GANs for Lossy Image Compression

Niklas Smedemark-Margulies *
smedemark-margulie.n@northeastern.edu

Jung Yeon Park *
park.jungy@northeastern.edu

Nicholas Vann *
vann.n@husky.neu.edu

Abstract

We apply generative adversarial networks (GANs) to lossy image compression, and explore how reconstruction error varies with latent dimension. Our experiments suggest that the trained linear layer helps bias the model to the training domain, and may increase representation error. By adjusting the initial linear layer of a trained model, we are able to smoothly vary the compression ratio without retraining, and achieve high quality reconstructions. This technique works for wildly out-of-domain images, and also achieves nearly state-of-the-art test reconstruction quality for compressive sensing.

1 Context

1.1 Generative models for inverse problems

Generative adversarial networks (GANs) have been used for a variety of inverse problems in imaging [10]. In these applications, we solve an optimization problem to approximately reconstruct an image from its measurements.

Reconstruction error in inverse problems comes from three sources: **measurement noise** (which is not present in compression), **optimization error** from failing to converge to a global optimum, and **representation error** when our global optimum is still far from the true image.

Recent literature suggests that representation error may be rate-limiting when using GANs for real-world problems such as compressive sensing [2] or phase retrieval [4].

1.2 Lossy Compression

We investigate lossy image compression within inverse problems. A compression system consists of an encoder E and a decoder D, and should satisfy $D(E(x)) \approx x$ and $size(E(x)) \ll size(x)$ for an input image x. Rate-distortion theory describes the trade-off between these two properties [8].

1.3 Main Contributions

We detail exactly how to compress/decompress images with a GAN and introduce a simple method to vary the compression ratio without retraining by replacing the initial linear layer at test time. We call this approach GANZ. Our method achieves high quality reconstructions at modest compression ratios, even for out-of-domain images. We experiment further with compressive sensing, and GANZ achieves nearly state-of-the-art reconstruction there as well. Source code is available at .

2 Question

In this paper, we seek to answer the following questions:

*Khoury College of Computer Science, Northeastern University, Boston, MA, 02115

1. Can we construct a useful image compression method using GANs?
2. How does changing the latent dimension of the GAN affect representation error and reconstruction quality change?
3. Can we increase the latent dimension of the model at test time without retraining the entire model?

3 Setup

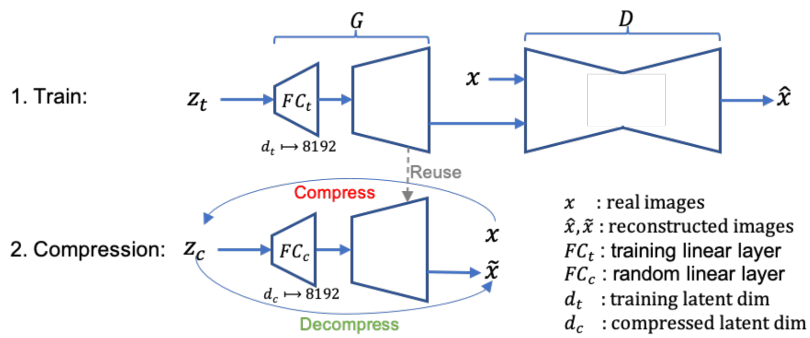


Figure 1: GANZ Model Architecture

3.1 GANZ training

We use Boundary Equilibrium GAN (BEGAN)[1] as the base architecture and vary the latent dimension at train (d_t) and compression time (d_c) to examine its effect on reconstruction error (see Figure 1). We choose BEGAN for its stable training and high-quality output (basis implementation provided by [3]).

3.2 Compression

For compression of an image x, we reuse the generator G, and optimize the latent vector z_c with SGD following (1). Decompression is simply $G(z_c^*)$. During both compression/decompression, the linear layer and generator weights are fixed.

$$z_c^* = \min_{z_c} \|G(z_c) - x\|^2 \tag{1}$$

Controllable compression ratio: linear layer adjustment We uncover a simple trick to vary the compression ratio without retraining (Fig. 1). For compression ratio R, we replace the first layer with a fixed-seed random linear layer with input size $d_c = (128 \cdot 128 \cdot 3)/R$, which allows for deterministic decompression.

3.3 Experimental Details

We train on CelebA [6], cropped to (128x128) pixels. We measure compression ratios in degrees of freedom and total file size (KiB). Note that our file size includes the latent vector with 32-bit entries and generator metadata. We measure reconstruction quality using Peak Signal-to-Noise Ratio (PSNR) (see 2). We experiment with varying the training latent dimension, linear layer adjustment. We also experiment briefly with compressive sensing with random measurement matrices (Fig. 8).

3.4 Wavelets compression baseline

We implement a naive wavelet baseline using PyWavelets [5] (Daubechies 4), based on [7]. Specifically, for compressed latent dimension M and input dimension N, we zero everything but the largest $M/(3N)$ coefficients per color channel.

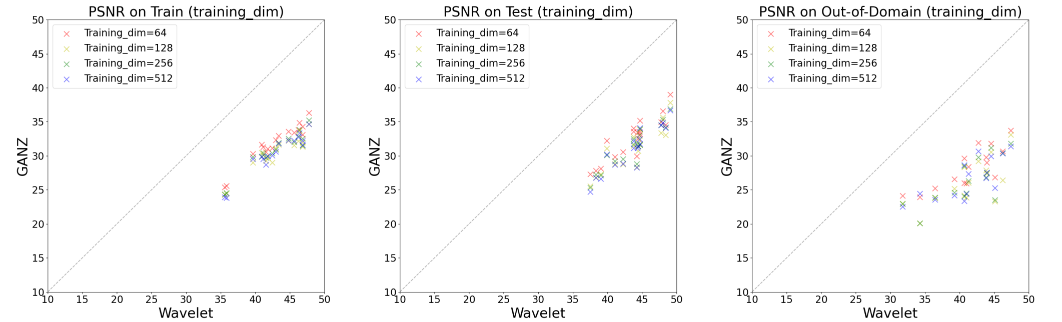


Figure 2: PSNR values of GANZ vs wavelets. Displaying varying training latent dimensions of GANZ with a fixed compression ratio of 10.

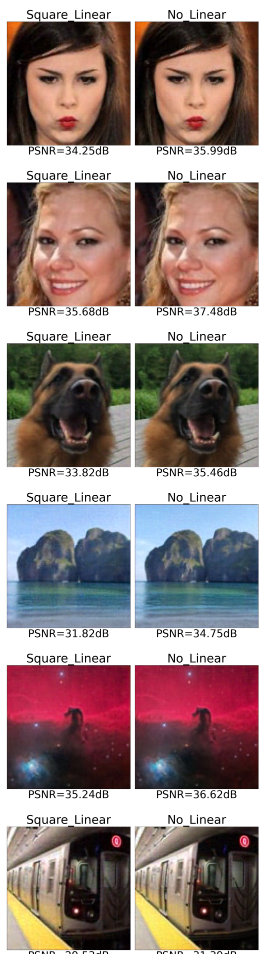
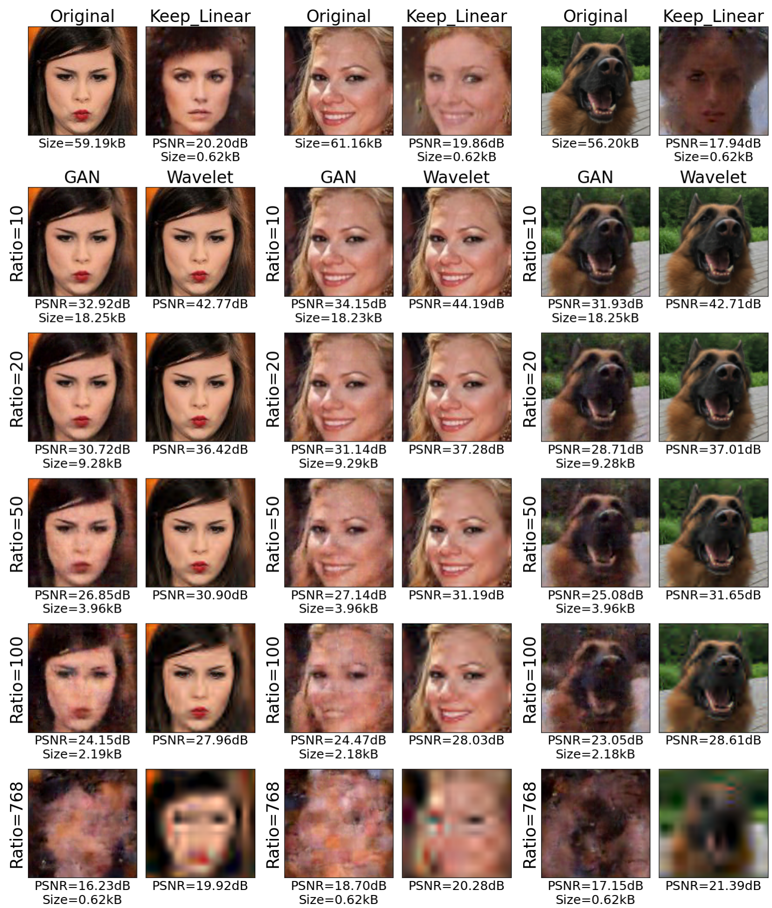


Figure 3: Square linear layer vs no linear layer at compression ratio 6



(a) Train (b) Test (c) Out-of-domain

Figure 4: Compression Ratios: GANZ vs Wavelets

4 Results

Effect of training latent dimension We first vary the training latent dimension of GANZ, and compare reconstruction PSNR against wavelets on train, test, and out-of-domain images (see Fig. 2). Wavelet compression outperforms GANZ for all cases. We find both the GANZ PSNR and quality of generator samples decreases with increasing training latent dimension. We therefore use training a low latent dimension of 64 for successive experiments.

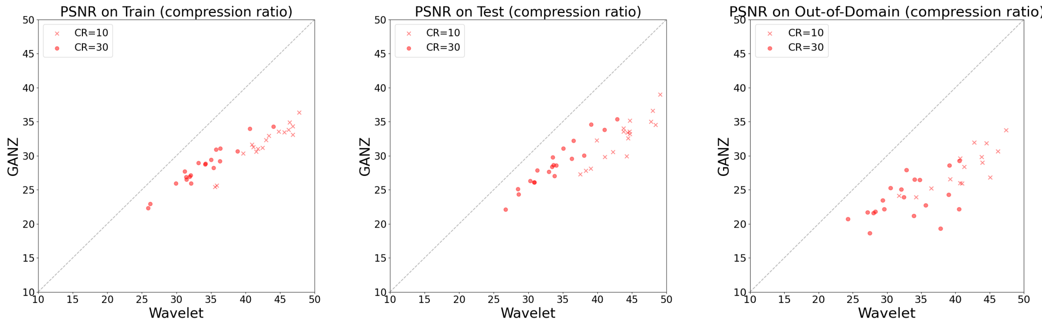


Figure 5: PSNR values of GANZ vs wavelets. Charts comparing different compression ratios with a training latent dimension of 64.

Effect of adjusting linear layer In Fig. 3, "no_linear" represents feeding latent vectors of dimension 8192 (compression ratio 6) directly to the generator's convolutional module, while "square_linear" adds a random square linear layer. As the latent dimensions are equal, we presume that these two models have equal representation error. Both methods generalize well even for out-of-domain images (last 4 rows of Fig. 3), indicating that the model's bias towards "natural" images may be contained in its convolutional module. Using a random linear layer incurs only a small additional optimization error, suggesting that random linear layers of varying dimension may work.

In Fig. 4, we show the original, the result of using the unmodified trained generator ("keep_linear"), and the result from using a random linear layer with compression ratios between 10 and 768 (compression ratio 768 equivalent to "keep_linear") alongside wavelet results. The wavelet baseline still outperforms GANZ at all compression ratios, but the difference in PSNRs decreases with higher compression ratios. Of note, we unexpectedly find that the quality at "Ratio=768" is extremely poor compared to "keep_linear" which seems to indicate our optimization procedure is a key source of error. Fig. 5 echos these results for 20 train and test images from CelebA, as well as 20 out-of-domain images (included in our source code).

Compressive Sensing We use GANZ for compressive sensing, and compare with results from Bora et al.[2] (see their Figure 1b and Figure 6b) at the same measurement ratios $M/N \in \{0.08, 0.2\}$. The best mean squared error reported by [2] at least 0.025 (equivalent to PSNR of 16.021) for $M/N \in \{0.08, 0.2\}$. We achieve much higher PSNR values ($23 \sim 32$dB) for a small selection of images (See Appendix: Fig. 8).

5 Discussion

Current results In this preliminary investigation, we find that GANZ can achieve high quality reconstructions for compression and compressive sensing. This, even on diverse out-of-domain images, using a model trained only on CelebA. Our results may suggest that the convolutional module biases the model towards natural images and the trained linear layer further biases the model to a subset of that domain (human faces), increasing representation error at test time. By using random linear layer adjustments, we can control the compression ratio of GANZ without retraining. However, it is not yet on par with naive approaches like wavelets.

Future Work Future work includes two main directions: improving optimization during training and during compression. Using batch normalization, different activation functions, or other diversity parameter values could improve GANZ's overall performance. More datasets could further improve the model. To reduce optimization error and improve quality at extreme compression ratios, we could also adopt an iterative "surfing" strategy [9] and/or use additional compression techniques such as quantization and run-length encoding. Different initialization schemes could be used for the random linear layers and latent vectors. Jointly optimizing the linear layer and latent vector might help dissect the role of optimization error and representation error in GANZ.

References

[1] David Berthelot, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[2] Ashish Bora, Ajil Jalal, Eric Price, and Alexandros G Dimakis. Compressed sensing using generative models. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 537–546. JMLR. org, 2017.

[3] Max Daniels, Paul Hand, and Reinhard Heckel. Reducing the representation error of gan image priors using the deep decoder. *arXiv preprint arXiv:2001.08747*, 2020.

[4] Paul Hand, Oscar Leong, and Vlad Voroninski. Phase retrieval under a generative prior. In *Advances in Neural Information Processing Systems*, pages 9136–9146, 2018.

[5] Gregory Lee, Ralf Gommers, Filip Waselewski, Kai Wohlfahrt, and Aaron O'Leary. Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, 2019.

[6] 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)*, December 2015.

[7] Miki Lustig, Frank Ong, and Jon Tamir. *Compressed Sensing Tutorial*, 2016 (accessed April 11, 2020).

[8] David JC MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[9] Ganlin Song, Zhou Fan, and John Lafferty. Surfing: Iterative optimization over incrementally trained deep networks, 2019.

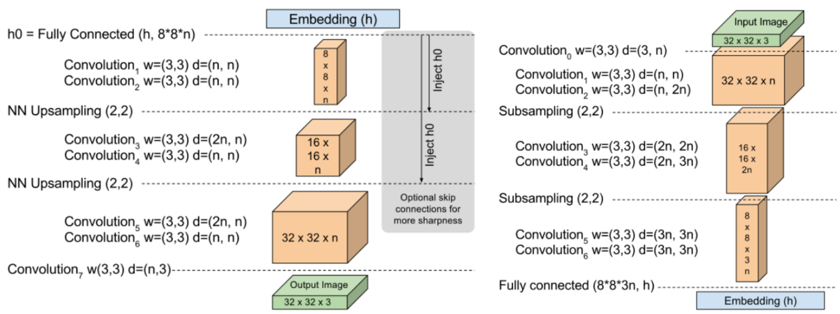
[10] Qiaojing Yan and Wei Wang. Dcgans for image super-resolution, denoising and debluring. *Advances in Neural Information Processing Systems*, pages 487–495, 2017.

6 Appendix

6.1 Acknowledgements

We thank Max Daniels for a reference implementation of BEGAN, and for help debugging our optimization process.

6.2 Setup Details



(a) Generator/Decoder (b) Encoder

Figure 6: BEGAN Model Architecture. The encoder and decoder are used together to form the discriminator. Note the generator linear layer "h0", which is key for our experiments.

For a true image $x \in \mathbb{R}^N$ and a reconstructed image $\hat{x}$, with a max pixel intensity MAX_I,

$$\text{PSNR}(x, \hat{x}) = 20 \cdot \log_{10}(\text{MAX}_I) - 10 \cdot \log_{10}\left(\frac{\|x - \hat{x}\|^2}{N}\right) \tag{2}$$

6.3 Additional Results

Figure 7: A 2 by 3 grid of generated images for each latent dimension in {64,128,256,512} clockwise from upper left

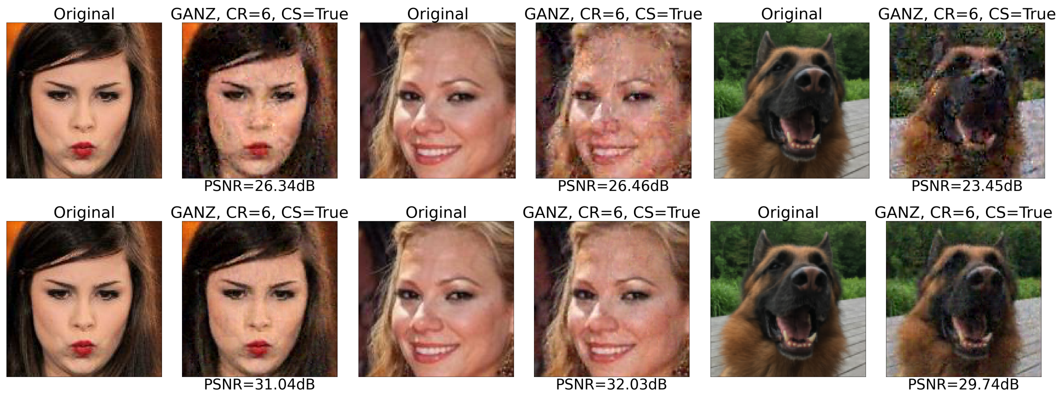


Figure 8: Compressive Sensing with $M = 4000$ (top) or $M = 10000$ (bottom) Gaussian measurements (Image dimension $N = 128 \cdot 128 \cdot 3$), for train (left), test (middle), and out-of-domain (right). We use GANZ model with compression ratio 6 for all images.