason to expect that a linear function is optimal for compressing the full spectrum of natural images. Even presuming they are optimal for a particular class of bitmap images, this performance is unlikely to generalize to emerging media formats, and the development and standardization of new codecs has historically taken many years.

A pleasing alternative is to replace hand-crafted linear transforms with artificial neural networks, i.e. replacing the analysis transform with a learnt encoder function, \( z = f_\theta(x) \), and the synthesis transform with a learnt decoder function, \( \hat{x} = g_\phi(z) \). One noteworthy example of using neural nets in a rate-distortion optimization framework is the compressive autoencoder (CAE) model presented by (Theis et al., 2017), which derives differentiable approximations for quantization and entropy rate estimation to allow end-to-end training by gradient backpropagation. (Ballé et al., 2016) achieve a similar result, using a joint nonlinearity as a form of gain control. These and similar unsupervised models are showing promising results in both lossless (van den Oord et al., 2016; Theis & Bethge, 2015) and lossy data compression (Gregor et al., 2016; Toderici et al., 2016), assisted by the practically unlimited volume of data for their training.

2. Generative Models for Image Compression

Here we make liberal use of standard terminology (e.g. “manifold”) to explain our compression methodology. Consider the top-down argument presented in Section 1, such that \( X \) is the 784-dimensional space of 28x28 grayscale images. Let us further assume that all MNIST samples of handwritten digits cluster together within some smaller subspace, \( Z \). Presented with a new MNIST vector, \( x \), one approach to compression is to represent \( x \) by the lower-dimensional coordinate, \( z \), of its orthogonal projection onto manifold \( Z \). Reconstructing the original image is the reverse process of mapping \( z \) to \( \hat{x} \in X \). The intuition behind this framework is that the distortion loss, \( L(x, \hat{x}) \), measures how poorly \( x \) is captured by our understanding of the latent structure of the manifold \( Z \) of MNIST images.

Generative modeling provides a convenient framework for implementing the above idea. Imagine that the role of the receiver is simply to synthesize some realistic looking MNIST sample. If we knew the true distribution, \( P(x) \), of MNIST characters defined over \( X \), we could simply sample \( \hat{x} \in \mathbb{R}^N \) from this distribution. Unfortunately, it is intractable to accurately estimate this density function for such a high-dimensional space (Salakhutdinov & Hinton, 2009). One remedy to this problem is to factorize \( P(x) \) as the product of conditional distributions over pixels. This sequence modeling problem can be solved effectively using autoregressive models (Bengio & Bengio, 1999; Larochelle & Murray, 2011) of recurrent neural networks (Graves & Schmidhuber, 2009), allowing the generation of high-quality images or in-filling of partial occlusions (van den Oord et al., 2016). However, these models forego a latent representation and as such do not provide a mechanism for decoding an image from a specific code.

To implement our decoder, we can instead apply a generator function, \( \hat{x} = g_\phi(z) \), to approximate \( P(x) \) as the transformation of some prior latent distribution, \( P(z) \). To generate realistic-looking samples, we wish to train \( g \) to minimize the difference between its distribution, \( P_\phi(x) = \mathbb{E}_{\mathbf{z} \sim P(z)} [P_\phi(x|\mathbf{z})] \), and the unknown true distribution, \( P(x) \). A popular solution to this problem is to introduce an auxiliary discriminator network, \( d_\phi(x) \), which learns to map \( x \) to the probability that it was sampled from \( P(x) \) instead of \( P_\phi(x) \) (Goodfellow et al., 2014). This framework of generative adversarial networks (GANs) simultaneously learns \( \phi \) and \( \theta \) by training against the minimax objective:

\[ \mathbb{E}_{x \sim P(x)} [\log d(x)] + \mathbb{E}_{z \sim P(z)} [\log(1 - d(g(z))]. \]

The authors showed that this objective is equivalent to minimizing the Jensen-Shannon divergence between \( P_\phi(x) \) and \( P(x) \) for ideal discriminators. Although GANs provide an appealing method for reconstructing quality images from their latent code, they lack the inference (encoder) function \( f : X \rightarrow Z \) necessary for image compression. Points can be mapped from \( Z \) to \( X \), but not vice versa.

An alternative to GANs for generative image modeling are variational autoencoders (Kingma & Welling, 2013). Similar to GANs, VAEs introduce an auxiliary network to facilitate training. Unlike GANs, this inference function, \( z = f_\theta(x) \), is trained to learn an approximation, \( Q_\theta(x|z) \), of the true posterior, \( P(z|x) \), and thus can be used as an encoder for image compression. This is achieved by maximizing the log-likelihood of the data under the generative
model in terms of the variational lower bound, $L(\theta, \phi; x)$:

$$
\mathbb{E}_{z \sim Q_\phi(x|x)} [\log P_\theta(x|z)] - D_{KL}(Q_\phi(z|x)||P(z)),
$$

where $D_{KL}$ is the Kullback-Leibler divergence of the approximate and true posteriors. This resembles the rate-distortion objective presented in Equation (1), and the two have been demonstrated as equivalent under certain constraints (Ballé et al., 2016).

Recent studies have demonstrated the potential of VAEs for compression by training $P_\theta(x|z)$ and $Q_\phi(z|x)$ as deep neural networks (Gregor et al., 2016). The authors report that they do not build an actual compression algorithm, but present sample reconstructions with perceptual quality similar to JPEG2000. For this class of models, the most common conditional likelihood is the Gaussian distribution:

$$
P_\theta(x|z) = \mathcal{N}(x|\mu_\theta(z), \text{diag}(\sigma^2_\theta(z))),
$$

where $\mu_\theta(z)$ is implemented as a convolutional neural network (ConvNet) and $\sigma^2_\theta(z)$ is a vector of variance parameters. A well-established limitation of these models is that maximizing this likelihood is equivalent to minimizing the $L^2$ loss between pixel intensity vectors. This loss is known to correlate poorly with human perception and attenuates the high-frequency image components, resulting in blurry reconstructions (Larsen et al., 2015; Theis & Bethge, 2015).

### 3. Neural Codes for Generative Compression

To implement an effective neural codec for image compression, we wish to replicate the paired encoder/decoder interface of a VAE while generating the higher-quality images expected of a GAN. We propose a simple neural codec (NCode) architecture that approaches this in two stages. First, a feed-forward decoder network, $g_\phi : Z \rightarrow X$, is greedily pre-trained using an adversarial loss with respect to the auxiliary discriminator network, $d_\phi : X \rightarrow [0, 1]$. For this stage, $g_\phi$ and $d_\phi$ are implemented using DCGAN-style ConvNets (Radford et al., 2015). Second, the parameter vector $\phi$ is frozen and an encoder network, $f_\theta : X \rightarrow Z$, trained to minimize some distortion loss, $L(x, g_\phi(f_\theta(x)))$. We also investigate methods for lossy quantization of $z$, motivated by recent studies demonstrating the robustness of deep neural nets to reduced numerical precision (Gupta et al., 2015; Hubara et al., 2016; Rastegari et al., 2016). The overall NCode architecture is presented in Figure 2.

The rationale behind the NCode model is straightforward. Traditional image compression algorithms have been crafted to minimize pixel-level loss metrics such as mean squared error (MSE) or cross entropy. Although optimizing for MSE can lead to good peak signal-to-noise ratio (PSNR) characteristics, the resulting images are perceptually implausible due to a depletion of high-frequency components (blurriness) (Ledig et al., 2016). By adversarially pre-training the decoder, the end-to-end codec is constrained to producing reconstructions more likely to fool the more frequency-sensitive auxiliary discriminator.

To further improve the plausibility of our reconstructed images, we also choose to enrich the distortion loss with an additional measure of perceptual quality. Recent studies have indicated that textural information of an image is effectively captured by the feature maps of deep ConvNets pretrained for object recognition (Gatys et al., 2015). Perceptual loss metrics derived from these features have been used to improve the plausibility of generative image models (Dosovitskiy & Brox, 2016) and successfully applied to applications including super-resolution (Ledig et al., 2016) and style transfer (Gatys et al., 2016). We take a similar approach in NCode, modeling the distortion between image, $x$, and reconstruction, $\hat{x} = g_\phi(f_\theta(x))$, as the weighted sum of pixel-level and perceptual losses:

$$
L(x, \hat{x}) = \lambda_1 ||x - \hat{x}||_2 + \lambda_2 ||\text{conv}_4(x) - \text{conv}_4(\hat{x})||_2,
$$

where $\text{conv}_4$ is the fourth convolutional layer of an ImageNet-pretrained AlexNet (Krizhevsky et al., 2012; Zhu et al., 2016). The full NCode training algorithm is described in Algorithm 1 with minibatch SGD updates for simplicity, though in practice we found the Adam optimizer (Kingma & Ba, 2014) to yield better performance. It is important to recognize the similarity of the NCode architecture to many existing models. Recent studies have proposed hybrid models that combine VAEs with GANs (Dosovitskiy & Brox, 2016; Lamb et al., 2016; Larsen et al., 2015). Our model is conceptually simpler than many of these variants, as we adopt a greedy pre-training strategy versus a hybrid loss function and a deterministic autoencoder in place of a probabilistic VAE. We make no claim that these simplifications improve the theoretical properties of our model – simply that it was designed with a specific application (compression) in mind, and to our knowledge we are the first to successfully apply an adversarially-trained neural codec to this domain.

Our NCode architecture was largely motivated by the work of (Zhu et al., 2016), where a similar framework was applied for on-manifold photo editing. Similarly, other recent
The traditional limitation of this approach is that interpola-
tion is to transmit only every

The simplest method of capturing temporal redun-
dations in natural video data (Budden et al., 2016; Shi et al.,

Ever, this approach fails to capture the rich temporal corre-
sion in pixel-space yields visually displeasing results. Bor-
rowing a popular example, interpolating between the same
object on both sides of the room should result in that ob-
ject in the middle of the room, but instead produces two
half-transparent objects on both sides.

Instead, we choose to model a video sequence as
uniformly-spaced samples along a path, T, on the mani-
fold, Z (Figure 3). As described in Section 2, we assume
that that Z is a lower-dimensional embedding of some lat-
ent image class, e.g. grass tennis courts or natural images
more generally. We further assume that for sufficiently
small N, the path x(t) → x(t+N) can be approximated
by a geodesic and thus the intermediate frames can be re-
constructed by linear interpolation on Z. This assumption
builds on the wealth of recent literature demonstrating that
interpolating on manifolds learnt by generative models pro-
duce perceptually cohesive samples, even between quite
dissimilar endpoints (Brock et al., 2016; Radford et al.,
2015; Zhu et al., 2016).

Similar to MPEG, we can further compress a video se-
cquence through delta and entropy coding schemes. Each
latent vector is transmitted as its delta with respect to the
previous transmitted frame, δ(t+N) = z(t+N) − z(t). We
observe that this representation gains far more from en-
tropy coding (specifically (Huffman, 1952)) than for indi-
vidual latent vectors sampled from P(z), leading to a fur-
ther (lossless) reduction in bitrate. We do not use entropy
coding for image NCode.

4. Experiments

Traditional compression benchmarks (e.g. the Kodak Pho-
toCD dataset (Kodak, 1999)) lack the scale necessary
for training deep generative models. Instead, we choose
to benchmark the performance of JPEG, JPEG2000 and
our proposed NCode method using the popular CIFAR-10
dataset (Krizhevsky, 2009). Beyond data volume, the ad-

dvantages of CIFAR are (a) that each example has a ground-
truth class label (useful for validation), and (b) that it is

3.1. Generative Video Compression

Extending neural codecs to video compression is straight-
forward. A video is simply a sequence of images, so im-
ages can be compressed and transmitted frame-by-frame
using NCode. This is reminiscent of the motion-JPEG
scheme in traditional video compression literature. How-
ever, this approach fails to capture the rich temporal corre-
lations in natural video data (Budden et al., 2016; Shi et al.,
2016). Instead, we would prefer our model to be inspired
by the interpolation (bidirectional prediction) scheme in-
troduced for the popular MPEG standard.

The simplest method of capturing temporal redu-
dancy is to transmit only every N-th frame, X =
[X(t), x(t+N), x(t+2N), ...], requiring the receiver to in-
terpolate the missing data with a small N-frame latency.
The traditional limitation of this approach is that interpola-

Algorithm 1 Image NCode training with minibatch SGD

\textbf{params:} learning rate \( \alpha \), batch size \( m \), step ratio \( k \)

\begin{algorithm}
\textbf{while} \( \phi \) has not converged \textbf{do}
\hspace{1em} \textbf{for} \( k \) steps \textbf{do}
\hspace{2em} Sample \( \{x^{(i)}\}_{i=1}^{m} \sim P(x) \)
\hspace{2em} Sample \( \{z^{(i)}\}_{i=1}^{m} \sim U(-1, 1) \)
\hspace{2em} \( L_d \leftarrow \sum_{i=1}^{m} \left[ \log d_{\phi}(x^{(i)}) + \log \left( 1 - d_{\phi}(g_{\phi}(z^{(i)})) \right) \right] \)
\hspace{2em} \( \theta \leftarrow \theta + \frac{\alpha}{m} \nabla \theta L_d \)
\hspace{1em} \textbf{end for}
\hspace{1em} Sample \( \{z^{(i)}\}_{i=1}^{m} \sim U(-1, 1) \)
\hspace{1em} \( L_g \leftarrow \sum_{i=1}^{m} \log \left( 1 - d_{\phi}(g_{\phi}(z^{(i)})) \right) \)
\hspace{1em} \( \phi \leftarrow \phi + \frac{\alpha}{m} \nabla \phi L_g \)
\hspace{1em} \textbf{end while}

\textbf{while} \( \theta \) has not converged \textbf{do}
\hspace{1em} Sample \( \{x^{(i)}\}_{i=1}^{m} \sim P(x) \)
\hspace{1em} \( z^{(i)} \leftarrow f_{\theta}(x^{(i)}) \quad \forall i = 1, \ldots, m \)
\hspace{1em} \( \theta \leftarrow \theta - \frac{\alpha}{m} \nabla \theta \sum_{i=1}^{m} L \left( x^{(i)}, g_{\phi}(f_{\theta}(x^{(i)})) \right) \)
\hspace{1em} \textbf{end while}
\end{algorithm}

studies have augmented GANs with inference functions, e.g.
adversarial feature learning (Donahue et al., 2016) and
adversarially learned inference (ALI) (Dumoulin et al.,
2016). The ALI study also describes a similar compression
pipeline, passing images through a paired encoder/decoder
and assessing the reconstruction quality. Although the ALI
generator produces high-quality images, they differ dra-

matically in appearance from the input. Unlike our NCode
model, the authors attribute this to a lack of explicit pixel-
level distortion in their optimization objective.

Figure 3. MCode procedure for video compression.
Figure 4. A comparison of graceful degradation of reconstructed images under various compression techniques: (a) Randomly sampled images from the CIFAR-10 dataset (Top panel) (Krizhevsky, 2009), CelebA dataset (Middle panel) (Liu et al., 2015) and Zappos50k dataset (Bottom panel) (Yu & Grauman, 2014). Rows (b-c) demonstrate the corresponding reconstructions, compression ratios and PSNR/SSIM metrics (averaged over the full test set) for JPEG and JPEG2000. Corresponding NCode performance is shown for varying quantization, i.e. (d) 16 bit, (e) 4 bit, and (f) 2 bit representations of the latent vector $z$. 

CIFAR-10
19 bits/px
JPEG2000
9x compression
JPEG
8x compression
NCode(16)
12x compression
NCode(4)
49x compression
NCode(2)
97x compression

CelebA
17 bits/px
JPEG2000
29x compression
JPEG
24x compression
NCode(16)
44x compression
NCode(4)
173x compression
NCode(2)
347x compression

UT Zappos50k
11 bits/px
JPEG2000
16x compression
JPEG
15x compression
NCode(16)
28x compression
NCode(4)
112x compression
NCode(2)
224x compression
one of very few large-scale image datasets to adopt lossless PNG compression. To demonstrate generalizability we further benchmark against CelebA (Liu et al., 2015) and UT Zappos50K (Yu & Grauman, 2014) of larger (64x64) images, though it is important to note that these are modestly pre-compressed using JPEG. For MCode video compression, we select two categories (boxing and hand-waving) from the KTH actions dataset (Schuldt et al., 2004).

The encoder ($f_θ(x)$), decoder ($g_θ(z)$) and discriminator ($d_θ(x)$) functions are all implemented as deep ConvNets (Krizhevsky et al., 2012). The decoder (generator) and discriminator networks adopt the standard DCGAN architecture of multiple convolutional layers with ReLU activation and batch normalization, which was shown to empirically improve training convergence (Radford et al., 2015). The encoder network is identical to the discriminator, except for the output layer which produces a length-$M$ latent vector rather than a scalar in $[0, 1]$. We choose the latent vector length $M = 100$ and sample from $U[-1, 1]$.

Each NCode image dataset is partitioned into separate training and evaluation sets. For MCode video compression, we use the whole duration of 75% and the first half of the remaining 25% of videos for training, and the second half of that 25% for evaluation. We use the Adam optimizer (Kingma & Ba, 2014) with learning rate 0.0002 and momentum 0.5, and weight the pixel and perceptual loss terms with $\lambda_1 = 1$ and $\lambda_2 = 0.002$ respectively.

### 4.1. NCode Image Compression

We use NCode to compress and reconstruct images for each of the aforementioned image datasets and compare performance with JPEG and JPEG2000 (ImageMagick toolbox). Performance is evaluated using PSNR and SSIM metrics, as is standard in the field, averaged over the test images. As these measures are known to correlate quite poorly with human perception of visual quality (Toderici et al., 2016; Ledig et al., 2016), we provide randomly-sampled images under each scheme in Figure 4 to visually validate reconstruction performance. We also leverage the class labels associated with CIFAR-10 to propose an additional evaluation metric, i.e. the classification performance for a ConvNet independently trained on uncompressed examples.

Figure 4 presents our results for CIFAR-10, CelebA and Zappos50k respectively. For each Figure, row (a) presents raw image samples, (b-c) their JPEG/JPEG2000 reconstructions, and (d-f) their NCode reconstructions. Rows (d-f) vary by quantization of the latent vector $z$ (for each of the $M$ elements); 16 bits per element for (d), 4 for (e) and 2 for (f). Across each dataset, it is clear that NCode(16) yields far higher quality reconstructions (in terms of SSIM and visual inspection) than JPEG/2000 at comparable compression levels. This compression ratio can be increased to a full order-of-magnitude greater than JPEG/2000 for NCode(4) while still maintaining recognizable reconstructions. Even for the failure case of over-compression, NCode(2) typically produces images that are plausible with respect to the underlying class semantics. This interpretation of visual quality is further supported by a ConvNet trained to classify uncompressed CIFAR-10 images into its ten constituent categories. Table 1 demonstrates that even at 4-bit quantization of $z$ (~60-fold compression), images are more recognizable than under the ~10-fold compression of the JPEG/2000 schemes.

### 4.2. MCode Video Compression

We apply MCode to compress and reconstruct frames from the aforementioned KTH dataset and compare performance against the MPEG4 (H.264) codec (FFmpeg toolbox). Similar to image compression, performance is evaluated against mean frame-wise PSNR and SSIM metrics (average over test videos) and visualizations provided in Figure 5. Comparing (b) MPEG to (c) frame-by-frame MCode, it is clear that our method provides much higher quality results at a comparable compression level. The relative preservation of background texture and limb sharpness is noteworthy, despite the similar PSNR between MPEG and MCode.

Motivated by MPEG bidirectional prediction, MCode can produce greater compression by interpolating between frames in latent space. This process is shown in Figure 5 (d-f); frames transmitted and reconstructed using standard NCode are are omitted, with remaining $N - 1$ interpolated frames shown for (d) $N = 2$, (e) $N = 4$ and (f) $N = 8$. These temporal correlations can be further leveraged by transmitting only Huffman-encoded difference between $z^{(t)}$ and $z^{(t+N)}$, leading to a further 20%-50% lossless compression on average. As shown in Figure 5, this can lead to order-of-magnitude reduction in bitrate compared to MPEG4 while simultaneously providing more visually plausible sequences.

| NCode | Compression Factor | Accuracy (test set) |
|-------|-------------------|---------------------|
| JPEG2000 | 2.050 | 8.6 | 37.85 |
| JPEG | 2.400 | 7.4 | 31.99 |
| NCode(4) | 0.391 | 45 | 48.39 |
| NCode(2) | 0.195 | 91 | 28.37 |
Figure 5. A comparison of graceful degradation of reconstructed videos under various compression techniques: (a) Hand-waving (Top panel) and boxing (Bottom panel) video sequence randomly sampled from the KTH actions dataset (Schuldt et al., 2004). Row (b) demonstrates the corresponding frame-by-frame reconstructions, bitrates and mean PSNR/SSIM metrics (averaged over the full test set) for MPEG4 (H.264). Row (c) shows the corresponding performance for MCode using $N = 1$, i.e. applying image NCode frame-by-frame. Rows (d-f) demonstrate the extra performance than can be leveraged by linear interpolation between latent vectors $z_t$ and $z_{t+N}$ for (d) $N = 2$, (e) $N = 4$ and (f) $N = 8$ (transmitted frames omitted). Bit rates are presented both before and after Huffman coding (parentheses).

4.3. Robustness to Noisy Channels

The experiments presented above have assumed that $z$ is transferred losslessly, with sender and receiver operating on a single machine. In practice this is not the case, with sender and receiver being physically separated and coupled by a potentially noisy channel. Where wireless signals are involved, bit error rates often occur with a frequency in the order of $\varepsilon = 10^{-3}$. It is also well established that traditional compression algorithms are not robust against these conditions, e.g. bit error rates in the order of just $\varepsilon = 10^{-4}$ result in unacceptable image distortion and a drop in PSNR of more than 7dB (Ho & Kahn, 1997; Santa-Cruz & Ebrahimi, 2000; Weerackody et al., 1996).

The lack of robustness for traditional codecs is largely due to the introduction of variable-length entropy coding schemes, whereby the transmitted signal is essentially a map key with no preservation of semantic similarity between numerically adjacent signals. By contrast, the
5. Discussion

In this paper we have demonstrated a proof-of-concept that adversarially trained neural codecs can yield an order-of-magnitude improvement in image and video compression when compared to traditional compression schemes, and that as compression levels increase, the transmitted data degrades in a graceful manner. Although this is a very promising result, it raises the obvious question of whether these results can generalize to less trivial examples. Handcrafted codecs such as JPEG2000 and MPEG4 are not restricted by image size or class semantics, whereas our NCode and MCode models were only demonstrated for relatively simple 32x32 and 64x64 examples.

Autoencoders can certainly be extended to larger images, as demonstrated by the beautiful reconstructions of (Theis et al., 2017). Instead, the major bottleneck in extending our method to larger images is the well-characterized instability of the adversarial training process (Theis et al., 2015). Compared to regular autoencoders (the encoder/decoder component of NCode), GANs (the decoder/discriminator) are notoriously difficult to train. Thankfully this is also an area receiving substantial attention and many GAN variants have been published that demonstrate improvement in both training dynamics and reconstruction plausibility. Recent examples include (but are not limited to) ALI (Dumoulin et al., 2016), BiGAN (Donahue et al., 2016), InfoGAN (Chen et al., 2016) and the unrolled GAN (Metz et al., 2016). Particularly promising results have recently been reported for WGAN, which uses the Wasserstein distance to reportedly overcome issues of mode collapse (Arjovsky et al., 2017); and EBGAN (Zhao et al., 2016), which is (to our knowledge) the first to generate plausible images at scale 256x256 over 1000 semantic classes. For this paper we instead adopted the simpler yet less effective DCGAN (Radford et al., 2015), though our framework permits the trivial substitution of these improved adversarial modules.

Another important consideration when profiling neural codecs is what exactly should be measured when calculating compression ratios. The premise of the NCode model is that the sender and receiver share an understanding of the underlying semantics of an image or scene, and that the image can be directly encoded with respect to this representation. Less abstractly, one might describe a particular Zappos50k shoe by describing its size, orientation, shape, color and so forth. Presuming the receiver has a prior understanding of shoes in the general sense, the sender can transmit a very detailed description (albeit in terms of more abstract latent variables) while still requiring far less information than transmitting individual pixel intensities. This can be seen clearly in our results. When providing too little information about the source shoe (e.g. Figure 4, bottom row), the receiver may misunderstand the sender’s description (e.g. fourth shoe from the left), resulting in a perceptually plausible shoe that otherwise bares little similarity.

What can we fairly assume that the receiver understands? In the limit, imagine a decoder that is trained to convert a small vector of ones into a single image of a particular shoe. Of course, in this scenario the information content of the image has simply been absorbed into the parameters of the network, and it is unfair to presume that the decoder could know these weights for an arbitrary shoe without the sender transmitting them first. We do believe, however, that there is a sensible middle ground whereby sender and receiver can be assumed to share an understanding of the physical world without requiring it to be explicitly transmitted. Much in the same way that the full JPEG codec is not transmitted alongside a traditionally-compressed image, modern devices could easily maintain a synchronized collection of cached manifolds that capture the low-dimensional embeddings of common objects and scenes (e.g. the 1000 ImageNet categories). This collection could be further augmented by and individual’s usage patterns, e.g. to learn rich manifolds over the appearance of rooms and faces that regularly appear via online video calls. We believe that this capability represents a large step toward replicating the efficiency with which humans can communicate complex experiences and concepts, and would be a valuable component of any generally intelligent system.
References

Arjovsky, Martin, Chintala, Soumith, and Bottou, Léon. Wasserstein GAN. arXiv preprint arXiv:1701.07875, 2017.

Ballé, Johannes, Laparra, Valero, and Simoncelli, Eero P. End-to-end optimized image compression. arXiv preprint arXiv:1611.01704, 2016.

Bengio, Yoshua and Bengio, Samy. Modeling high-dimensional discrete data with multi-layer neural networks. In NIPS, volume 99, pp. 400–406, 1999.

Brock, Andrew, Lim, Theodore, Ritchie, J M, and Weston, Nick. Neural photo editing with introspective adversarial networks. arXiv preprint arXiv:1609.07093, 2016.

Budden, David, Matveev, Alexander, Santurkar, Shibani, Chaudhuri, Shraman Ray, and Shavit, Nir. Deep tensor convolution on multicores. arXiv preprint arXiv:1611.06565, 2016.

Chen, Xi, Duan, Yan, Houthish, Rein, Schulman, John, Sutskever, Ilya, and Abbeel, Pieter. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In Advances in Neural Information Processing Systems, pp. 2172–2180, 2016.

Donahue, Jeff, Krähenbühl, Philipp, and Darrell, Trevor. Adversarial feature learning. arXiv preprint arXiv:1605.09782, 2016.

Dosovitskiy, Alexey and Brox, Thomas. Generating images with perceptual similarity metrics based on deep networks. In Advances in Neural Information Processing Systems, pp. 658–666, 2016.

Dumoulin, Vincent, Belghazi, Ishmael, Poole, Ben, Lamb, Alex, Arjovsky, Martin, Mastropietro, Olivier, and Courville, Aaron. Adversarially learned inference. arXiv preprint arXiv:1606.00704, 2016.

Gatys, Leon, Ecker, Alexander S, and Bethge, Matthias. Texture synthesis using convolutional neural networks. In Advances in Neural Information Processing Systems, pp. 262–270, 2015.

Gatys, Leon A, Ecker, Alexander S, and Bethge, Matthias. Image style transfer using convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2414–2423, 2016.

Gersho, Allen and Gray, Robert M. Vector quantization i: Structure and performance. In Vector quantization and signal compression, pp. 309–343. Springer, 1992.

Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron, and Bengio, Yoshua. Generative adversarial nets. In Advances in neural information processing systems, pp. 2672–2680, 2014.

Graves, Alex and Schmidhuber, Jürgen. Offline handwriting recognition with multidimensional recurrent neural networks. In Advances in neural information processing systems, pp. 545–552, 2009.

Gregor, Karol, Besse, Frederic, Rezende, Danilo Jimenez, Danhelka, Ivo, and Wierstra, Daan. Towards conceptual compression. In Advances In Neural Information Processing Systems, pp. 3549–3557, 2016.

Gupta, Suyog, Agrawal, Ankur, Gopalakrishnan, Kailash, and Narayanan, Pritish. Deep learning with limited numerical precision. In ICML, pp. 1737–1746, 2015.

Ho, Keang-Po and Kahn, Joseph M. Image transmission over noisy channels using multicarrier modulation. Signal Processing: Image Communication, 9(2):159–169, 1997.

Hubara, Itay, Courbariaux, Matthieu, Soudry, Daniel, El-Yaniv, Ran, and Bengio, Yoshua. Quantized neural networks: Training neural networks with low precision weights and activations. arXiv preprint arXiv:1609.07061, 2016.

Huffman, David A. A method for the construction of minimum-redundancy codes. Proceedings of the IRE, 40(9):1098–1101, 1952.

Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

Kingma, Diederik P and Welling, Max. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013.

Kodak. Kodak lossless true color image suite. http://r0k.us/graphics/kodak/, 1999.

Krizhevsky, Alex. Learning multiple layers of features from tiny images. Technical report, 2009.

Krizhevsky, Alex, Sutskever, Ilya, and Hinton, Geoffrey E. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pp. 1097–1105, 2012.

Lamb, Alex, Dumoulin, Vincent, and Courville, Aaron. Discriminative regularization for generative models. arXiv preprint arXiv:1602.03220, 2016.
Laparra, Valero, Ballé, Johannes, Berardino, Alexander, and Simoncelli, Eero P. Perceptual image quality assessment using a normalized laplacian pyramid. *Electronic Imaging*, 2016(16):1–6, 2016.

Larochelle, Hugo and Murray, Iain. The neural autoregressive distribution estimator. In *AISTATS*, volume 1, pp. 2, 2011.

Larsen, Anders Boesen Lindbo, Sønderby, Søren Kaae, Larochelle, Hugo, and Winther, Ole. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

Ledig, Christian, Theis, Lucas, Huszár, Ferenc, Caballero, Jose, Cunningham, Andrew, Acosta, Alejandro, Aitken, Andrew, Tejani, Alykhan, Totz, Johannes, Wang, Zehan, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

Liu, Ziwei, Luo, Ping, Wang, Xiaogang, and Tang, Xiaoou. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

Metz, Luke, Poole, Ben, Pfau, David, and Sohl-Dickstein, Jascha. Unrolled generative adversarial networks. *arXiv preprint arXiv:1611.02163*, 2016.

Radford, Alec, Metz, Luke, and Chintala, Soumith. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

Rastegari, Mohammad, Ordonez, Vicente, Redmon, Joseph, and Farhadi, Ali. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pp. 525–542. Springer, 2016.

Salakhutdinov, Ruslan and Hinton, Geoffrey E. Deep boltzmann machines. In *AISTATS*, volume 1, pp. 3, 2009.

Santa-Cruz, Diego and Ebrahimi, Touradj. An analytical study of jpeg 2000 functionalities. In *Image Processing, 2000. Proceedings. 2000 International Conference on*, volume 2, pp. 49–52. IEEE, 2000.

Schuldt, Christian, Laptev, Ivan, and Caputo, Barbara. Recognizing human actions: A local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pp. 32–36. IEEE, 2004.

Shannon, C. E. A mathematical theory of communication. *Bell system technical journal*, 27, 1948.

Shi, Wenzhe, Caballero, Jose, Huszár, Ferenc, Totz, Johannes, Aitken, Andrew P, Bishop, Rob, Rueckert, Daniel, and Wang, Zehan. 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.

Sønderby, Casper Kaae, Caballero, Jose, Theis, Lucas, Shi, Wenzhe, and Huszár, Ferenc. Amortised MAP Inference for Image Super-resolution. *arXiv preprint arXiv:1610.04490*, 2016.

Theis, L., Shi, W., Cunningham, A., and Huszr, F. Lossy image compression with compressive autoencoders. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/pdf?id=rJkNwv9gg.

Theis, Lucas and Bethge, Matthias. Generative image modeling using spatial LSTMs. In *Advances in Neural Information Processing Systems*, pp. 1927–1935, 2015.

Theis, Lucas, Oord, Aäron van den, and Bethge, Matthias. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.

Toderici, George, Vincent, Damien, Johnston, Nick, Hwang, Sung Jin, Minnen, David, Shor, Joel, and Covell, Michele. Full resolution image compression with recurrent neural networks. *arXiv preprint arXiv:1608.05148*, 2016.

van den Oord, Aaron, Kalchbrenner, Nal, and Kavukcuoglu, Koray. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

Wang, Zhou, Bovik, Alan C, Sheikh, Hamid R, and Simoncelli, Eero P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Weerackody, Vijitha, Podilchuk, Christine, and Estrella, Anthony. Transmission of jpeg-coded images over wireless channels. *Bell Labs Technical Journal*, 1(2):111–126, 1996.

Yu, A. and Grauman, K. Fine-Grained Visual Comparisons with Local Learning. In *Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Zhao, Junbo, Mathieu, Michael, and LeCun, Yann. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

Zhu, Jun-Yan, Krähenbühl, Philipp, Shechtman, Eli, and Efros, Alexei A. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, pp. 597–613. Springer, 2016.