LEARNING TO GENERATE IMAGES WITH PERCEPTUAL SIMILARITY METRICS

Karl Ridgeway*, **Jake Snell**†, **Brett D. Roads***, **Richard S. Zemel**†, **Michael C. Mozer***
*Department of Computer Science, University of Colorado, Boulder
†Department of Computer Science, University of Toronto

ABSTRACT

Deep networks are increasingly being applied to problems involving image synthesis, e.g., generating images from textual descriptions, or generating reconstructions of an input image in an autoencoder architecture. Supervised training of image-synthesis networks typically uses a pixel-wise squared error (SE) loss to indicate the mismatch between a generated image and its corresponding target image. We propose to instead use a loss function that is better calibrated to human perceptual judgments of image quality: the structural-similarity (SSIM) score of Wang, Bovik, Sheikh, and Simoncelli (2004). Because the SSIM score is differentiable, it is easily incorporated into gradient-descent learning. We compare the consequences of using SSIM versus SE loss on representations formed in deep autoencoder and recurrent neural network architectures. SSIM-optimized representations yield a superior basis for image classification compared to SE-optimized representations. Further, human observers prefer images generated by the SSIM-optimized networks by nearly a 7:1 ratio. Just as computer vision has advanced through the use of convolutional architectures that mimic the structure of the mammalian visual system, we argue that significant additional advances can be made in modeling images through the use of training objectives that are well aligned to characteristics of human perception.

1 INTRODUCTION

Recently, interest in developing methods for training neural networks to synthesize images has exploded. The reason for this surge is threefold. First, the problem of image generation spans a wide range of difficulty, from synthetic images to handwritten digits to naturally cluttered and high-dimensional scenes, the latter of which provides a fertile development and testing ground for generative models. Second, learning good generative models of images involves learning new representations. Such representations are believed to be useful for a variety of machine learning tasks, such as classification or clustering, and can also transfer between tasks. Third, image generation is fun and captures popular imagination, as efforts such as Google's Inceptionism machine demonstrate.

One of the primary methods for learning generative models of images is the autoencoder architecture. Autoencoders are made up of two functions, an encoder and a decoder. The encoder compresses an image into a feature vector, typically of low dimension, and the decoder takes that vector as input and reconstructs the original image as output. The autoencoder is trained to reproduce an image that is similar to the input, where similarity is typically measured in terms of the Euclidean distance between the image and its reconstruction. In a probabilistic autoencoder, where the output is viewed as a distribution over images, the model is trained to maximize the log-likelihood of the original image under this distribution.

Autoencoders use a *full-reference metric* to compare an original image and its reconstruction. Such a metric is based on the complete pixel-based representation of the image. The simplest full-reference metric is mean square error (MSE), which is computed by averaging the square of the pixel intensity differences for every pixel in an image. However, MSE is known to be a poor representation of human judgments of quality. For example, a distorted image created by decreasing the contrast can yield the same MSE as one created by increasing the contrast, but the two distortions can yield quite different human judgments of visual quality; and distorting an image with salt-and-pepper impulse noise obtains a small MSE but is judged as having low visual quality relative to the original image.

In this paper, we explore the effects of incorporating a loss function that, unlike MSE, is grounded in human perceptual judgements. We show that this perceptually-optimized loss leads to generated images that are judged to be of higher quality. We also show that representations learned via this perceptually-optimized loss are better suited for image classification.

2 Related Work

2.1 Models for Generating Images

The primary class of image-generating neural networks are autoencoders. There are two main types of autoencoders. The first set are deterministic, which directly map the input through hidden layers and output a reconstruction of the original image. Typically, MSE is used to evaluate the reconstruction. The second type are probabilistic models. With these models the key issue concerns the intractability of inference in the latent variables, e.g., Helmholtz Machines (Dayan et al., 1995) and variational autoencoders (VAE) (Kingma & Welling, 2013). The encoder is used to approximate a posterior distribution and the decoder is used to stochastically reconstruct the data from latent variables. Gregor et al. (2015) further introduced the Deep Recurrent Attention Writer (DRAW), extending the VAE approach by incorporating a novel differentiable attention mechanism. In all of these methods, the model output is treated as a distribution, and the evaluation of this output is the log-likelihood of the original image.

A second class of generative models are variants of Boltzmann Machines (Smolensky, 1986; Hinton & Sejnowski, 1986) and Deep Belief Networks (Hinton et al., 2006) While these models are very powerful, each iteration of training requires a computationally costly step of MCMC to approximate derivatives of an intractable partition function (normalization constant), making it difficult to scale them to large datasets.

A more recent approach to learning generative image models involves directly training a generator, which maps samples drawn from a uniform distribution through a deep neural network that outputs images, and trains to make the set of images generated by the model indistinguishable from real images. Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) is a paradigm that involves training a discriminator that attempts to distinguish real from generated images, along with a generator that attempts to trick the discriminator. Recently, Denton et al. (2015) have scaled this approach by training conditional GANs at each level of a Laplacian pyramid of images. An alternative approach, moment-matching networks (Li et al., 2015), directly trains the generator to make the statistics of these two distributions match. Neither of these approaches, moment-matching or adversarial, directly train the network to reconstruct each training image, so they do not utilize any error measure on an image and its reconstruction.

Autoencoders and deep belief nets have one advantage over models that directly generate images, such as GANs: they *interpret* images in addition to generating images. For example, Krizhevsky & Hinton (2011) used deep autoencoders to discover compact codes that were better for classifying images than using the raw image data. In this paper, we use a similar autoencoder/image-classification paradigm as a tool to compare and evaluate models.

2.2 Perception-Based Error Metrics

As digitization of photos and videos became commonplace in the 1990s, the need for digital compression became apparent. Lossy compression schemes distorted image data, and it was important to quantify the drop in quality resulting from compression in order to optimize the compression scheme. Because compressed digital artifacts are eventually used by humans, researchers attempted to develop full-reference image quality metrics that take into account features to which the human visual system is sensitive and that ignore features to which it is insensitive. Some of these metrics are built on complex models of the human visual system, such as the Sarnoff JND model (Lubin, 1998), the visual differences predictor (Daly, 1992), the moving picture quality metric (Van den Branden Lambrecht & Verscheure, 1996), and the perceptual distortion metric (Winkler, 1998).

Other metrics take more of an engineering approach, and are based on the extraction and analysis of specific features of an image to which human perception is sensitive. The most popular of these metrics is the structural similarity metric (SSIM) (Wang et al., 2004), which aims to match

the luminance, contrast, and structure information in an image. Other such metrics are the visual information fidelity metric (Sheikh & Bovik, 2006), which is an information theory-based measure, and the visual signal-to-noise ratio (Chandler & Hemami, 2007).

Finally, there are transform-based methods, which compare the images after some transformation has been applied. Some of these methods include DCT/wavelets, discrete orthonormal transforms, and singular value decomposition.

2.2.1 STRUCTURAL SIMILARITY

In this paper, we train autoencoders with the structural-similarity (SSIM) metric and compare to autoencoders trained with MSE. We chose the SSIM metric for our initial investigation because it is well accepted and frequently utilized in the literature. Further, its pixelwise gradient has a simple analytical form and is inexpensive to compute. In this work, we focus on the original grayscale SSIM, although there are interesting variations and improvements including color SSIM (Hassan & Bhagvati, 2012), and multiscale SSIM (Wang et al., 2003).

The original SSIM metric, as described in Wang et al. (2004), is a pixelwise measure that compares corresponding pixels in two images, denoted x and y, with three comparison functions—luminance (I), contrast (C), and structure (S)—defined in Equation 1:

$$I(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \qquad C(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \qquad S(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \tag{1}$$

The variables μ_x, μ_y, σ_x, and σ_y denote mean pixel intensity and the standard deviations of pixel intensity in a local image patch centered at either x or y. Following Wang et al. (2004), we chose a square neighborhood of 5 pixels on either side of x or y, resulting in 11×11 patches. The variable σ_{xy} denotes the sample correlation coefficient between corresponding pixels in the patches centered at x and y. The constants C_1, C_2, and C_3 are small values added for numerical stability. The three comparison functions are combined to form the SSIM score:

$$\text{SSIM}(x,y) = I(x,y)^\alpha \cdot C(x,y)^\beta \cdot S(x,y)^\gamma \tag{2}$$

In our work, we weight each function equally ($\alpha = \beta = \gamma = 1$) and set $C_1 = C_2$ to end up with the formula for SSIM:

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{3}$$

Our objective is to minimize the loss related to the sum of structural-similarity scores across all image pixels,

$$\mathcal{L}(X,Y) = -\sum_i \text{SSIM}(X_i, Y_i),$$

where X and Y are the original and reconstructed images, and i is an index over image pixels. Equation 2.2.1 has a simple analytical derivative, as found in Wang & Simoncelli (2008), and therefore it is trivial to perform gradient descent in the SSIM-related loss.

3 METHODOLOGY

3.1 NETWORK ARCHITECTURES

Our primary experiments are based on two autoencoder architectures: a fully-connected network and a convolutional network. Each architecture was constructed with a bottleneck layer—the middle layer of the deep autoencoder; we used 256 nodes for the fully-connected net and either 256 or 512 nodes for the convolutional net. Each architecture was trained either with MSE or SSIM-related loss. We refer to a specific model by its architecture, the size of the bottleneck layer, and the loss function it was trained with. The fully-connected models are referred to as FC-256-{SSIM,MSE}, and the convolutional models are referred to as Conv-{256,512}-{SSIM,MSE}. Through the use of multiple model variants, we aim to demonstrate that our results are robust to architectural details.

3.1.1 FULLY CONNECTED ARCHITECTURE

We adopted the fully-connected autoencoder architecture of Krizhevsky & Hinton (2011), detailed in the left panel of Table 1. Every layer has rectified linear activation functions, except the bottleneck and output layers, which have tanh activations. Krizhevsky & Hinton (2011) trained their network by stacking restricted Boltzmann machines and then fine tuning with back propagation. Instead, we train our network from scratch using back propagation and stochastic gradient descent.

3.1.2 CONVOLUTIONAL ARCHITECTURE

Because of the popularity and power of convolutional nets for image processing, we included a convolutional autoencoder in our experiments. The right panel of Table 1 shows the structure of our model which uses convolutional layers to encode the input and then deconvolutional layers to decode the feature representation in the bottleneck layer. The deconvolutional layers are implemented as convolutional layers that are preceded by an upsampling step that creates a layer with 2 times the dimensions of the input layer by repeating the values of the input. To explore the role of the capacity of the convolutional layer, we built models with both 256 and 512 node bottlenecks. The specifics of convolutional autoencoders are described in Masci et al. (2011).

outputs	**activation**
1024	(Input)
8192	ReLU
4096	ReLU
2048	ReLU
1024	ReLU
512	ReLU
256	binary-tanh
512	ReLU
1024	ReLU
2048	ReLU
4096	ReLU
8192	ReLU
1024	tanh

size out	**kernel**	**stride**	**activation**	**type**
32×32×1				Input
15×15×32	4×4×1, 32	2	tanh	convolution
6×6×64	4×4×32, 64	2	tanh	convolution
256/512			binary-tanh	FC
6×6×64			tanh	FC
15×15×32	4×4×64, 32	2	tanh	deconvolution
32×32×1	4×4×32, 1	2	tanh	deconvolution

Table 1: Left: Details of the fully-connected (FC-256) architecture. Right: details of the convolutional architectures (Conv-{256,512}). Convolutional and deconvolutional kernels are described in the format (width×height×channels, filters).

3.1.3 ACTIVATION QUANTIZATION

In order to enforce a strong compression of the signal in our autoencoders, we force the activations of nodes in the bottleneck layer to be binary (-1 or $+1$). Following Krizhevsky & Hinton (2011), we threshold the activations in the forward pass and use the original continuous value for the purpose of gradient calculation during back propagation. We perform this quanitization both during training and testing and all results reported are based on the quantized bottleneck-layer representations. Krizhevsky & Hinton (2011) quantized in order to obtain binary codes for hash table indexing. Our interest is to enforce a more categorical representation. We also experimented with models in which the bottleneck layer was not quantized, and we found that the quantized representations were better at predicting image classification (e.g., dog versus cat versus airplane).

3.2 DATA SETS AND TRAINING METHODOLOGY

We train autoencoders using a subset of approximately two million images of the 80 million tiny-images data set (Torralba et al., 2008), consisting of the first 30 images for every English proper noun. The images in this dataset have dimensions 32×32 pixels and consist of three RGB color channels. We mapped the three color channels to a single grayscale channel using the python `pillow` library's `convert` function. The input pixels are rescaled to the range $[-1, 1]$, to match the tanh activation function on all of our output layers.

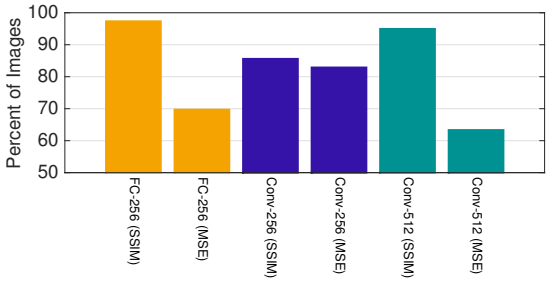

Figure 1: For each architecture and optimization metric, this plot shows the percent of test images from the network trained to optimize that metric that score better than the other network; e.g., the rightmost bar shows that the Conv-512 network trained to optimize MSE beat the same network trained to optimize SSIM on 63% of the test images, when MSE was the comparison metric.

All testing and evaluation of our models used the CIFAR-10 data set, which consists of 60,000 color images, each drawn from one of ten categories. We chose a diverse data set for training in order ensure that the autoencoders were learning general statistical characteristics of images, and not peculiarities of the CIFAR-10 data set. The CIFAR-10 color images were converted to a single grayscale channel, as was done for the training data set. We divided the CIFAR-10 images into a *search database* (48,000 images) and a *query list* (12,000 images). The purpose of these two subsets will be explained in our results section. One use of search database was as a validation set to determine when to stop training: training terminated when the reconstruction error—as measured by the appropriate training metric, either MSE or SSIM—stopped improving following one complete pass through the training set.

We train using mini-batches of size 64. The SSIM and MSE metrics are scaled differently, so we performed empirical explorations to set the learning rate appropriately for each. For MSE, we use a learning rate of 5×10^{-5}, and for SSIM 5×10^{-2}. All architectures were trained with a momentum of 0.9 and with weight decay of 5×10^{-5}.

4 RESULTS

As expected, the MSE-optimized nets tend to achieve better MSE reconstruction scores, and the SSIM-optimized nets tend to achieve better SSIM reconstruction scores. Figure 1 shows the relative performance of each network on the metric it is supposed to optimize. Each bar indicates the proportion of images for which an architecture trained to optimize performance metric X obtains better performance than an architecture trained to optimize the other performance metric, when evaluated on metric X. The fact that all bars are above 50% indicates that training on one metric or the other has a significant influence on the resulting models.

To further compare models trained with the SSIM and MSE metric, we utilize both subjective and objective characteristics of the model output. Subjective characteristics are determined by asking humans to judge image reconstruction quality. Objective characteristics are determined by examining categorical clustering of the bottleneck-layer representations.

4.1 JUDGMENTS OF IMAGE RECONSTRUCTION QUALITY

Do human observers prefer reconstructions produced by the SSIM-optimized networks or by the MSE-optimized networks? We collected judgments of perceptual quality on Amazon Mechanical Turk. Participants were presented with a sequence of image triplets with the original (reference) image in the center and the SSIM- and MSE-optimized reconstructions on either side. Participants were instructed to select which of the two reconstructions they preferred. Half the time the SSIM-optimized reconstruction appeared on the left and half the time it appeared on the right. All reconstructions came from the FC-256-{SSIM,MSE} networks.

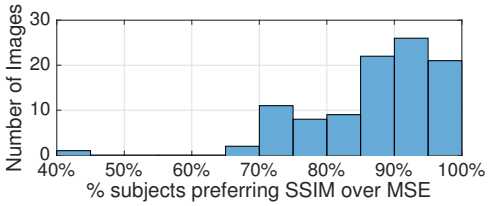

Figure 2: Distribution of participant SSIM preference proportion for 100 image triplets. For example, the graph shows that for 9 of the test triplets, 16 of the 20 participants preferred SSIM.

In a first study, twenty participants provided preference judgments on the same set of 100 randomly selected images from the CIFAR-10 data set. For each image triple, we recorded the proportion of participants who choose the SSIM reconstruction of the image over the MSE reconstruction. Figure 2 shows the distribution of inter-participant preference for SSIM reconstructions across all 100 images. If participants were choosing randomly, we would expect to see roughly 50% preference for most images. However, a plurality of images have over 90% inter-participant agreement on SSIM, and almost no images have MSE reconstructions that are preferred over SSIM reconstructions by a majority of participants.

Figure 3a shows the sixteen image triplets for which the largest proportion of participants preferred the SSIM reconstruction. The original image is shown in the center of the triplet and the MSE- and SSIM-optimized reconstructions appear on the left and right, respectively. (In the actual experiment, the two reconstructions were flipped on half of the trials.) In this Figure, the SSIM reconstructions all show important object details that are lost in the MSE reconstructions.

Figure 3b shows the sixteen image triples for which the smallest proportion of participants preferred the SSIM reconstruction. In the first 15 of these images, still a majority (55-80%) of participants preferred the SSIM reconstruction to the MSE reconstruction; only in the image in the lower right corner did a majority prefer the MSE reconstruction (60%). In this Figure, the SSIM-optimized reconstructions still seem to show as much detail as the MSE-optimized reconstructions, and the inconsistency in the ratings may indicate that the two reconstructions are of about equal quality.

In a second study on Mechanical Turk, twenty new participants each provided preference judgments on a randomly drawn set of 100 images and their reconstructions. The images were different for each participant; consequently, a total of 2000 images were judged. Participants preferred the SSIM- over MSE-optimized reconstructions by nearly a 7:1 ratio: the SSIM reconstruction was chosen for 86.25% of the images. Examining individual participants, The participant choosing SSIM reconstructions the least still preferred them 63% of the time, and the participant choosing SSIM reconstructions the most preferred them 99% of the time.

4.2 Evaluation of Learned Representations

In the previous section, we showed that using a perceptually-aligned training objective improves the quality of image synthesis, as judged by human observers. In this section, we go further and claim that the SSIM objective leads to the discovery of internal representations in the neural net that are more closely tied to the category associated with an image.

We examine the compressed representations of the image in the bottleneck layer, which we'll refer to as the image *code*. As explained in the Methodology section above, these codes are binary vectors of either 256 or 512 elements. If the SSIM objective biases learning toward the discovery of codes that convey good information about the object present in an image, then we should see categorical clustering of codes. That is, the code associated with the image of one dog should be more similar to codes for images of other dogs than perhaps the code for an image of a visually similar cat. Using the method of Krizhevsky & Hinton (2011), we probe the network with a set of *query* images and we use the corresponding code to index into a *search database*—a set of 48,000 images whose codes and category labels have been stored. Using the search database to identify the k nearest neighbors

(a)  (b) 

Figure 3: Examples of both MSE and SSIM image reconstructions. Image triples are sorted, from top to bottom and left to right, by the percentage of participants that preferred SSIM. (a) Sixteen images for which participants strongly preferred the SSIM reconstruction over the MSE reconstruction. Participants preferred the SSIM reconstruction of the top 10 unanimously, and only 1 participant (of 20) preferred MSE on the last six. (b) Sixteen images for which the smallest proportion of participants preferred the SSIM reconstruction. In the first 15 of these images, only 20-45% of participants preferred the MSE reconstruction, and the last image 60%.

in Hamming distance, we can compute the proportion of the nearest k that have the same category label as the query image. Codes that embody category information will yield a higher score.

Figure 4 shows the k-NN classification results for each of the six models on the CIFAR-10 test set. The abscissa specifies $k \in \{1, ..., 10\}$ and the ordinate indicates the mean proportion of the k nearest neighbors that are of the same class as the query. For each of the three architectures, the SSIM-optimized model obtains better classification performance than the MSE-optimized model. The difference is larger for the two convolutional architectures than for the fully connected architecture. We speculate that convolutional nets might benefit more because the SSIM measure itself is also based on a convolution operator, and convolutional networks are able to more efficiently represent the type of information that SSIM tries to preserve.

To more directly link SSIM with codes that embody object-category information, Figure 5 shows mean MSE and SSIM reconstruction scores for the six architectures (top and middle rows), along with the proportion correct classification for $k = 10$ (bottom row). Note that the classification performance is correlated with the SSIM score but not the MSE score.

4.3 Qualitative Experiments on Recurrent Image Generation

In order to further explore the role of perceptual losses in learning models for image generation, we adapt the DRAW model of Gregor et al. (2015) to be trained with an arbitrary differentiable image similarity metric. The DRAW model generates an image by sampling a latent $z_t \sim P(Z_t)$ for each of a fixed number of timesteps. Each z_t is passed as input to a decoder RNN. The output of the decoder RNN is used to update c_t, the model's accumulated representation of the output image (also known as the *canvas*). Due to the intractable posterior over z_t given an image x, the decoder RNN is simultaneously trained with an encoder RNN that produces a variational approximation $Q(Z_t|x)$ to the true posterior. During training, the expected sum of the KL-divergence of $P(Z_t)$ from $Q(Z_t|x)$ and the negative log probability of x under the model is minimized:

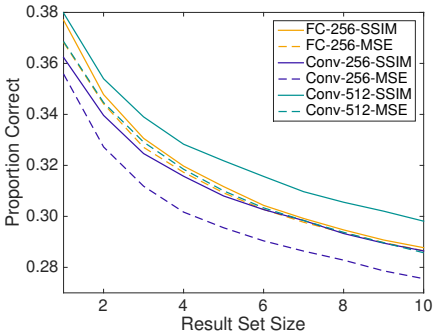

Figure 4: k-NN classification accuracy for the test set, for various k along the abscissa. The ordinate specifies the proportion of the top-k neighbors that share the same class as the probe image.

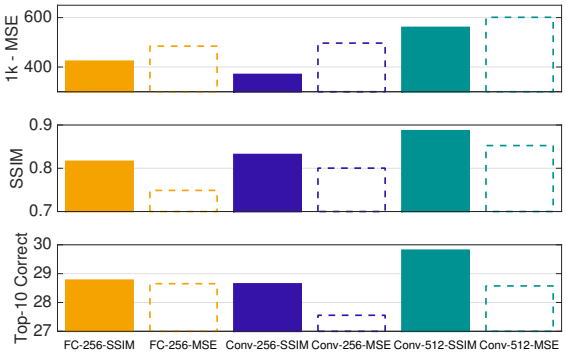

Figure 5: Comparison of the networks on MSE and SSIM metrics and top-10 classification performance. The MSE scores are reversed so that they have the same polarity as the SSIM and classification scores.

$$\mathcal{L}^{DRAW} = \mathbb{E}_{z\sim Q}\left[\sum_{t=1}^{T} KL(Q(Z_t|x)||P(Z_t)) - \log D(x|c_T)\right] \tag{4}$$

where $D(X|c_T)$ is the model of the input data given the canvas at the final timestep. We modify this learning objective by replacing $-\log D(x|c_T)$ with an arbitrary loss between images $\Delta(x, y)$ and weighting the resulting sum. We call this modification *Expected-Loss DRAW* (EL-DRAW). Its objective takes the following form:

$$\mathcal{L}^{EL-DRAW} = \mathbb{E}_{z\sim Q}\left[\sum_{t=1}^{T} KL(Q(Z_t|x)||P(Z_t)) + C \cdot \Delta(x, \hat{x})\right] \tag{5}$$

where C is a constant governing the trade-off between the latent loss and image-specific loss, and $\hat{x} = f(c^T|z)$ is the model's deterministic prediction of the image given the final state of the canvas.

We trained EL-DRAW on grayscale 32 x 32 images of dogs from the CIFAR-10 dataset. We experimented with both MSE and negative SSIM as our image-specific loss Δ and trained models with a range of C for each. Models trained with different loss functions will in general require separately chosen settings of C due to differences in scaling. We chose a setting for each loss that represented a comparable tradeoff between reconstruction error and KL-divergence from the prior. Details regarding the dataset and experimental hyperparameters are provided in Section 6.1 of the supplementary material, and our method for selecting C values that makes the comparison fair is described in Section 6.2.

Test reconstructions produced by both the MSE-optimized and SSIM-optimized EL-DRAW models are shown in Figure 6. Samples from both models are visualized in Figure 7. The reconstructions and samples from the SSIM-optimized EL-DRAW model are noticeably sharper than those of the MSE-optimized model. The superiority of the former samples is due to the use of a perceptually-grounded loss, which is better suited to capturing—and generating—salient details in images.

5 Discussion and Future Work

We have investigated the consequences of replacing the standard MSE loss function with a perceptually-grounded loss function, SSIM, in neural networks that generate images. Human observers judge SSIM-optimized images to be of higher quality than MSE-optimized images. Beyond this subjective measure, we also showed that the compressed representations of SSIM-optimized

Figure 6: EL-DRAW test results. Each triple consists of the input image (center), and the MSE- and SSIM-optimized reconstructions (left and right, respectively).

Figure 7: Samples from MSE- and SSIM-optimized EL-DRAW models (left and right panels, respectively). Samples from the SSIM-optimized model are sharper and contain more easily recognizable features such as noses and ears.

autoencoders preserve more information about object categories as compared to those of MSE-optimized autoencoders. These key results hold for both fully-connected and convolutional architectures and for various bottleneck sizes. In addition, we have shown that recurrent neural network architectures also benefit from training with SSIM in that they produce qualitatively better reconstructions and generated samples compared to those obtained from training with MSE.

With respect to the experiments on recurrent image generation, we plan to go beyond qualitative assessment of the EL-DRAW model, both by collecting human judgments of reconstruction quality, and by evaluating held-out data likelihood. The latter could be accomplished for example by Parzen window estimation, or by Hybrid Monte Carlo (HMC) methods as done by Kingma & Welling (2013). We are also investigating how probabilistic generative models can be formulated by taking into account perceptual loss.

Given our encouraging results, it seems appropriate to investigate other perceptually-grounded loss functions. SSIM is the low-hanging fruit because it is differentiable. Nonetheless, even black-box loss functions can be cached into a *forward model* neural net (Jordan & Rumelhart, 1992) that maps image pairs into a quality measure. We can then back propagate through the forward model to transform a loss derivative expressed in perceptual quality into a loss derivative expressed in terms of individual output node activities. This flexible framework will allow us to combine multiple perceptually-grounded loss functions. Further, we can refine any perceptually-grounded loss functions with additional data obtained from human preference judgments, such as those we collected in the present set of experiments.

REFERENCES

Chandler, D. M. and Hemami, S. S. Vsnr: A wavelet-based visual signal-to-noise ratio for natural images. *IEEE Transactions on Image Processing*, 16(9):2284–2298, 2007.

Daly, S. J. Visible differences predictor: an algorithm for the assessment of image fidelity. In *SPIE/IS&T 1992 Symposium on Electronic Imaging: Science and Technology*, pp. 2–15. International Society for Optics and Photonics, 1992.

Dayan, P., Hinton, G. E., Neal, R. M., and Zemel, R. S. The Helmholtz machine. *Neural Computation*, 7(5):889–904, 1995.

Denton, E., Chintala, S., Szlam, A., and Fergus, R. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. *arXiv 1506.05751 [stat.ML]*, pp. 1–10, 2015.

Goodfellow, I. J., Pouget-Abadie, J., and Mirza, M. Generative adversarial networks. *arXiv 1406.266v1 [stat.ML]*, pp. 1–9, 2014.

Gregor, K., Danihelka, I., Graves, A., and Wierstra, D. DRAW: A recurrent neural network for image generation. *arXiv preprint arXiv:1502.04623*, 2015.

Hassan, Mohammed and Bhagvati, Chakravarthy. Structural Similarity Measure for Color Images. *International Journal of Computer Applications (0975 8887)*, 43(14):7–12, 2012.

Hinton, G. E. and Sejnowski, T. J. Learning and relearning in boltzmann machines. In Rumelhart, D. E. and McClelland, J. L. (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*, volume 1, pp. 283–317. MIT Press, Cambridge, Mass., 1986, 1986. ISBN 0262181207.

Hinton, G. E., Osindero, S., and Teh, Y. W. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–54, 2006.

Jordan, M. I. and Rumelhart, D. E. Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16(3):307–354, 1992.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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

Krizhevsky, Alex and Hinton, Geoffrey E. Using very deep autoencoders for content-based image retrieval. In *ESANN*. Citeseer, 2011.

Li, Yujia, Swersky, Kevin, and Zemel, Rich. Generative Moment Matching Networks. In *Proceedings of The 32nd International Conference on Machine Learning*, pp. 1718–1727, 2015.

Lubin, Jeffrey. A human vision system model for objective image fidelity and target detectability measurements. In *Proc. EUSIPCO*, volume 98, pp. 1069–1072, 1998.

Masci, Jonathan, Meier, Ueli, Cireşan, Dan, and Schmidhuber, Jürgen. Stacked convolutional auto-encoders for hierarchical feature extraction. In *Artificial Neural Networks and Machine Learning–ICANN 2011*, pp. 52–59. Springer, 2011.

Sheikh, Hamid Rahim and Bovik, Alan C. Image information and visual quality. *IEEE Transactions on Image Processing*, 15(2):430–444, 2006.

Smolensky, P. Information processing in dynamical systems: Foundations of harmony theory. In Rumelhart, D. E. and McClelland, J. L. (eds.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1*, pp. 194–281. 1986.

Torralba, Antonio, Fergus, Rob, and Freeman, William T. 80 million tiny images: A large data set for nonparametric object and scene recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(11):1958–1970, 2008.

Van den Branden Lambrecht, C. J. and Verscheure, O. Perceptual quality measure using a spatiotemporal model of the human visual system. In *Electronic Imaging: Science & Technology*, pp. 450–461. International Society for Optics and Photonics, 1996.

Wang, Zhou and Simoncelli, Eero P. Maximum differentiation (mad) competition: A methodology for comparing computational models of perceptual quantities. *Journal of Vision*, 8(12):8, 2008.

Wang, Zhou, Simoncelli, Eero P, and Bovik, Alan C. Multi-scale structural similarity for image quality assessment. *IEEE Asilomar Conference on Signals, Systems and Computers*, 2:9–13, 2003. doi: 10.1109/ACSSC.2003.1292216.

Wang, Zhou, Bovik, Alan Conrad, Sheikh, Hamid Rahim, and Simoncelli, Eero P. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

Winkler, Stefan. A perceptual distortion metric for digital color images. In *ICIP (3)*, pp. 399–403, 1998.

6 SUPPLEMENTARY MATERIAL

6.1 EXPERIMENTAL DETAILS OF EL-DRAW TRAINING

For the experiments with the EL-DRAW model in section 4.3, we used the predefined test splits of CIFAR-10 to form a test set of 1,000 images with class dog, and randomly selected 4,500 of the training images of class dog as our training set. We used the remaining 500 images of class dog for validation.

As in the original DRAW model, we took $P(Z)$ to be a standard Gaussian with zero mean and unit variance for each latent dimension. We chose the logistic sigmoid function to be the activation applied to the final canvas, i.e. $\hat{x} = \sigma(c_T|z) = \frac{1}{1+\exp(-c_T)}$.

We used similar hyperparameters to those of the CIFAR DRAW model trained by Gregor et al. (2015): 400 hidden units for the encoder and decoder LSTM, 200 dimensions for each latent z_t, and 5x5 size for read and write operations. Our deviation in architecture was to use 32 timesteps rather than 64 in order to mitigate difficulties training the model due to exploding gradients. We clipped gradients during training by independently scaling the gradient for each weight matrix and bias vector such that the norm of each gradient was at most 10. We used the Adam method of Kingma & Ba (2014) to optimize the network.

6.2 CHOICE OF C IN EL-DRAW OBJECTIVE

The value of C in the EL-DRAW objective (Equation 5) governs the trade-off between the KL loss and reconstruction error. As C increases, the model will put greater emphasis on reconstructions. At the same time, the KL-divergence of the prior from the approximate posterior will increase, leading to poorer samples. Selecting a value of C is further complicated due to the different scaling depending on the choice of the image-specific loss Δ.

We trained MSE-optimized and SSIM-optimized EL-DRAW models with a range of values of C from 1 to 1000. We then evaluated the KL component of $\mathcal{L}^{EL-DRAW}$ on the validation set and attempted to select a setting of C for each loss that yielded comparable KL divergences. We chose $C = 10$ for MSE, yielding a validation KL loss of 52.8535 and $C = 500$ for SSIM, which yielded

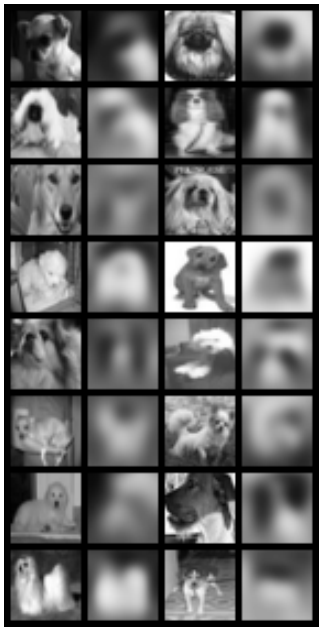

Figure 8: Test reconstructions of EL-DRAW with $C = 1$ and Δ as binary cross-entropy.

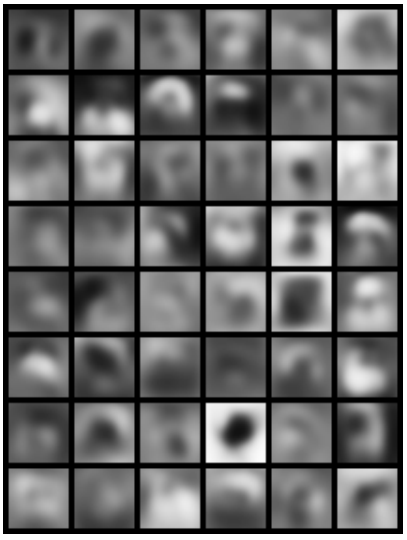

Figure 9: Samples from EL-DRAW with $C = 1$ and Δ as binary cross-entropy.

a validation KL loss of 57.4709. Thus we would expect the SSIM-optimized model to have slightly better reconstructions at the expense of slightly worse samples. However, despite this, we observe in Figure 7 that samples from the SSIM-optimized network are noticeably better.

6.3 DRAW AS A SPECIAL CASE OF EL-DRAW

As mentioned by Gregor et al. (2015), a natural choice of D for the DRAW model in the case of binary data is the Bernoulli distribution. We note that this setting of DRAW can be viewed as a special case of EL-DRAW, in which C is set to be 1 and Δ is taken to be binary cross-entropy. For the sake of completeness, we provide reconstructions and samples of EL-DRAW trained with this setting of C and Δ in Figures 8 and 9, respectively.