An Augmentative Interpolative Autoencoder

MVA — Deep Learning: Final Project Report

Daniil Lotkov
ENS Paris-Saclay

Pierre Ebert
ENS Paris-Saclay, University of Paris

Abstract

The aim of this project was to develop an image generation model that, once trained on an initial domain, could generate images from a different domain based on only a few seed images of this new domain. Following the ideas of Anonymous Authors [1], we implemented the following autoencoders: a classical autoencoder, VanillaAE, an Augmentation-Interpolative autoencoder (AugIntAE) which comprised both random augmentation and interpolation of input images and a second AugIntAE which incorporated an adversarial loss. All three were trained on the MNIST domain.

After testing the three AEs on MNIST reconstruction and random image generation tasks, the autoencoders were given images from different domains in order to attempt few-shot image generation. With EMNIST seeds, the Adversarial AugIntAE was superior to its competitors in terms of Fréchet Inception Distance, successfully producing relatively realistic EMNIST images from the seeds. AugIntAE also produced credible EMNIST images from the seeds, although of a lesser quality than those of the adversarial model, and VanillaAE failed to generalise successfully to this new domain. The autoencoders were unsuccessful at few-shot image generation with Fashion MNIST seeds, indicating that they are only capable of realistic few-shot generation on domains close to the original training domain.

1. Introduction

Current generative models are able to produce synthetic images which are highly realistic, random and of high quality in terms of resolution. Such models can generate images from any domain provided that a sufficiently large dataset for the domain in question was available for training. For a specific domain, such a dataset does not however always exist and this is the case for many domains.

For example, autonomous driving training videos may only contain data captured during average weather conditions but may not comprise data from extreme driving conditions. Researchers developing a medical imaging classifier may not have access to a sufficient number of samples for certain classes and, in order to remedy the lack of samples in these classes, may want to augment the few samples at their disposal. Finally, designers may wish to use generative models to help them to design future products based on a small number of innovative images they have created.

From these examples, it is clear that there is a need for generative models which can be trained on a certain training domain for which sufficient data is available and, when provided with a few images from a new domain, are able to generate high-quality, diverse images in this new domain. This is the concept of *few-shot image generation* for which Anonymous Authors [1] have developed a solution.

2. Problem Definition

2.1. Reasoning behind the use of an AE

Most current generative models are based on variational autoencoders (VAEs) or generative adversarial networks (GANs). The latent spaces of such models are however known not to generalise well to new classes and domains without training. According to Anonymous Authors [1], the latent spaces of autoencoders (AEs) however generalise well to new classes and can therefore be used to generate images of a new class by interpolating the seed images but fail to generate images outside the initial training domain. Based on interpolative AEs, Anonymous Authors[1] have developed a novel method to generate images in new domains. This method is successful because it uses image augmentation and interpolation of image pairs which are semantically linked, instead of interpolating between arbitrary image pairs. The model then is trained on simpler, more effective interpolations that generalise better to new domains.

2.2. Problem definition

Our aim is to implement a few-shot image generator similar to that designed by Anonymous Authors [1] and assess its performance on a variety of tasks and datasets.

This model will be assessed on multiples tasks:

1. Reconstruction of images from the training domain
2. Image generation from a random latent vector
3. Few-shot image generation from seeds of a new domain

Formally, we define X as the space of images (of fixed 28x28 dimensions) and we define L as an un-annotated set of images from a set of classes C(in practice this is the MNIST dataset). We define S as a set of images from a class c such that $c \notin C$ (this is the EMNIST dataset). An autoencoder model f, with θ its parameters, is then trained on L. For the first task in the list above, the aim will be minimise the Mean Square Error (MSE) between the generated images and the input images from the training domain, *i.e.* we randomly split L into a training set L_{train} and a test set L_{test} and our objective is to find θ that minimises:

$$\sum (l - f_\theta(l))^2$$

$\forall l \in L_{test}$.

As L_{test} is the test set, the parameters θ of the model are only updated on the training phase on set L_{train}.

Let $e : L \to Z$ and $g : Z \to X$ be the encoder and decoder that compose f, such that $f(l) = g(e(l))$. Let $FID(x, y)$ be the Fréchet Inception Distance which measures the general quality of the generated images compared to the training images. For the second task, the aim is to train f in order to use the generator component g to minimise:

$$FID(g_\theta(z), L)$$

with z drawn from a normal distribution $\mathcal{N}(\mu_\theta, \Sigma_\theta)$ with μ_θ and $\Sigma\theta$ produced by the encoder e_θ.

Finally, for the third task, we aim to minimise for $s \in S$:

$$FID(f_\theta(s), S)$$

The model f will be trained and tested for the different tasks on different datasets, namely:

- MNIST [9]
- EMNIST [4]
- Fashion MNIST [8]

3. Related work

Most current generative models are based on VAEs or GANs. The latent spaces of such models are known not to generalise well to new classes without training on them beforehand.

Sainburg *et al.* [5] have investigated the use of an interpolative adversarial autoencoder. In their model, the generator is trained on interpolated samples in the latent space and both a pixel-wise loss and a discriminator are used to improve the smoothness and convexity of the latent space.

Berthelot *et al.* [2], proposed to implement a regularisation method to force the autoencoder to create more realistic interpolated output images. This regularisation method involves using a discriminator network to predict the mixing coefficient α from the interpolated data. This method has been shown to strongly improve the quality of the interpolated output.

Verma *et al.* [7] recognise that CNNs do not give satisfactory predictions when faced with test data which is different in terms of distribution to its training samples, whether the test data is simply an outlier, is generated adversarialy or comes from an entirely new domain. The authors here propose to randomly select a layer k within the CNN and process two distinct batches independently through the CNN until layer k at which point random linear interpolation is performed on both batches. The resulting interpolated batch is then processed through the rest of the network. Such models learn less variance directions for each class and therefore generalise better.

Anonymous Authors [1] conclude that interpolative autoencoders may be able to generate new combinations of the initial images, using an adversarial loss as suggested by Sainburg *et al.* [5] and Berthelot *et al.* [2] but that producing realistic images from the interpolation of two samples from different classes is highly difficult. The authors find that interpolating independently augmented images from the same class can combine the generalisation strength of autoencoders, the generative power of interpolation and the simplicity of training the model to reconstruct the interpolation of two samples from the same class, instead of different classes. They infer that the combination of these characteristics can allow an augmentation-interpolative autoencoder to successfully generate images in a new domain from a small number seed images.

4. Methodology

The following subsections present the architecture of the AugIntAE and Adversarial AugIntAE models we have implemented and the associated losses and training methods.

4.1. Architecture

The general architecture of f is that of a classical CNN-based autoencoder. The model begins with one batch of images X as an input. Two sets of data augmentations, λ_1 and λ_2 are then randomly and independently sampled. The augmentations and associated ranges of values are presented in Table 1. These augmentations are then applied to the batch, yielding two augmented, different batches, X_1 and X_2. The two batches are then both run through the encoder component e of the model f yielding two batches of latent vectors z_1 and z_2. An interpolation coefficient α is then randomly drawn from an uniform distribution on $[0, 1]$ and used to linearly interpolate the latent vectors to obtain z, *i.e.* $z := \alpha z_1 + (1-\alpha) z_2$. This batch of interpolated latent vectors is then forwarded through the generator component g of the model f which yields a batch of images X'.

Transformation	Distribution	Range
Rotation	Uniform	$[-10, 10]$
Translation	Uniform	$[-5, 5]^2$
Scale	Uniform	$[0.9, 1.1]$
Normalisation	Deterministic	-

Table 1. Stochastic image augmentations

In parallel to this forward pass, X_1 and X_2 are interpolated using the same α to obtain a batch of images Y.

The specific architecture of the encoder component is detailed in Table 2, with a latent vector dimension of 10. This architecture is inspired from that of InfoGAN developed by Chen *et al.* [3] for the MNIST dataset.

Operation	Parameters	
Convolution (2D)	Out channel: 64, kernel size: 4	
Leaky ReLU		
Convolution (2D)	Out channel: 128, kernel size: 4	
Leaky ReLU		
Batch Norm (2D)		
Flatten to vector	Output: 128 x 7 x 7	
Linear (FC)	Output: 64 x 7 x 7	
Leaky ReLU		
Linear (FC)	Encoder Output: 10	Discrim. Output: 1

Table 2. Architecture of the encoder/discriminator

The specific architecture of the decoder component is symmetrical to that of the encoder, as shown in Table 3.

Operation	Parameters
Linear (FC)	In: 10 Out: 64 x 7 x 7
ReLU	
Batch Norm (1D)	
Linear (FC)	In: 64 x 7 x 7 Out: 2 x 64 x 7 x 7
ReLU	
Batch Norm (1D)	
Unflatten to matrix	
Transposed Convolution	In: 2 x 64 Out: 64
Batch Norm (2D)	
Transposed Convolution	In: 64 Out: 1
tanh	

Table 3. Architecture of the decoder

4.2. Losses

4.2.1 AugIntAE

The aim of the AugIntAE model during training is to minimise the difference between the augmented-interpolated batch Y and the batch X' generated from the interpolated latent vector batches. A reconstruction loss L_{recon} is used to train the autoencoder f to reconstruct the interpolated augmented input image Y. Here, the MSE loss is used, yielding for one batch:

$$L_{recon}(X_1, X_2, Y) = ||Y - g(\alpha e(X_1) + (1 - \alpha e(X_2)))||_2^2$$

4.2.2 Adversarial AugIntAE

In addition to L_{recon} which forces the AE to reconstruct the input images realistically, Adversarial AugIntAE uses an adversarial loss during the training phase. For this, a discriminator network is created, identical to the encoder component of f except for the last layer which outputs a single value (instead of a latent vector). The following cross-entropy loss is used to train the discriminator to distinguish between true augmented-interpolated samples and the reconstruction of augmented-interpolated samples by the AE. Formally, at a batch level, we have:

$$\begin{aligned} L_{adv}(X', Y) = \frac{1}{N}\sum_{i=1}^{N} d_i \log(x'_i) + (1 - d_i)\log((1 - x'_i)) \\ + \frac{1}{N}\sum_{i=1}^{N} d_i \log(y'_i) + (1 - d_i)\log((1 - y'_i)) \end{aligned}$$

with $d_i \in \{0, 1\}$, with 1 the label associated with true interpolated images and 0 the label associated with the generated interpolated images.

For this model, the parameters of the network f and the discriminator are updated at each epoch with respect to L_{adv}, whereas the parameters of the network f are updated every five epochs with respect to L_{recon}.

4.2.3 VanillaAE

The VanillaAE model uses the same architecture as the other two autoencoders but does not comprise augmentations and interpolations: a single, un-augmented batch is entered into the network. The L_{recon} loss is used for this model.

4.3. Training

The models are trained on the MNIST dataset for 50 epochs with the Adam optimiser, a learning rate of $1e^3$ and a batch size of 128.

4.4. Generation

In this subsection, we refer back to the tasks outlined in the problem definition section.

For the reconstruction task, test images from the MNIST dataset are simply entered as batches into the trained AE models and the output is compared to the initial images.

For the random image generation task, images are entered into the encoder and statistics (mean μ and standard variation σ) are calculated from the sample of latent vectors produced by the encoder. Random vectors are then drawn from a normal distribution $\mathcal{N}(\mu_\theta, \Sigma_\theta)$ and forwarded through the decoder to produce random images.

Finally, for the few-shot image generation task, seed images from a new dataset (EMNIST or Fashion MNIST) are forwarded through the models, generating novel images which should be belong to the new domain.

5. Evaluation

5.1. Image reconstruction

Autoencoders were initially developed to compress and decompress files and it is therefore not a surprise that they possess strong reconstruction qualities. For this task, we have compared the performance of our three AE models.

The following figure shows the training error curves over the number of epochs of the training phase of the models.

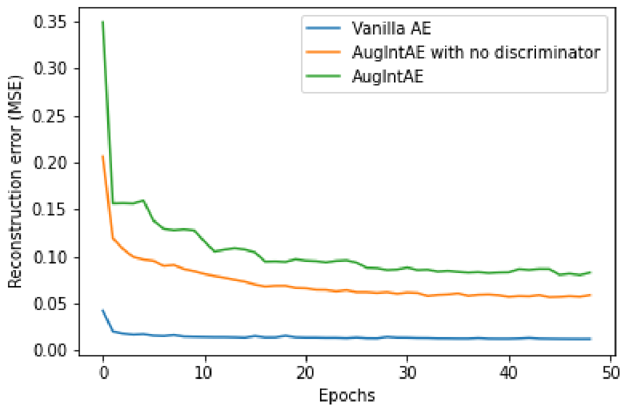


Figure 1. Training error (measured as MSE) over MNIST

The three AEs were then tested on the MNIST test set, the results of which are in Table 4.

Autoencoder model	MSE on MNIST test set
VanillaAE	0.041
AugIntAE	0.076
Adversarial AugIntAE	0.090

Table 4. Reconstruction error (MSE)

After 50 epochs of training, it is clear that VanillaAE is able to reconstruct MNIST images from the test dataset significantly better than the other two AEs. Indeed, over 50 epochs, VanillaAE focuses only on reconstructing images from the training dataset and its parameters are updated every epoch through backpropagation based on the MSE loss (which is used to evaluate the test reconstruction performance). The AugIntAE is trained on augmented and interpolated images, whose distribution is further away from the test set. It is logical that AugIntAE's reconstruction performance is below that of VanillaAE. Furthermore, the training mechanism of Adversarial AugIntAE is even more different, in the sense that it also trains on augmented and interpolated images, but that the parameters of network are updated at each epoch with respect to the adversarial loss and only every five epochs for the reconstruction loss. Consequently, Adversarial AugIntAE's performance on the reconstruction task is sub-par.

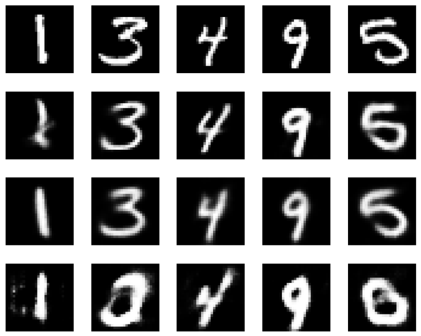


Figure 2. Reconstructed Images. 1st row: Initial 2nd row: VanillaAE 3rd row: AugIntAE 4th row: Adversarial AugIntAE

The conclusions made on the reconstruction performance of the three AEs in this comparison can be observed in the reconstructed images of Figure 2 with VanillaAE and AugIntAE images being far more realistic than those of Adversarial AugIntAE, which does not succeed to consistently reconstruct the original image.

5.2. Random image generation

In this task, the autoencoders are used to generate images from random latent vectors.

Table 5 shows the Fréchet Inception Distance of the generated images relative to images in the true dataset, calculated with a FID-calculator developed by Seitzer [6].

Autoencoder model	FID on 2048 images to the MNIST test set
VanillaAE	52.53
AugIntAE	90.54
Adversarial AugIntAE	111.29

Table 5. Fréchet Inception Distance of generated images

These results indicate that the FID of the images generated by VanillaAE are closer in distribution to the original handwritten digits in the MNIST test set than those generated by the other two AEs. This is not surprising as, similarly to the reconstruction task, AugIntAE and Adversarial AugIntAE are trained on augmented and interpolated images which may be quite different in distribution to traditional MNIST images. Figure 3 shows the generated images from random latent vectors for the three autoencoders in our comparison. These digits may look poor compared to those obtained with standard MNIST GANs or MNIST VAEs but it must be noted that none of the three models compared here are specifically designed for image generation from random latent vectors. The difference in quality between the images generated by VanillaAE and the two other AEs is not obvious to the human eye.

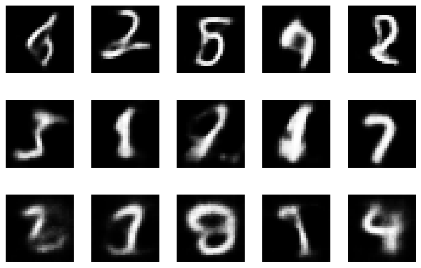


Figure 3. Generated Images. 1st row: VanillaAE 2nd row: AugIntAE 3rd row: Adversarial AugIntAE

5.3. Few-shot image generation in the EMNIST domain

Finally, the autoencoders, trained on the MNIST dataset with a latent vector size of 32, were used to generate images using seed images from other domains, namely EMNIST [4] and Fashion MNIST (FMNIST) [8]. The FIDs between their output and the EMNIST and Fashion MNIST datasets respectively were calculated over 2048 images.

Autoencoder type	FID to EMNIST
VanillaAE	78.67
AugIntAE	60.27
Adversarial AugIntAE	58.49

Table 6. FID of the AEs trained on MNIST but seeded with EMNIST and FMNIST inputs

Table 6 indicates that the FID of few-shot images generated by Adversarial AugIntAE to the EMNIST dataset is lower than that of VanillaAE and AugIntAE. The lower FIDs of Adversarial AugIntAE and AugIntAE relative to that of VanillaAE were expected as, although their reconstruction capabilities are lower on MNIST than VanillaAE as shown in subsection 5.1, they have been exposed to a greater variety of images during their training, due to both the random augmentations and interpolation in the latent space. It can therefore be said that Adversarial AugIntAE and AugIntAE were better prepared during their training to the move to the EMNIST domain generation, yielding more realistic images in Figure 4 than VanillaAE which did not benefit from such variety in the training set. Furthermore, the difference in FID between Adversarial AugIntAE and AugIntAE is significant. Indeed, Adversarial AugIntAE and AugIntAE differ greatly in their training procedure. AugIntAE's parameters are updated at every epoch with respect to the gradient of the MSE loss, whereas Adversarial AugIntAE's parameters are updated with respect to the gradient of the adversarial cross-entropy loss at every epoch (as well as updating the discriminator) and to the gradient of the MSE loss only every five epochs. From these FID metrics, it appears that Adversarial AugIntAE is better trained for few-shot generation due to the adversarial component of its training with the discriminator forcing it out of known image distributions and preparing it for few-shot generation in a different domain.

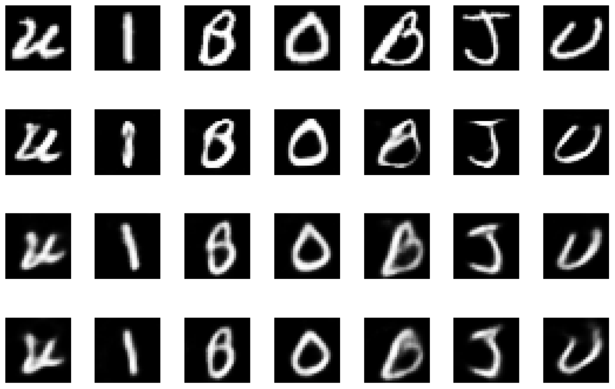


Figure 4. Few-shot EMNIST Images. 1st row: Seed images 2nd row: VanillaAE 3rd row: AugIntAE 4th row: Adversarial AugIntAE

From Figure 4 however, the superiority of Adversarial AugIntAE is not obvious. Adversarial AugIntAE and AugIntAE possess random generator qualities with a dual level of randomness embedded in their models: random augmentation of the seed images prior to input and random interpolation in the latent space. In this sense, they can be qualified as *few-shot generators*.

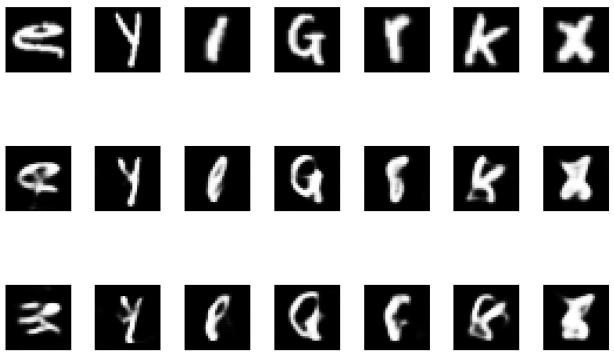


Figure 5. Few-shot Images from VanillaAE. 1st row: Seed images 2nd row: VanillaAE 3rd row: VanillaAE with augmentation and interpolation

Figure 5 compares the few-shot generation of VanillaAE in two different configurations. On the first row are images generated by VanillaAE seeded with a single EMNIST image, whereas on the second row are images generated by VanillaAE seeded with two augmented images and interpolated in the latent space. As VanillaAE has been trained on MNIST in the first configuration, it is logical that it should perform well in few-shot generation for this configuration and poorly in the second configuration, as shown in the figure. Although reconstruction is good, VanillaAE does not incorporate any random component (no augmentations or random interpolations), the output images cannot be considered randomly generated. Moreover, as we have seen in Figure 5, VanillaAE is not able to generate realistic images from augmented interpolated seeds. In this sense, VanillaAE should be qualified as a *few-shot reconstructor*. Applying random augmentation to the training set (but not interpolating) would further improve VanillaAE's few-shot reconstruction performance. Further few-shot images are provided in Appendix A.

5.4. Few-shot image generation in the Fashion MNIST domain

This last section investigates the few-shot generation of images in a domain distant from the training domain. Indeed, both MNIST and EMNIST are handwritten signs which are close in terms of image distribution. Fashion MNIST [8] is quite different as a domain, with images representing different clothes classes. Figure 6 shows that all three AEs developed in this project have failed to generate realistic clothes images, with Adversarial AugIntAE transforming a shoe into a very convincing number "4". This brief investigation highlights the limits of our few-shot AEs when faced with seed images from very different domains.

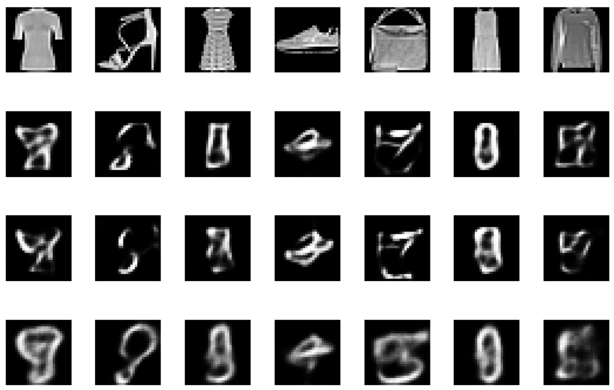


Figure 6. Few-shot Fashion MNIST Images. 1st row: Seed images 2nd row: VanillaAE 3rd row: AugIntAE 4th row: Adversarial AugIntAE

6. Conclusion

In this report, we have implemented three generative models: a classical autoencoder, VanillaAE, and two Augmentation-Interpolative autoencoders as originally designed by Anonymous Authors [1], namely AugIntAE which used a reconstruction loss and Adversarial AugIntAE which incorporated an adversarial loss. After extensive experimentation on these last two autoencoders, the three autoencoders were trained on the MNIST dataset and tested on three tasks. The autoencoders were first used to encode and reconstruct MNIST digits from the MNIST test set, with VanillaAE providing the best MSE reconstruction, as its architecture was specifically conceived for this task. The autoencoders were then used to generate random images from random vectors sampled in the latent space. Again, VanillaAE outperformed its competitors with its generated images having a lower FID to the MNIST dataset. Finally, the autoencoders were given images from different domains in order to attempt few-shot image generation. With EMNIST seeds, the Adversarial AugIntAE was superior to the others in terms of FID, successfully generating relatively realistic EMNIST images from the seeds. The autoencoders were unsuccessful at few-shot image generation with Fashion MNIST seeds, which indicates that they are only capable of realistic few-shot generation on domains close to the initial training domain.

References

[1] Anonymous Authors. Augmentation-interpolative autoencoders for unsupervised few-shot image generation. *Under review as a conference paper at ICLR 2021*, 2021.

[2] David Berthelot, Colin Raffel, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer, 2018.

[3] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets, 2016.

[4] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters, 2017.

[5] Tim Sainburg, Marvin Thielk, Brad Theilman, Benjamin Migliori, and Timothy Gentner. Generative adversarial interpolative autoencoding: adversarial training on latent space interpolations encourage convex latent distributions, 2019.

[6] Maximilian Seitzer. pytorch-fid: FID Score for PyTorch. `https://github.com/mseitzer/pytorch-fid`, August 2020. Version 0.1.1.

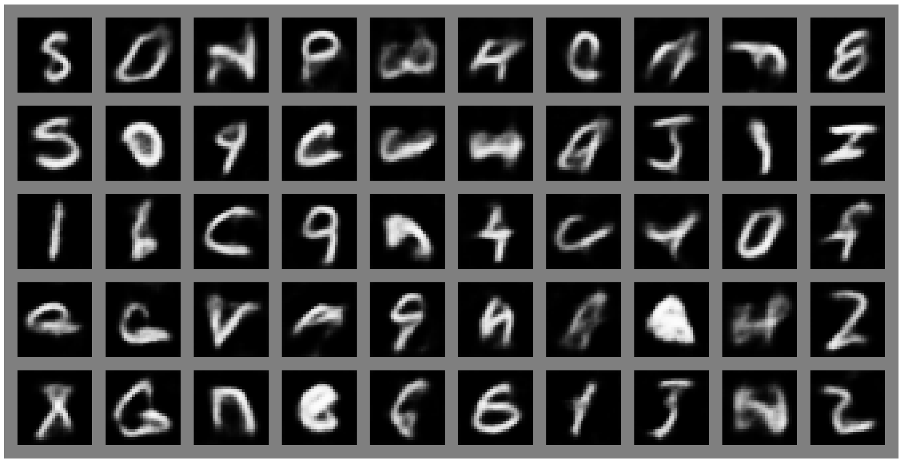
[7] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, Aaron Courville, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states, 2019.

[8] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

[9] Corinna Cortes Yann LeCun and Christopher J. C. Burges. The mnist database of handwritten digits. `http://yann.lecun.com/exdb/mnist/`, August 1998.

A. Appendix

Seed EMNIST Images



Few-shot EMNIST Images using Adversarial AugIntAE