

MIT Open Access Articles

Generative Compression

The MIT Faculty has made this article openly available. **Please share** how this access benefits you. Your story matters.

Citation: S. Santurkar, D. Budden and N. Shavit, "Generative Compression," 2018 Picture Coding Symposium (PCS), San Francisco, CA, 2018, pp. 258-262, doi: 10.1109/PCS.2018.8456298.

Published Version: 10.1109/PCS.2018.8456298

Publisher: Institute of Electrical and Electronics Engineers (IEEE)

Permanent Link: <https://hdl.handle.net/1721.1/125883>

Version: Original manuscript: author's manuscript prior to formal peer review

Terms of use: <http://creativecommons.org/licenses/by-nc-sa/4.0/>



Generative Compression

Shibani Santurkar David Budden[†] Nir Shavit
 Massachusetts Institute of Technology
 {shibani,budden,shanir}@mit.edu

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 suggest that it is a direction worth pursuing 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 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. Traditional compression techniques, such as JPEG, are agnostic to the data being compressed and do not degrade gracefully. This is shown in Figure 1, which compares (a) two original images to (b) their JPEG2000-compressed representations. Building upon the ideas of [1] and the recent promise of deep generative models [2], 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.

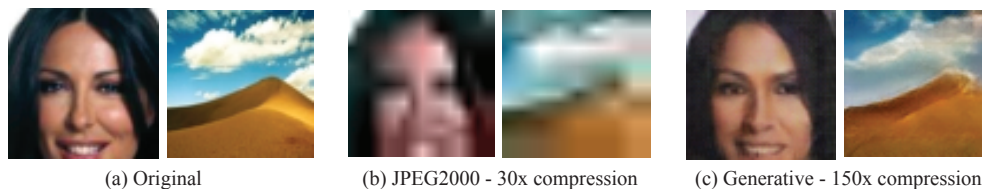


Figure 1: Traditional versus generative image compression.

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. 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. To see why, 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. Likewise, 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? The promise of generative compression is to translate this perceptual redundancy into a reduction in code verbosity.

[†] This author is now at Google DeepMind.

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. In this form, compression seeks to minimize both the rate of the latent code, lower-bounded by the entropy of its distribution, and the distortion of the output, typically reported as a signal-to-noise-ratio (PSNR) or structural similarity (SSIM) metric.

Joint optimization over rate and distortion has long been considered an intractable problem for images and other high-dimensional spaces [3]. 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 (JPEG2000). 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.

2 Generative Models for Image Compression

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, $\mathbf{z} = f_\theta(\mathbf{x})$, and the synthesis transform with a learnt decoder function, $\hat{\mathbf{x}} = g_\phi(\mathbf{z})$. Noteworthy examples include the compressive autoencoder [4], which derives differentiable approximations for quantization and entropy rate estimation to allow end-to-end training by gradient backpropagation. The authors of [5] achieve a similar result, using a joint nonlinearity as a form of gain control. Also noteworthy is the LSTM-based autoencoder framework presented in [6], specifically designed for the common failure case of compressing small thumbnail images. This approach was later extended using fully-convolutional networks for the compression of full-resolution images [7]. Collectively, these and similar models are showing promising results in both lossless [8, 9] and lossy data compression [1, 7].

Recent advancements in generative modelling also show promise for compression. Imagine that the role of the receiver is simply to synthesize some realistic looking MNIST sample. If we knew the true distribution, $P(\mathbf{x})$, of this class of images defined over X , we could simply sample $\hat{\mathbf{x}} \in \mathbb{R}^N$ from this distribution. Unfortunately, it is intractable to accurately estimate this density function for such a high-dimensional space. One remedy to this problem is to factorize $P(\mathbf{x})$ as the product of conditional distributions over pixels. This sequence modeling problem can be solved effectively using autoregressive models of recurrent neural networks, allowing the generation of high-quality images or in-filling of partial occlusions [8]. 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, $\hat{\mathbf{x}} = g_\phi(\mathbf{z})$, to approximate $P(\mathbf{x})$ as the transformation of some prior latent distribution, $P(\mathbf{z})$. To generate realistic-looking samples, we wish to train g to minimize the difference between its distribution, $P_\phi(\mathbf{x}) = \mathbb{E}_{\mathbf{z} \sim P(\mathbf{z})} [P_\phi(\mathbf{x}|\mathbf{z})]$, and the unknown true distribution, $P(\mathbf{x})$. A popular solution to this problem is to introduce an auxiliary discriminator network, $d_\vartheta(\mathbf{x})$, which learns to map \mathbf{x} to the probability that it was sampled from $P(\mathbf{x})$ instead of $P_\phi(\mathbf{x})$ [2]. This framework of generative adversarial networks (GANs) simultaneously learns ϕ and ϑ by training against the minimax objective:

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

The authors showed that this objective is equivalent to minimizing the Jensen-Shannon divergence between $P_\phi(\mathbf{x})$ and $P(\mathbf{x})$ 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 [10]. Similar to GANs, VAEs introduce an auxiliary network to facilitate training. Unlike GANs, this inference function is trained to learn an approximation, $Q_\theta(\mathbf{z}|\mathbf{x})$, of the true posterior, $P(\mathbf{z}|\mathbf{x})$, 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 a variational lower bound, $\mathcal{L}(\theta, \phi; \mathbf{x})$. Recent studies have demonstrated the potential of VAEs for compression by training $P_\theta(\mathbf{x}|\mathbf{z})$ and $Q_\phi(\mathbf{z}|\mathbf{x})$ as deep neural

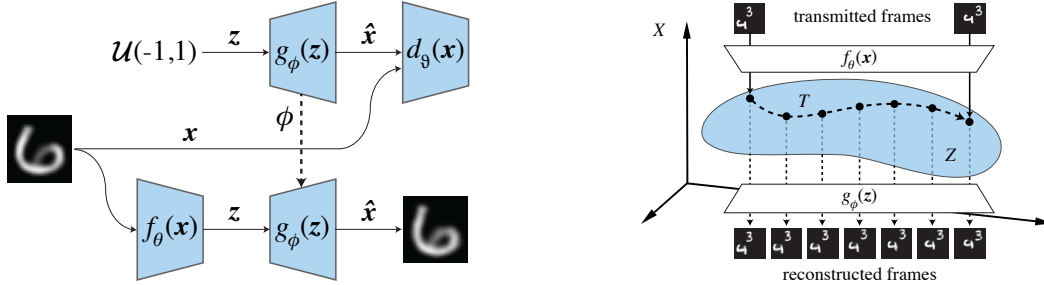


Figure 2: Generative compression architectures for (left) image and (right) video data.

networks. The authors of [1] report that they do not build an actual compression algorithm, but present sample reconstructions with perceptual quality similar to JPEG2000. However, a well-established limitation of VAEs (and autoencoders more generally) is that maximizing a Gaussian likelihood is equivalent to minimizing the L^2 loss between pixel intensity vectors. This loss is known to correlate poorly with human perception and leads to blurry reconstructions [9, 11].

3 Neural Codecs for Generative Compression

To build an effective neural codec for image compression, we implement the paired encoder/decoder interface of a VAE while generating the higher-quality images expected of a GAN. We propose a simple neural codec architecture (NCode, Figure 2a) that approaches this in two stages. First, a decoder network, $g_\phi : Z \rightarrow X$, is greedily pre-trained using an adversarial loss with respect to the auxiliary discriminator network, $d_\theta : X \rightarrow [0, 1]$. For this stage, g_ϕ and d_θ are implemented using DCGAN-style ConvNets [12]. Second, an encoder network, $f_\theta : X \rightarrow Z$, is trained to minimize some distortion loss, $L(\mathbf{x}, g_\phi(f_\theta(\mathbf{x})))$, with respect to this non-adaptive decoder. We also investigate methods for lossy quantization of \mathbf{z} , motivated by recent studies demonstrating the robustness of deep neural nets to reduced numerical precision [13, 14]. Compression is improved by either (a) reducing the length of the latent vector, and/or (b) reducing the number of bits used to encode each entry.

Traditional image compression algorithms have been crafted to minimize pixel-level loss metrics. Although optimizing for MSE can lead to good PSNR characteristics, the resulting images are perceptually implausible due to a depletion of high-frequency components (blurriness) [15]. By adversarially pre-training a non-adaptive decoder, the codec will tend to produce samples more like those that fool a 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 [16]. Perceptual loss metrics derived from these features have been used to improve the plausibility of generative image models [17] and successfully applied to applications including super-resolution [15] and style transfer [18]. We take a similar approach in NCode, modeling the distortion between image, \mathbf{x} , and reconstruction, $\hat{\mathbf{x}} = g_\phi(f_\theta(\mathbf{x}))$, as the weighted sum of pixel-level and perceptual losses:

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

where conv_4 is the fourth convolutional layer of an ImageNet-pretrained AlexNet [19, 20].

Our NCode model architecture was largely motivated by the work of [20], where a similar framework was applied for on-manifold photo editing. It also bears similarity to many other existing models. For example, recent studies have proposed hybrid models that combine VAEs with GANs [11, 17, 21]. Our model differs in the adoption of a non-adaptive and adversarially pre-trained decoder, our hybrid perceptual/pixel loss function, and the use of a vanilla autoencoder architecture (in place of a VAE). Similarly, other studies have augmented GANs with inference functions, e.g. adversarial feature learning [22] and adversarially learned inference (ALI) [23]. 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 and plausible samples, they differ dramatically 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.

3.1 Generative Video Compression

Here we present what is, to our knowledge, the first example of neural network-based compression of video data at sub-MPEG rates. As a video is simply a sequence of images, these images can be compressed and transmitted frame-by-frame using NCode. This is reminiscent of the motion-JPEG scheme in traditional video compression literature. However, this approach fails to capture the rich temporal correlations in natural video data [24, 25]. Instead, we would prefer our model to be inspired by the interpolation (bidirectional prediction) scheme introduced for the popular MPEG standard.

The simplest method of capturing temporal redundancy is to transmit only every N -th frame, $\mathbf{X} = [\mathbf{x}^{(t)}, \mathbf{x}^{(t+N)}, \mathbf{x}^{(t+2N)}, \dots]$, requiring the receiver to interpolate the missing data with a small N -frame latency. The traditional limitation of this approach is that interpolation in pixel-space yields visually displeasing results. Instead, we choose to model a video sequence as uniformly-spaced samples along a path, T , on the manifold, Z (Figure 2). We assume that Z is a lower-dimensional embedding of some latent image class, and further that for sufficiently small N , the path $\mathbf{x}^{(t)} \rightarrow \mathbf{x}^{(t+N)}$ can be approximated by linear interpolation on Z . This assumption builds on the wealth of recent literature demonstrating that interpolating on manifolds learnt by generative models produce perceptually cohesive samples, even between quite dissimilar endpoints [12, 20, 26].

Similar to MPEG, we can further compress a video sequence through delta and entropy coding schemes (specifically, Huffman coding). Each latent vector is transmitted as its difference with respect to the previous transmitted frame, $\delta^{(t+N)} = \mathbf{z}^{(t+N)} - \mathbf{z}^{(t)}$. We observe that this representation gains far more from entropy coding than for individual latent vectors sampled from $P(\mathbf{z})$, leading to a further (lossless) reduction in bitrate. We do not use entropy coding for NCode image compression.

4 Experiments

We selected the CelebA [27], UT Zappos50K [28] and MIT Places (outdoor natural scenes) [29] datasets for compression benchmarking. These datasets have the necessary data volume required for training, and are small enough (64×64) to use with current GAN training (see Section 5). Traditional compression benchmarks, such as the Kodak PhotoCD dataset [30], currently fail on both criteria. Moreover, patch-based approaches to piecewise compression are unable to capture the image-level semantics that allow an image to be efficiently represented in terms of a low-dimensional latent vector from a generative model. We further evaluate our model on the popular CIFAR-10 dataset [31]. Beyond data volume, the advantages of CIFAR are (a) that each example has a ground-truth class label (useful for validation), and (b) that it is one of very few large-scale image datasets to adopt lossless PNG compression. For MCode video compression, we select two categories (hand-waving and boxing) from the KTH actions dataset [32].

The encoder ($f_\theta(\mathbf{x})$), decoder ($g_\phi(\mathbf{z})$) and discriminator ($d_\psi(\mathbf{x})$) functions are all implemented as deep ConvNets [19]. 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 [12]. 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 vary the latent vector length $M = \{25, 50, 100\}$ and sample from the uniform prior, $\mathcal{U}[-1, 1]$. Substituting this for a truncated normal distribution had no notable impact.

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 [33] 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

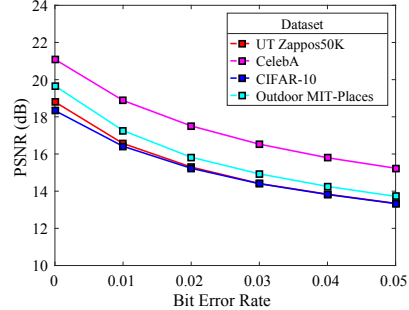
We use NCode to compress and reconstruct images for each dataset and compare performance with JPEG/2000 (ImageMagick toolbox). As these schemes are not designed specifically for small images, we also compare to the state-of-the-art system for thumbnail compression presented by Toderici et al. [6, 7]. Performance is evaluated using the standard PSNR and SSIM metrics, averaged over the held-aside test set images. As these measures are known to correlate quite poorly with human

(a)	CelebA 17 bits/px		<i>bits/px</i> PSNR SSIM
(b)	JPEG2000 41x compression		<i>0.406</i> 20.49 0.588
(c)	JPEG 28x compression		<i>0.611</i> 20.33 0.590
(d)	Toderici et al. 136x compression		<i>0.125</i> 21.45 0.673
(e)	NCode(100, 5) 140x compression		<i>0.121</i> 20.72 0.740
(f)	NCode(25, 4) 708x compression		<i>0.024</i> 18.79 0.679
(g)	NCode(25, 2) 1416x compression		<i>0.012</i> 14.97 0.545
(a)	UT Zappos50k 11 bits/px		<i>bits/px</i> PSNR SSIM
(b)	JPEG2000 21x compression		<i>0.520</i> 19.63 0.705
(c)	JPEG 17x compression		<i>0.642</i> 19.90 0.707
(d)	Toderici et al. 88x compression		<i>0.125</i> 18.73 0.703
(e)	NCode(100, 5) 90x compression		<i>0.121</i> 18.25 0.720
(f)	NCode(25, 4) 458x compression		<i>0.024</i> 17.23 0.694
(g)	NCode(25, 2) 917x compression		<i>0.012</i> 12.34 0.542
(a)	MIT Places 16 bits/px		<i>bits/px</i> PSNR SSIM
(b)	JPEG2000 30x compression		<i>0.532</i> 20.28 0.498
(c)	JPEG 29x compression		<i>0.556</i> 18.37 0.457
(d)	Toderici et al. 128x compression		<i>0.125</i> 21.33 0.592
(e)	NCode(100, 5) 132x compression		<i>0.121</i> 19.32 0.574
(f)	NCode(25, 4) 667x compression		<i>0.024</i> 18.38 0.558
(g)	NCode(25, 2) 1333x compression		<i>0.012</i> 11.83 0.366

Figure 3: A comparison of image reconstruction under various compression techniques. (a) Randomly sampled images from the CelebA dataset (*top panel*) [27], Zappos50k dataset (*middle panel*) [28] and Outdoor MIT Places dataset (*bottom panel*) [29]. Rows (b-d) demonstrate the corresponding reconstructions, compression ratios and PSNR/SSIM metrics (averaged over the full test set) for JPEG2000, JPEG and the thumbnail compression approach of Toderici et al. [6, 7] respectively. Corresponding NCode performance is shown for varying latent vector dimension and quantization levels: (e) (100, 5 bit), (f) (25, 4 bit), and (g) (25, 2 bit) latent representations of the latent vector. File header size was excluded from JPEG/2000 results for fairer comparison. Best viewed zoomed in.

CIFAR-10	bits/pixel	η	accuracy
Original	19	–	70.53
JPEG2000	1.354	14	37.48
JPEG	1.962	10	32.61
Toderici et al.	0.125	152	34.15
NCode(100, 5)	0.4883	39	54.96
NCode(25, 4)	0.0977	194	39.46
NCode(25, 2)	0.0488	389	18.95

(a)



(b)

Figure 4: (a) Validation of perceptual plausibility of images compressed (η =compression factor) with NCode versus headerless JPEG, headerless JPEG2000 and Toderici et al [6], using a ConvNet independently trained on uncompressed images to categorize each sample into its respective CIFAR-10 [31] class. (b) Reduction in reconstruction quality (PSNR) as a function of bit error rate, ε , for each NCode image dataset. JPEG PSNR is known to degrade by more than 7dB at $\varepsilon \sim 10^{-4}$.

perception of visual quality [7, 15], we provide randomly-sampled images under each scheme in Figure 3 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. As file headers are responsible for a non-trivial portion of file size for small images, these were deducted when calculating compression for JPEG/JPEG2000. Huffman coding and quantization tables were however included for JPEG.

Our results are presented in Figure 3. For each panel, row (a) presents raw image samples, (b-d) their JPEG2000, JPEG and Toderici et al. [6, 7] reconstructions, and (e-g) their reconstructions using our proposed NCode method. To illustrate how NCode sample quality distorts with diminishing file size, we present sample reconstructions at varying latent vector length and quantization levels. These specific values were chosen to demonstrate (e) improved visual quality at similar compression levels, and (f-g) graceful degradation at extreme compression levels. It is clear that NCode(100,5) (length-100 latent representation, at 5 bits per vector entry) yields higher quality reconstructions (in terms of SSIM and visual inspection) than JPEG/2000 at ~ 4 -fold higher compression levels. This compression ratio can be increased to a full order-of-magnitude greater than JPEG/2000 for NCode(25, 4) while still maintaining recognizable reconstructions. Even for the failure case of over-compression, NCode(25, 2) typically produces images that are plausible with respect to the underlying class semantics. NCode(100, 5) samples appear to be visually sharper and with fewer unnatural artifacts compared to the Toderici et al. approach set to maximum allowable compression.

Our appraisal of improved perceptual quality is supported by training a ConvNet to classify uncompressed CIFAR-10 images into their ten constituent categories, and observing how its accuracy drops when presented with images compressed under each scheme. Figure 4a demonstrates that even using NCode(25, 4) (~ 190 -fold compression), images are more recognizable than under the ~ 150 -fold compression of the Toderici et al. approach or ~ 15 -fold compression of JPEG/2000.

4.2 Robustness to Noisy Channels

The experiments presented above have assumed that \mathbf{z} is transferred losslessly, with sender and receiver operating on a single machine. Where wireless signals are involved and in the absence of explicit error correction, 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 [34, 35, 36]. 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 NCode system transmits explicit coordinates in the latent space Z and thus should be robust against bit errors in \mathbf{z} , as shown in Figure 4b. Even at bit error rates of $\varepsilon = 10^{-2}$, which is greater than one should experience in practice, PSNR degrades by just ~ 1 dB.

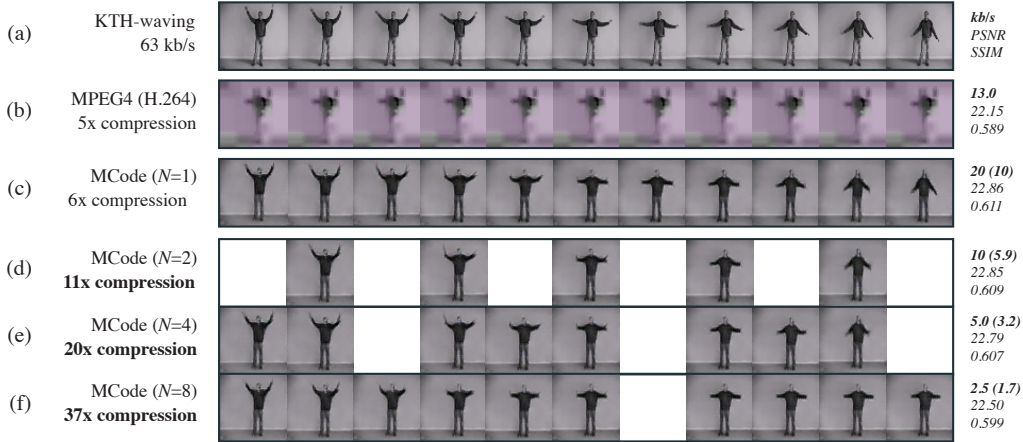


Figure 5: (a) Hand-waving video sequence randomly sampled from the KTH actions dataset [32]. Row (b) demonstrates the corresponding frame-by-frame reconstructions, bitrates and mean PSNR/SSIM metrics (averaged over the full test set) for MPEG4 (H.264). Row (c) shows the corresponding performance for MCode using $N = 1$, i.e. applying image NCode frame-by-frame. Rows (d-f) demonstrate the extra performance that can be leveraged by linear interpolation between latent vectors $\mathbf{z}^{(t)}$ and $\mathbf{z}^{(t+N)}$ 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). Best viewed zoomed in.

4.3 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 for the handwaving dataset. Results from the boxing dataset are similar and included in the Supplementary Material. Comparing (b) MPEG to (c) frame-by-frame MCode, it is clear that our method provides higher quality results at a comparable compression level. Despite similar PSNR, the relative preservation of background texture and limb sharpness is noteworthy.

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 omitted, with the 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 the Huffman-encoded difference between $\mathbf{z}^{(t)}$ and $\mathbf{z}^{(t+N)}$, 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 over MPEG4 while providing more visually plausible sequences.

The robustness analysis in Section 4.2 extends to video MCode in the absence of inter-frame entropy coding. Figure 5 presented MCode compression factors both with and without Huffman coding, which introduced 20%-50% further performance improvement on average. We present both options for the user to choose based on the robustness versus compression constraints of their application.

5 Large Image Compression

So far we have only demonstrated generative compression for relatively simple 64x64 examples. Although our results are promising, this raises the obvious question of whether these results can generalize to larger images with more complex class semantics, e.g. those represented by ImageNet. In fact, we believe that the compression factors presented here should only continue to improve for larger images. The latent vector \mathbf{z} describes semantic content of an image, which should grow sub-linearly (or likely remain constant) for higher-resolution images of equivalent content.

Current limitations in this direction are not fundamental to generative compression, but are instead those of generative modelling more generally. Although autoencoders can certainly be extended to larger images, as demonstrated by the beautiful reconstructions of [4], GANs fail for large images

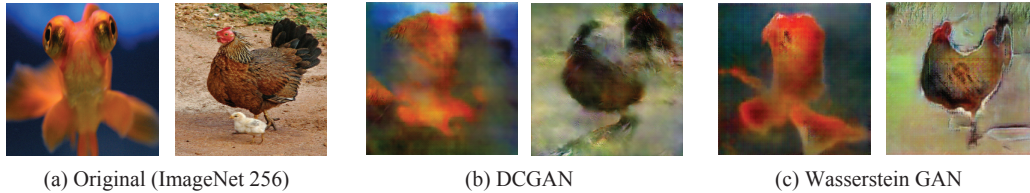


Figure 6: The impact of one year of GAN advancements – (b) DCGAN [12] to (c) Wasserstein GAN [38] – on generative compression performance (256x256 ImageNet). Best viewed zoomed in.

due to well-established instabilities in the adversarial training process [37]. Thankfully this is also an area receiving substantial attention, and in Figure 6 we demonstrate the impact of the past year of GAN research on 256x256 image compression (held-aside test samples from a 10-class animal subset of ImageNet). Unlike vanilla DCGAN (used throughout this paper), the more recent Wasserstein GAN [38] produces recognizable reconstructions. Likewise, this and other improved generative models could be leveraged to improve the compression factors and reconstruction quality demonstrated throughout this paper. We expect this field will continue to improve at a rapid pace.

6 Discussion

All compression algorithms involve a pair of analysis and synthesis transforms that aim to accurately reproduce the original images. Traditional, hand-crafted codecs lack adaptability and are unable to leverage semantic redundancy in natural images. Moreover, earlier neural network-based approaches optimize against a pixel-level objective that tends to produce blurry reconstructions. In this paper we propose generative compression as an alternative, where we first train the synthesis transform as a generative model. We adopted a simple DCGAN for this purpose, though any suitable generative model conditioned on a latent representation could be applied. This synthesis transform is then used as a non-adaptive decoder in an autoencoder setup, thereby confining the search space of reconstructions to a smaller compact set of natural images enriched for the appropriate class semantics.

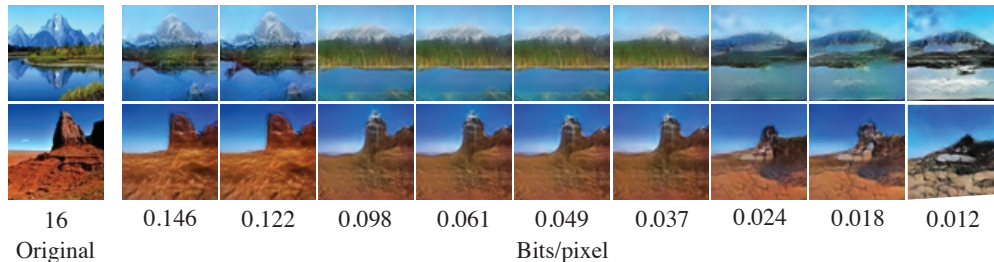


Figure 7: Graceful degradation of generative compression at 2-to-3 orders-of-magnitude compression.

We have demonstrated the potential of generative compression for orders-of-magnitude improvement in image and video compression – both in terms of compression factor and noise tolerance – when compared to traditional schemes. Generatively compressed images also degrade gracefully at deeper compression levels, as shown in Figure 7. These results are possible because the transmitted data is merely a description with respect to the receiver’s shared understanding of natural image semantics.

To implement generative compression as a practical compression standard, we envision that devices could maintain a synchronized collection of cached manifolds that capture the low-dimensional embeddings of common objects and scenes (e.g. the 1000 ImageNet categories). In much the same way that the JPEG codec is not transmitted alongside every JPEG-compressed image, this removes the need to explicitly transmit network weights for previously encountered concepts. 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 such a system would better replicate the efficiency with which humans can store and communicate memories of complex scenes, and would be a valuable component of any generally intelligent system.

Acknowledgments

Support is gratefully acknowledged from the National Science Foundation (NSF) under grants IIS-1447786 and CCF-1563880, and the Intelligence Advanced Research Projects Activity (IARPA) under grant 138076-5093555.

References

- [1] Karol Gregor, Frederic Besse, Danilo Jimenez Rezende, Ivo Danihelka, and Daan Wierstra. Towards conceptual compression. In *Advances In Neural Information Processing Systems*, pages 3549–3557, 2016.
- [2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.
- [3] Allen Gersho and Robert M Gray. Vector quantization i: Structure and performance. In *Vector quantization and signal compression*, pages 309–343. Springer, 1992.
- [4] L. Theis, W. Shi, A. Cunningham, and F. Huszar. Lossy image compression with compressive autoencoders. In *International Conference on Learning Representations*, 2017.
- [5] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.
- [6] 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.
- [7] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. *arXiv preprint arXiv:1608.05148*, 2016.
- [8] Aaron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.
- [9] Lucas Theis and Matthias Bethge. Generative image modeling using spatial LSTMs. In *Advances in Neural Information Processing Systems*, pages 1927–1935, 2015.
- [10] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.
- [11] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.
- [12] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.
- [13] Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *ICML*, pages 1737–1746, 2015.
- [14] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *arXiv preprint arXiv:1609.07061*, 2016.
- [15] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.
- [16] Leon Gatys, Alexander S Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 262–270, 2015.
- [17] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pages 658–666, 2016.

- [18] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.
- [19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [20] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision*, pages 597–613. Springer, 2016.
- [21] Alex Lamb, Vincent Dumoulin, and Aaron Courville. Discriminative regularization for generative models. *arXiv preprint arXiv:1602.03220*, 2016.
- [22] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.
- [23] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martin Arjovsky, Olivier Mastropietro, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.
- [24] David Budden, Alexander Matveev, Shibani Santurkar, Shraman Ray Chaudhuri, and Nir Shavit. Deep tensor convolution on multicores. *arXiv preprint arXiv:1611.06565*, 2016.
- [25] Wenzhe Shi, Jose Caballero, Ferenc Huszar, 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*, pages 1874–1883, 2016.
- [26] Andrew Brock, Theodore Lim, JM Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016.
- [27] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.
- [28] A. Yu and K. Grauman. Fine-Grained Visual Comparisons with Local Learning. In *Computer Vision and Pattern Recognition (CVPR)*, June 2014.
- [29] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.
- [30] Kodak. Kodak lossless true color image suite. <http://r0k.us/graphics/kodak/>, 1999.
- [31] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.
- [32] Christian Schuldt, Ivan Laptev, and Barbara Caputo. Recognizing human actions: A local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004.
- [33] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [34] Keang-Po Ho and Joseph M Kahn. Image transmission over noisy channels using multicarrier modulation. *Signal Processing: Image Communication*, 9(2):159–169, 1997.
- [35] Diego Santa-Cruz and Touradj Ebrahimi. An analytical study of jpeg 2000 functionalities. In *Image Processing, 2000. Proceedings. 2000 International Conference on*, volume 2, pages 49–52. IEEE, 2000.
- [36] Vijitha Weerackody, Christine Podilchuk, and Anthony Estrella. Transmission of jpeg-coded images over wireless channels. *Bell Labs Technical Journal*, 1(2):111–126, 1996.
- [37] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2015.
- [38] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017.