Faithful Autoencoder Interpolation by Shaping the Latent Space

Alon Oring 1  Zohar Yakhini 1  Yacov Hel-Or 1
School of Computer Science, The Interdisciplinary Center, Herzliya, Israel

Abstract

One of the fascinating properties of deep learning is the ability of the network to reveal the underlying factors characterizing elements in datasets of different types. Autoencoders represent an effective approach for computing these factors. Autoencoders have been studied in the context of their ability to interpolate between data points by decoding mixed latent vectors. This interpolation often incorporates disrupting artifacts or produces unrealistic images during reconstruction. We argue that these incongruities are due to latent space vectors that deviate from the data manifold and that they can be overcome by considering the manifold structure of latent spaces. In this paper, we propose two regularization techniques that encourage the latent manifold to be smooth and thus enable faithful interpolation between data points. The two methods are proposed for the supervised and for the unsupervised cases. We measure the quality of data points and demonstrate empirically that our technique improves the quality of interpolated data.

1. Introduction

The goal of artificial intelligence is to understand and analyze the world around us by inferring from acquired data. Given a set of data points, data interpolation or extrapolation aims at predicting novel data points between given samples (interpolation) or predicting novel data outside the sample range (extrapolation). Faithful data interpolation between sampled data points can be seen as a measure of the generalization capacity of a learning system (Berthelot et al., 2018). In the context of computer vision and computer graphics, data interpolation may refer to generating novel views of an object between two given views or predicting in-between animated frames from key frames. Interpolation that produces novel views of a scene may require the geometric and photometric parameters of the object, the camera parameters and additional scene components, such as lighting and the reflective characteristics of nearby objects. Unfortunately, these characteristics are not always available or are difficult to extract in real-world scenarios. Thus, in this case, it is useful to apply data-driven interpolation. That is, interpolation that is deduced based on a sampled set of instances. In computer graphics, it is common to deal with two types of renderings: Model-based rendering, that requires geometric and photometric characteristics, and Image-based rendering (IBR) where a scene is represented using a large set of images acquired under various viewing and lighting conditions (Shum et al., 2008). In image-based rendering, the stored images are used to generate novel views, typically by applying geometric and projective models. In this paper, our goal is to generate novel views where the interpolations are generated using machine learning methodologies.

The task of data interpolation aims at extracting a new set of samples (possibly continuous) between known acquired data samples. Clearly, linear interpolation between two images in the input (image) domain does not work as it produces a cross-dissolve effect between the intensities of the two images. Adopting the manifold view of the data samples (Goodfellow et al., 2016; Verma et al., 2018; Bengio et al., 2013), this task can be seen as sampling data points on the geodesic path between the given points. The problem is that this manifold is unknown in advance and one has to approximate it from the given data. Alternatively, adopting the probabilistic perspective, interpolation can be viewed as drawing samples from highly probable areas in the data space.

One of the fascinating properties of unsupervised learning is the ability of the network to reveal the underlying characteristics, determinants or factors of a discrete or continuous...
dataset. Autoencoders (Doersch, 2016; Kingma & Welling, 2013) represent an effective approach for exposing these factors. Autoencoders have demonstrated the ability to interpolate by decoding a convex sum of latent vectors (Shu et al., 2018). However, this interpolation often incorporates visible artifacts during reconstruction.

To illustrate the problem, consider the following toy example: a scene is composed of a vertical pole at the center of a flat plane. A single light source illuminates the scene and its direction can vary along the upper hemisphere. Hence, the underlying parameters controlling the generated scene are \((\theta, \phi)\), the elevation and the azimuth, respectively. The interactions between the light and the pole produce a cast shadow projected onto the plane with direction and length dictated by the light direction. A set of images of this scene is acquired from a fixed viewing position (from above) with various lighting directions. Our goal in this example is to train a model that is capable of interpolating between two given images using the given set of acquired images. Figure 1, top row, depicts a set of interpolated images where the interpolation is performed in the input domain. As illustrated, the interpolation is not natural as it produces cross-dissolve effects in image intensities. Training a standard autoencoder and applying linear interpolation in its latent space generates images that are much more realistic, as shown in Figure 1, bottom row. Nevertheless, this interpolation is not perfect as visible artifacts occur during the reconstruction. The source of these artifacts can be investigated by closely inspecting the 2D manifold embedded in the latent space.

Figure 2 shows the obtained manifold in the latent space, with data embedded into 3D (upper-left) and 2D (upper-right) latent spaces. The grid lines represent the \((\theta, \phi)\) parameterization. It can be seen that the encoder produces a non-smooth surface in 3D and a highly irregular manifold in 2D. Thus, linear interpolation between two data points may produce in-between points that leave the manifold. In practice, the decoded images of such points are unpredictable and may produce non-realistic artifacts. This is visualized in Figure 3. Reliable results are produced when the interpolation points lie on the manifold (top) but when the interpolation points depart from the manifold, the resulted image is unfaithful and includes unrealistic artifacts (bottom). In this paper, we argue that the common statistical view of autoencoders is not appropriate for data generated from continuous factors and one cannot make a precise inference on such data relying only on statistical tools. We suggest that, for better results, the manifold structure of continuous data must be considered, taking into account the geometry of the manifold. If the data is governed by \(k\)-dimensional continuous vectors then it can be viewed as a \(k\)-dimensional manifold embedded in high dimensional space (either in latent space or data space). Accordingly, we propose two data interpolation techniques as follows. If the sampled data is available with labels in the form of the underlying generating parameters (self supervised), we propose the convexity loss that penalizes interpolated latent vectors that differ from a convex latent space. Using our convexity loss, the latent space is shaped to be a linear manifold, thus linearly interpolated points are all located on the manifold. When data is available without labels, that is, without the values of the determinant \(k\)-dimensional vectors (unsupervised), convexity loss cannot be applied. In such unsupervised cases, we propose the adversarial loss where
2. Previous Work

Generative models suggest an attractive way of learning data distributions using unsupervised learning. These techniques have become very popular in recent years, demonstrating successful results. In particular, two approaches for generating new data points have become common. The Variational Autoencoder (VAE) (Kingma & Welling, 2013) and the Generative Adversarial Network (GAN) (Goodfellow et al., 2014). GAN aims at generating new samples drawn from the same distribution of a given dataset. One of the appealing properties of GAN is that it can produce reliable results even with small samples of data points (Gurumurthy et al., 2017). Although the GAN approach is highly attractive, it cannot be used directly for data interpolation due to three shortcomings: first, GAN is trained to generate random samples from the learned distribution. Thus, the generator learns a mapping from a random distribution to the desired distribution. To interpolate between two real datapoints, we must map the datapoints back into the latent domain and apply the interpolation in the latent space. Such inverse mapping is not part of the GAN framework although some attempts have been made (Radford et al., 2015). Second, the latent space of GAN does not necessarily encode a smooth parameterization of the data. There is no guarantee that applying continuous smooth interpolation in the latent space will produce faithful results in the data space. Finally, GAN is known to suffer from mode-collapse phenomena (Srivastava et al., 2017), thus latent space representations of some training images are not necessarily available.

The other approach for generative models is the autoencoder (Doersch, 2016). In its simplest version, the autoencoder is trained to obtain a reduced representation of the input data while removing data redundancies and revealing the determinant characteristics or generating factors. The reduced space can be viewed as an efficient representation space in which data interpolation can be attempted. Formally, the autoencoder is composed of two parts: the encoder \( z = f(x) \) and the decoder \( \hat{x} = g(z) \), where \( \hat{x} \) is expected to be similar to \( x \). The latent vector \( z \) is a vector whose dimension is typically much lower than that of the input space.

There are many improvements for the autoencoder that have been proposed over recent years, including new techniques designed for improved convergence and accuracy. These include introducing new regularization terms, new loss objectives (such as adversarial loss) and new network designs (Doersch, 2016; Kingma & Welling, 2013; Larsen et al., 2015; Makhzani et al., 2015; Vincent et al., 2010; Larsen et al., 2016). Other new autoencoder techniques provide frameworks that attempt to shape the latent space to be efficient with respect to factor disentanglement or to make it conducive to latent space interpolation (Kingma & Welling, 2013; Bouchacourt et al., 2017; Makhzani & Frey, 2017; Vincent et al., 2008; Yeh et al., 2016). Within this second category, the VAE was shown to be very successful in applying interpolation in the latent space.

The core idea of the VAE involves replacing the deterministic mapping from the input space to the latent space with a probabilistic mapping. This fact blurs out the resulting manifold in the latent space that corresponds to real data. Additionally, by adding a KL divergence term into the VAE loss, multi-modal distributions, such as MNIST (LeCun & Cortes, 2010), tend to cluster the modes together within the latent space (Dieng et al., 2018). Consequently, linearly interpolating between different modes in the latent space may provide pleasing results that smoothly transition between the modes. Unfortunately, this practice does not apply for data points whose generating factors are continuous (in contrast to multi-modal distributions) as the KL loss term tends to tightly fold the manifold into a compact space making it squeezed and wiggly. This phenomenon is demonstrated in Figure 2 (lower left image).

In order to address data distributions whose generating factors are continuous, (Berthelot et al., 2018) propose using a critic network to predict the interpolation parameter \( \alpha \in [0, 1] \) while an autoencoder is trained to fool the critic. The motivation behind this approach is that the interpolation parameter \( \alpha \) can be estimated for badly-interpolated images, while it is unpredictable for faithful interpolation. While this approach might work for multi-modal data, it doesn’t seem to work for data sampled from a continuous manifold. In such a case, the artifacts and the unrealistic-generated data do not provide any hint about the interpolating factor.

In this paper, we argue that the common statistical view of autoencoders is not appropriate for data generated from continuous factors and that the geometric manifold structure of the latent representation must be considered. More specifically, in order to better deal with continuous data, the geometry of the manifold must be taken into account as points outside of the manifold produce non-realistic images that often include many artifacts and inaccuracies. Accordingly, we propose two approaches for improved data interpolation by shaping the embedded manifold in the latent space.
3. The Proposed Approach

In the following we propose two approaches for data interpolation that address two different use cases. If the underlying parameters are known (self-supervised), we propose the convexity loss that penalizes interpolated latent vectors according to the corresponding parameters. When the parameters are unknown (unsupervised), we propose the adversarial loss along with the cycle-consistency loss. With the adversarial loss the autoencoder is optimized to produce "realistic" interpolations, while the cycle-consistency loss encourages the mapping from input to latent space to be bijective. In the following we explain these proposed techniques.

3.1. Supervised Manifold Interpolation

Assume the data is labeled with the associated parameters that generated it. For example, using our synthetic example, if we are given images of a scene under varying single source illumination directions, then each image is given along with its light direction \( p = (\theta, \phi) \). Our proposed architecture (see Figure 4) forces the latent manifold to be linear with respect to the given generating parameters, by adding a linearity constraint to the autoencoder loss. Denote by \( x(p) \) a sample point generated with the parameter vector \( p \). Assume we are given three data points \( x_i = x(p_i), x_j = x(p_j) \) and \( x_\alpha = x(p_\alpha) \) where \( x_\alpha \) is an intermediate point between \( x_i \) and \( x_j \) with respect to the parameter \( p \), i.e. \( p_\alpha = (1-\alpha)p_i + \alpha p_j \) where \( \alpha \in [0,1] \). These three images, \( x_i, x_j, \) and \( x_\alpha \), are inputs to an autoencoder and are mapped into three points in the latent space, respectively: \( z_i = f(x_i), z_j = f(x_j) \) and \( z_\alpha = f(x_\alpha) \). In order to force the latent space to be convex, we require that \( z_\alpha = \tilde{z}_\alpha \) where

\[
\tilde{z}_\alpha = (1-\alpha)z_i + \alpha z_j
\]

We apply this constraint by adding an additional loss to each triplet input that is given to the autoencoder. For each such triplet we require:

\[
L_{\text{triplet}} = \sum_{k \in \{i,j,\alpha\}} L_R(x_k, \hat{x}_k) + \lambda L_L(z_\alpha, \hat{z}_\alpha)
\]

where \( \hat{x}_i = g(z_i) \) is the reconstructed image after passing through the autoencoder network, \( L_R \) is a reconstruction loss (binary cross-entropy or L2 norm) and \( L_L \) is the convexity loss encouraging the latent manifold to be linear. Finally, we sum the loss over all sampled triplets. Applying the convexity loss, as suggested in Equation 2, to the toy example illustrated above (illuminated pole) results with a flat manifold as presented in Figure 2 (lower right image). The generated manifold is flattened and accordingly, the reconstruction of interpolated points is faithfully generated as will be shown in Section 4. The wrap-around of the parameterization prevents the latent space to form a linear manifold.

To avoid this effect, we "cut" the entire parameter range into several pieces (charts) that jointly cover the entire parameter space, and map each part into a linear manifold. In our case, during training, the dataset is partitioned into two parts, each of which includes a continuous half of the hemisphere along the azimuth. Each part is mapped separately into a linear manifold in the latent space.

Additionally, we experiment with an alternative approach for supervised interpolation by substituting the convexity loss in Equation 2 with an image convexity loss. In this case, we encourage the decoding \( g(\hat{z}_\alpha) \) to be similar to \( x_\alpha \). The total loss for a triplet then reads:

\[
L'_{\text{triplet}} = \sum_{k \in \{i,j\}} L_R(x_k, \hat{x}_k) + \lambda L_R(x_\alpha, g(\hat{z}_\alpha))
\]

This approach enforces latent space linearity by regularizing the outputs of the decoder, but it does not operate directly on the latent space. As will be elaborated in Section 4, this approach did not provide any benefit in terms of the faithfulness of latent space interpolation compared to the regular autoencoder or the variational autoencoder.

3.2. Unsupervised Manifold Interpolation

In cases where the data is given without order or labels, convexity loss cannot be applied. In such cases we present adversarial interpolation where we train a discriminator \( D(x) \) to differentiate between real and interpolated data. For pairs of input data-points \( x_i, x_j \), we linearly interpolate between them in the latent space: \( \hat{z}_\alpha = (1-\alpha)z_i + \alpha z_j \) where \( \alpha \in [0,1] \) and we would like \( \hat{x}_\alpha = g(\hat{z}_\alpha) \) to look real and fool the discriminator \( D \). Additionally, we add a cycle consistency loss and encourage the latent representation of \( \hat{x}_\alpha \) to be \( \hat{z}_\alpha \) again, namely \( f(g(\hat{z}_\alpha)) = \hat{z}_\alpha \). Putting
Figure 5. Left: using only the reconstruction loss guarantees faithful reconstruction of images that appear in the dataset, $x_i$, $x_j$. Middle: adding adversarial loss improves the reconstruction quality of in-between latent codes, but does not affect the encoder mapping since in-between images can be mapped to latent vectors distant from the linear line in the latent space. Right: cycle consistency loss forces the mapping of interpolated images to map back to the same latent vector, thus directly affecting the shape of the manifold and promoting bijective mapping.

### 3.3. Architecture

The chosen encoder architecture was VGG-inspired (Simonyan & Zisserman, 2014). We extract the features using convolutional blocks that start from 16 feature maps and gradually increase the number of feature maps to 128 at the last convolutional block. We then flatten the extracted fea-
tures and pass them through fully connected layers until we reach our desired latent dimensionality. The decoder architecture was symmetrical to that of the encoder. We use max-pooling after each convolutional block and batch normalization with ReLU activations after each learned layer. For the COIL100 training set, we use a random 80-20 training-testing split for each experiment. During hyper-parameter optimization, we found that $\lambda_1 = \lambda_2 = 10^{-2}$ produces the best results. All experiments were performed using a single NVIDIA V100 GPU.

4. Results

Evaluating the faithfulness of interpolation is often illusive. In the supervised case, where the exact parameterization and labels are known, for each interpolated image, $\hat{x}_\alpha$, we can retrieve the corresponding ground truth image, $x_\alpha$, and use it for evaluation. However, defining a path between two images in the parameter space $p_i, p_j$ depends on the parameterization of the underlying factors governing the data which is unknown (parameters in our toy example are $\theta, \phi$). Since such a parameterization is not unique, there are multiple viable paths in the latent space that correspond to faithful interpolation between the datapoints created by $p_i, p_j$. In the case where the latent manifold exhibits a certain parameterization in the latent space $z$, it can be adopted in the parameter space $p$ to evaluate the fidelity of interpolated images.

4.1. Dataset

We used our synthetic illuminated pole dataset in order to be able to visualize the resulting manifold in low dimensions. The images were rendered using the Unity game engine where all images are taken from a fixed viewing position (from above). A single illumination source rotates at intervals of 5 degrees along the azimuth at different altitudes, ranging from 45 to 80 degrees with respect to the plane in 5 degrees intervals. This dataset contains a total of 576 images. In order to test our methods against real images with complex geometric and photometric parameterization, we used the COIL-100 dataset (Nene et al., 1996) that contains color images of 100 objects. The objects were placed on a motorized turntable against a black background. The images were taken at pose intervals of 5 degrees resulting in a total of 72 images for each class.

4.1.1. Supervised Interpolation Evaluation

For the supervised case, we tested our approach against the synthetic dataset. We train an autoencoder using the latent convexity loss where the dimensionality of the latent space was 2D or 3D. Figure 2 (lower right image) presents the resulting manifold in 3D. It can be seen that the 2D manifold is flat and linear, satisfying the convexity constraints.

Given an interval in the parameter space defined by $(p_i, p_j)$, for some fixed $(i, j)$, we iterate over all such intervals and compare the reconstruction of the latent space interpolation starting from $z_i = f(x_i)$ to $z_j = f(x_j)$, where $x_i, x_j$ are the images with the illumination parameters $p_i, p_j$, respectively. In our evaluation, the interpolation consists out of 37 points taken along the vector $z_k = z_i + (z_j - z_i) \cdot \frac{k}{36}$, where $k \in \{0, 1, 2, ..., 36\}$ and the reconstruction error was obtained for each such point. The parameterization was linear in the parameter space.

We compare the following network architectures: regular autoencoder, variational autoencoder and our proposed methods using supervised manifold interpolation via latent convexity loss as described in Section 3.1. We also test the resulting images while applying the image convexity loss (Equation 3). Figure 7 shows the MSE of the interpolated images with respect to the ground-truth. It is demonstrated that while convexity loss in the image space does not yield a significant improvement in terms of interpolation faithfulness, latent space convexity loss considerably lowers the reconstruction error. The latent space convexity loss provides the lowest MSE in comparison to all other methods, and it does not harm the reconstruction of the boundary images $x_i, x_j$. It is also demonstrated that data interpolation in 2D latent space outperforms interpolations in 3D latent space. This outcome stems from the real intrinsic dimensionality of the manifold. This number, however, is not necessarily known in real world scenarios. Figure 8 demonstrates qualitative evaluation for our proposed architecture on our synthetic dataset. Each of the blocks represent interpolation between two fixed points $x_i, x_j$. The first and second rows of each block correspond to a naive autoencoder and a variational autoencoder, respectively. Note that the leftmost and rightmost images, that correspond to the test set images $x_i$ and $x_j$, exhibit realistic reconstruction while in-between reconstructions are unfaithful and contain

![Figure 7](image-url)

**Figure 7.** Mean squared error as a function of the interpolation variable $\alpha \in [0, 1]$ on the synthetic dataset. Reconstruction performance on end-point images are similar for all methods. Latent space convexity loss outperforms all other methods.
Faithful Autoencoder Interpolation by Shaping the Latent Space

unrealistic artifacts. The third and fourth rows correspond to the supervised and unsupervised methods. It is shown that image reconstructions appear realistic and exhibit no artifacts. Note that the parameterization in both methods is different however it does not affect the faithfulness of interpolation.

Figure 8. In each block, we perform linear interpolation between the left-most and the right-most images. The first row in each block corresponds to a naive autoencoder (AE), The second row to a variational autoencoder (VAE) and our proposed supervised and unsupervised methods are used in the third and fourth rows (P_S and P_U, respectively).

4.1.2. UNSUPERVISED INTERPOLATION EVALUATION

For the unsupervised case, we analyze our results using qualitative and quantitative measures. The bottom row of each block in Figure 8 shows the interpolation results on the synthetic dataset using our proposed method for the unsupervised case. It can be seen that we obtain realistic-looking reconstructions, comparable to the results we obtained for the supervised case. Note, however, that it is impossible to define a "correct" geodesic path between two points since the parameterization of the dataset is redundant. Thus, interpolation results of our supervised and unsupervised methods are realistic and without artifacts yet produce different parameterization in latent space. In both cases, the autoencoders were trained using a three-dimensional latent space. Figure 10 visualizes the resulting latent manifold for the unsupervised case. The learned manifold is mostly flat however not as smooth as in the supervised case. Note that the different parameterization obtained in the unsupervised case does not allow us to accurately quantify the reconstruction error. We tested the unsupervised interpolation method on real images from the COIL-100 dataset. Since those images exhibit complex geometric and reflectance variations, for the sake of faithful reconstruction, dimensionality of the latent space had to be increased to 256. Figure 9 compares the resulting images of latent interpolation on COIL-100 dataset using three methods: naive autoencoder, a variational autoencoder and our proposed unsupervised method. It is demonstrated that when the interpolated images approach the middle range between \( x_i, x_j \), the reconstruction
Faithful Autoencoder Interpolation by Shaping the Latent Space

Figure 11. Linear latent space interpolation on COIL-100 dataset. At each row, the images at the rightmost and leftmost sides are \( x_i, x_j \), respectively. We present the resulting interpolations of the naive autoencoder (AE), variational autoencoder (VAE) and our proposed supervised (P_S) and unsupervised (P_U) methods.

Our method considerably reduces those artifacts. Figure 11 demonstrates interpolation results using all methods on a COIL-100 object. For a quantitative comparison, we fix an interval length, which is a multiplicative of 5 degrees, and calculate the reconstruction error against the available ground-truth images. In this experiment, we use an interval length of 80 degrees and interpolate 14 in-between images. The reconstruction error of interpolated images is presented in Figure 13. It is demonstrated that our method reduces both the mean squared error and the standard deviation of the MSE for different alpha values. We then inspect the average reconstruction error on a range of intervals from 25 degrees to 70 degrees as presented at the bottom part of Figure 13. Note that our proposed method is able to consistently reduce the reconstruction error of interpolated images even when the interval length increases.

Figure 12. Examples of a non-uniform parameterization in latent space. The interpolation changes abruptly without smoothly transitioning from one mode to another. Note that our unsupervised technique interpolates without artifacts on the top row while the VAE and AE exhibit unrealistic artifacts during the transition.

4.2. Conclusion & Discussion

The problem of realistic and faithful interpolation in the latent spaces of generative models has achieved tremendous success in the last few years. We argue that generative approaches that deal with manifold data are not as common as multi-modal data, and this misinterpretation of manifold data harms the competence of generative models to deal with them successfully. In this work, we argue that the manifold structures of data generated from continuous factors should be taken into account. Our main contribution is generalizing our supervised approach for the unsupervised case by applying convexity regularization using adversarial and cycle consistency losses. Using this technique, we manage to drastically improve the fidelity of interpolated images using a small dataset of images taken from various viewing directions. In future work, we intend to further investigate latent spaces that exhibit non-uniform parameterization as visualized in Figure 12 by implementing a directional derivative loss along the interpolation vector. In addition, we intend to develop a well-defined metric for performance evaluation of generative models capable of faithful interpolation.
References

Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

Berthelot, D., Raffel, C., Roy, A., and Goodfellow, I. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *arXiv preprint arXiv:1807.07543*, 2018.

Bouchacourt, D., Tomioka, R., and Nowozin, S. Multi-level variational autoencoder: Learning disentangled representations from grouped observations. *arXiv preprint arXiv:1705.08841*, 2017.

Dieng, A. B., Kim, Y., Rush, A. M., and Blei, D. M. Avoiding latent variable collapse with generative skip models. *CoRR*, abs/1807.04863, 2018.

Doersch, C. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in neural information processing systems*, pp. 2672–2680, 2014.

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. *Deep learning*, volume 1. MIT press Cambridge, 2016.

Gurumurthy, S., Sarvadevabhatla, R. K., and Babu, R. V. Deligan: Generative adversarial networks for diverse and limited data. In *CVPR*, pp. 4941–4949. IEEE Computer Society, 2017. ISBN 978-1-5386-0457-1.

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

Larsen, A. B. L., Sønderby, S. K., Larochelle, H., and Winther, O. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

Larsen, A. B. L., Sønderby, S. K., Larochelle, H., and Winther, O. Autoencoding beyond pixels using a learned similarity metric. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pp. 1558–1566. JMLR.org, 2016. URL http://proceedings.mlr.press/v48/.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. 2010. URL http://yann.lecun.com/exdb/mnist/.

Makhzani, A., and Frey, B. J. Pixelgan autoencoders. In *Advances in Neural Information Processing Systems*, pp. 1975–1985, 2017.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., and Frey, B. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

Nene, S. A., Nayar, S. K., and Murase, H. object image library (coil-100). Technical report, 1996.

Radford, A., Metz, L., and Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

Shu, Z., Sahasrabudhe, M., Alp Guler, R., Samaras, D., Paragios, N., and Kokkinos, I. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *The European Conference on Computer Vision (ECCV)*, September 2018.

Shunn, H.-Y., Chan, S.-C., and Kang, S. B. *Image-based rendering*. Springer Science & Business Media, 2008.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL http://arxiv.org/abs/1409.1556.

Srivastava, A., Valkov, L., Russell, C., Gutmann, M. U., and Sutton, C. Veegan: Reducing mode collapse in gans using implicit variational learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 3308–3318. Curran Associates, Inc., 2017.

Verma, V., Lamb, A., Beckham, C., Courville, A., Mitliagkis, I., and Bengio, Y. Manifold mixup: Encouraging meaningful on-manifold interpolation as a regularizer. *arXiv preprint arXiv:1806.05236*, 2018.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103. ACM, 2008.

Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., and Manzagol, P.-A. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(Dec):3371–3408, 2010.

Yeh, R., Liu, Z., Goldman, D. B., and Agarwala, A. Semantic facial expression editing using autoencoded flow. *arXiv preprint arXiv:1611.09961*, 2016.