# Unsupervised Anomaly Detection with Adversarial Mirrored AutoEncoders (Supplementary material)

**Gowthami Somepalli**[1] **Yexin Wu**[2] **Yogesh Balaji**[1] **Bhanukiran Vinzamuri**[3] **Soheil Feizi**[1]

[1]{gowthami,yogesh,soheil}@cs.umd.edu , University of Maryland, College Park,
[2]yw2423@cornell.edu , Cornell University ,
[3] bhanu.vinzamuri@ibm.com , IBM Research,

## A PROOF OF LEMMA 1

**Lemma 1** *If E and G are optimal encoder and generator networks, i.e.,* $\mathbb{P}_{X,\mathbf{G}(\mathbf{E}(X))} = \mathbb{P}_{X,X}$*, then* $\mathbf{x} = \mathbf{G}(\mathbf{E}(\mathbf{x}))$

**Proof:**

$$\begin{aligned} \mathbb{P}_{X,\hat{X}} &= \mathbb{P}_{X,X} \\ p(\hat{x} = k) &= \int_x p(x, \hat{x} = k) dx \\ &= p(x = k, \hat{x} = k) \end{aligned}$$

Therefore $p(\hat{x} = k)$ is meaningful only if when $\hat{x} = x$, which means $\mathbf{G}(\mathbf{E}(x)) = \mathrm{x}$

## B ARCHITECTURE AND TRAINING DETAILS

We present the complete architecture information of Adversarial Mirrored AutoEncoder (AMA) for CIFAR-10 and SVHN experiments in Table-1 and for Fashion MNIST and MNIST experiments in Table-2. We have used Spectral GAN normalization only in CIFAR-10; SVHN experiments where, in the discriminator $\mathbf{D}$, BatchNorm is replaced by Spectral Norm similar as implemented in SN-GAN paper Miyato et al. [2018].

The whole pipe-line of our model, AMA, is trained end-to-end with Adam optimizer with $\beta_1 = 0$ and $\beta_2 = 0.9$ for Generator and Discriminator and $\beta_1 = 0.5$ and $\beta_2 = 0.9$ for Encoder, initial learning rate of 3e-4 and decaying it by a factor of 0.1 at 30, 60 and 90 training epochs. We trained each model for 100 epochs with a batch size of 256 for all the datasets. If Atypical selection is enabled, we train the model for first the 10 epochs only on normal samples, and from $11^{th}$ epoch onward we generate synthetic anomalies and use them along with normal samples in training. We use $\lambda_{inter} = 0.5$, $\lambda_{neg} = 5$ and $\lambda_{reg} = 1$ in all of our CIFAR-10 and SVHN experiments. Refer to the appendix for the hyperparameter values of MNIST experiments. In OOD experiments, for CIFAR-10, we sampled for synthetic anomalies *inward* and for SVHN and Fashion MNIST we sampled *outward*. Experiments are performed using two NVIDIA GTX-2080TI GPUs.

## C LATENT SPACE REGULARIZATION ABLATIONS

**Simplex Interpolation vs MixUp** MixUp Zhang et al. [2017] is a popular augmentation technique based on 2-point interpolation in image space. So we have performed a comparative analysis of Simplex Interpolation in latent space with MixUp in image space as shown in Fig. 1 using AMA architecture. In this figure, we show the results when the anomalies arise within the same dataset i.e. CIFAR-10 where one class is considered normal and the rest 9 as anomalous. Except for 2 classes ('dog' and 'airplane'), AMA with Simplex Interpolation outperformed AMA with MixUp in the rest. On average, AMA with Simplex Interpolation model has 4% AUROC absolute gain over AMA with Mixup.

**Atypical selection vs Sipple 2020** Performance comparison of Atypical Selection against Negative Sampling proposed in Sipple [2020] is presented in Table 3. Atypical Selection outperforms Sipple [2020]'s technique in all studied cases. We hypothesize that, since Atypical Selection samples near the boundary of the latent space, it enforces the encoder to create more compact latent space for normal samples.

## D ADDITIONAL ANALYSIS AND RESULTS

**Results on Tabular data:** We present the performance of AMA and a few baselines on two tabular datasets Arrhyth-

| Generator | Encoder | Discriminator |
|---|---|---|
| $\mathbf{z} \in \mathbb{R}^{128}$ | $x \in \mathbb{R}^{32\times32\times3}$ | $(x, \hat{x}) \in \mathbb{R}^{32\times32\times6}$ |
| Dense, 4*4*256 | ((4,4),2,1,64) | ResBlock down 128 |
| ResBlock up 256 | BN + LeakyReLU(0.2) | ResBlock down 128 |
| ResBlock up 256 | ((4,4),2,1,128) | ResBlock 128 |
| ResBlock up 256 | BN + LeakyReLU(0.2) | ResBlock 128 |
| BN,ReLU,3 $\times$ 3 Conv, Tanh | ((4,4),2,1,256) | ReLU |
| Adam($\beta_1$= 0, $\beta_2$= 0.999) | BN + LeakyReLU(0.2) | Global sum pooling |
| | ((4,4),1,0,128) | dense $\rightarrow$ 1 |
| | Adam($\beta_1$= 0.5, $\beta_2$= 0.999) | Adam($\beta_1$= 0, $\beta_2$= 0.999) |

LR = 3e-4, Batch Size = 256, Epoch=100, $\lambda_{inter} = 0.5$, $\lambda_{neg} = 5$, $\lambda_{reg} = 1$ and $\delta = 0.5$.

Table 1: CIFAR-10 and SVHN Architecture Detail. We use architecture from Miyato et al. [2018] and Gulrajani et al. [2017], Encoder Structure is represented as (filter_size,stride,padding,output_ channel). The input to the discriminator is $(x, x)$ or $(x, \hat{x})$, where images are stacked on the channel dimension. $\hat{x}$ is the reconstruction of $x$. Atypical Selection done inward for CIFAR-10 experiments while it is done outward for SVHN experiments.

| Generator | Encoder | Discriminator |
|---|---|---|
| $\mathbf{z} \in \mathbb{R}^{64}$ | $x \in \mathbb{R}^{28\times28\times1}$ | $(x, \hat{x}) \in \mathbb{R}^{28\times28\times2}$ |
| Linear(64,1024)+ReLU | Conv((4,4),2,2,64) + RELU | Conv((4,4),2,2,64) + |
| Linear(1024,7*7*128) | Conv((4,4),2,2,128) + RELU | LeakyReLU(0.2) |
| +ReLU | Dense(1024) + RELU | Conv((4,4),2,2,128) + |
| ConvTranspose2d(4,4),2,2,64 + | Dense(64) | LeakyReLU(0.2) |
| RELU | | Dense(1024) + LeakyReLU(0.2) |
| ConvTranspose2d(4,4),2,2,1 + Tanh | | Dense(1) |

LR = 3e-4, Batch Size = 256, Epoch=100, $\lambda_{inter} = 0.5$, $\lambda_{neg} = 5$, $\lambda_{reg} = 1$ and $\delta = 0.5$. Atypical Selection done outward.

Table 2: FashionMNIST and MNIST Architecture Detail for OOD as well as the within-dataset anomalies experiments.

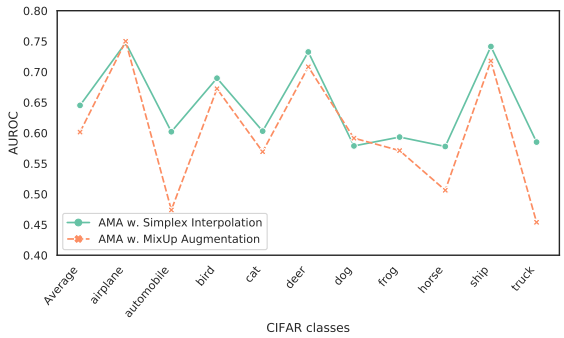



Figure 1: **Comparative analysis - Simplex Interpolation vs MixUp augmentation** when either of them are used in AMA during training. The experiments are performed on CIFAR-10 dataset with respective class on X-axis considered normal while rest of the 9 classes considered anomalous. We observe that Simplex Interpolation in training leads to a better model than using MixUp augmentation. The performance is reported in AUC scores, higher the better.

Table 3: In this table we present the performance of various models trained using Negative selection proposed by Sipple [Sipple, 2020] *vs* Atypical Selection proposed by us, in presence and absence of Simplex interpolation. Values reported are AUROC scores in format Sipple / Atypical Selection

| Experiment | No interpolation | With interpolation |
|---|---|---|
| FashionMNIST *vs* MNIST | 0.778 /**0.960** | 0.824 / **0.987** |
| CIFAR-10 *vs* SVHN | 0.752 / **0.820** | 0.819 / **0.958** |
| SVHN *vs* CIFAR-10 | 0.723/**0.991** | 0.896/**0.993** |

mia[1] and KDD99[2] in Table 4.

**Loss function ablations:** In the main paper, we have shown both qualitatively and quantitatively how Mirrored Wasserstein loss improves the performance compared to Wasserstein loss. We explored a different paradigm where we add an $L_2$ loss to the regular Wasserstein loss and see how the performance changes. We observe that the $L_2$ loss term indeed improves the performance. However it does not perform better than the Mirrored Wasserstein loss. We present results on 3 OOD detection cases in Table 5.

**Interpolation ablations:** We also explore how applying Berthelot interpolation Berthelot et al. [2018] in our model impacts the performance. We observe that the interpolation proposed by us performs better than the Berthelot interpolation. We present the results on 3 datasets in Table 6

**How do the synthetic anomalies look like?** Because of curse of dimensionality, the space spanned by all possible negatives (or anomalies) is huge and sampling from this space is a non-trivial problem. Hence, we use Atypical Se-

[1]http://odds.cs.stonybrook.edu/arrhythmia-dataset/

[2]http://kdd.ics.uci.edu/databases/kddcup99

Table 4: Performance of AMA and a few baselines on tabular datasets. The performance is measured using AUROC scores, higher the better.

| Model | Arrhythmia | KDD99 |
|---|---|---|
| Ano-GAN | 0.59 | 0.89 |
| ALAD | 0.74 | 0.94 |
| AMA (Ours) | **0.83** | **0.99** |

Table 5: AMA's performance with different losses. The performance is measured using AUROC scores, higher the better.

| Dataset | Wass. loss | Wass. loss + $L_2$ | Mirrored W. loss |
|---|---|---|---|
| FashionMNIST *vs* MNIST | 0.653 | 0.921 | **0.987** |
| CIFAR-10 *vs* SVHN | 0.8 | 0.87 | **0.958** |
| SVHN *vs* CIFAR-10 | 0.503 | 0.819 | **0.993** |

lection to sample synthetic anomalies. We choose the points lying on the line passing a random normal sample and origin. Since we are sampling in latent space, for Fig. 2, we use the generator to visualize them in image space. In reality these synthetic anomalies look like low-contrast version of normal samples. This is due to the fact that they are lying on the line connecting a normal sample and origin. It would be extremely hard to chance upon the latent representation of other datasets in such high dimensional spaces.

**Convergence analysis** In Fig. 3, we present the case of OOD detection in CIFAR-10 case and show how the AUROC scores converge with progressing number of epochs. We show 4 cases, (1) AMA with Mirrored Wasserstein loss and Latent space regularization aspects (2) AMA without Mirrored Wasserstein loss (3) AMA without simplex interpolation of normal samples in latent space (4) AMA without Atypical selection. In most of the cases, model converges by 30th epoch except for case 2. It seems Mirrored Loss is important for quick convergence and better AUROC scores.

**Corruption in the training data** One of the primary assumptions we made in our problem setup is, the training data consists of only normal samples. It is not a very realistic assumption since anomalies get stowed away as normal samples sometimes. We checked how our model performance gets affected when a few anomalies masked as normals (corruptions) are mixed in during the training time. We analysed the case where anomalies arise from the same manifold (same dataset) since it is a harder problem compared to OOD anomalies from different dataset. We present the results of experiments performed on CIFAR-10 dataset at different levels of corruptions in Fig. 4. Following the same setting from the main paper, in each of the experiments, one class is considered normal and rest of the 9 classes are considered anomalous. Let's consider case of 'dog' as normal class with 5% corruption, in this experiment, in training we will have 5000 images of 'dog' class with 250 images randomly sampled from rest of the 9 classes and are marked normal. As the level of corruption increases, intuitively the AUC scores are dropping, but the performance drop is not quite drastic, between 0 and 10% corruption, the average AUC drops by 4.6% in absolute. But anomalies, by definition should be sparse, so 10% corruption model can be considered as the lower bound of our model's performance.

Table 6: Performance of AMA with different types of interpolations on OOD detection case. The performance is measured using AUROC scores, higher the better.

| Dataset | AMA w. Berthelot | AMA |
|---|---|---|
| FashionMNIST *vs* MNIST | 0.93 | **0.99** |
| CIFAR-10 *vs* SVHN | 0.87 | **0.96** |
| SVHN *vs* CIFAR-10 | 0.92 | **0.99** |

Figure 2: **Generations of synthetic anomalies** sampled by Atypical selection. Since the sampling is done on the line connecting the origin and a random normal sample, some of the synthetic anomalies look like low-contrast version of normal samples.

**Additional ablation results** In Fig. 5, we show ablation studies of the regularization components of AMA. In this figure we consider the case where anomalies arise from the same dataset (In this case CIFAR-10). For few cases, the contributions made by either of the components, Simplex Interpolation or Atypical Selection is quite significant.

**How do the reconstructions look like?** We show how our model is reconstructing the samples in Figures 6, 7 and 8. It is interesting that our model does not reconstruct the anomalies well in some experiments, so while using our model, it is possible to qualitatively tell apart which images are normal and which ones are anomalies just by comparing an image to its reconstruction.

**How do the interpolations look like?** In Figures 9 and 10, we show how the interpolated latent representations look

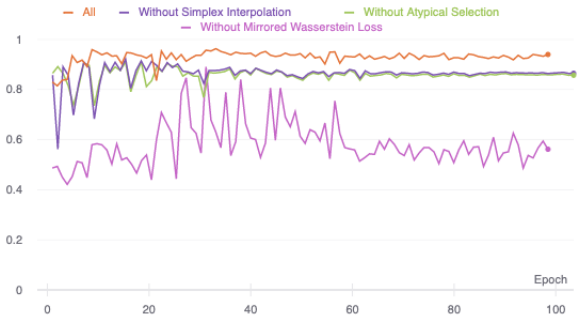



Figure 3: **Convergence analysis**: In this figure, we show how the validation AUROC scores change with respect to training epochs for AMA model trained on CIFAR-10 dataset. We show the trend in all 4 ablation cases we discussed in the main paper. Intuitively, Mirrored Wasserstein Loss plays an important role in convergence.

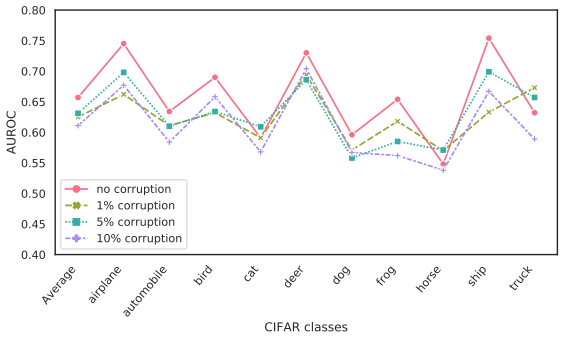



Figure 4: **Comparative analysis** of the case when corruptions (anomalies masked as normals) are introduced in training for CIFAR-10. Each experiment considers the class(on X-axis) as normal data and rest 9 as anomalous data. Performance is measured as AUC scores, higher the better. Despite corruption, performance drop is not too high. All the experiments are conducted with same hyper-parameters.

like compared to interpolations in image space. It is very visible in FashionMNIST case, the latent interpolations are generating realistic images as we interpolate between two points in test data set. For the sake of brevity we do not show the plots for the model trained on SVHN as the interpolation performance is similar to CIFAR-10.

**Misclassified** In Figures 11, 12 and 13, we show the misclassified images from both normal and anomaly data distribution. The False Negatives from both of the experiments seem to have sharp changes across the pixels, like dotted pattern frogs in CIFAR10 case, and clothes with patterns in FashionMNIST case. While in case of False positives in FashionMNIST vs MNIST case are mostly 1's and 7's which resemble 'pants' category of FashionMNIST data. One interesting observation is False Negatives in FashionMNIST dataset are very similar to the low likelihood ratio images suggested by LLR method Ren et al. [2019], which

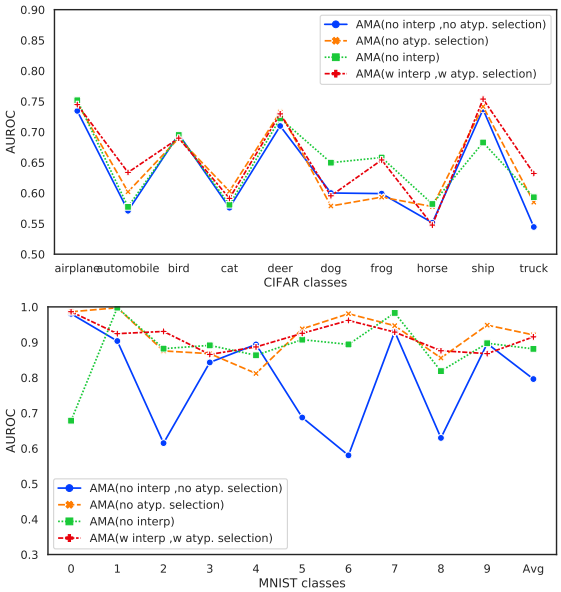



Figure 5: **Ablation analysis** on the use of simplex interpolation and atypical selection. The plot shows anomaly detection when one class of the respective dataset is used as normal class, while the other 9 classes are used as anomalies. We observe that both simplex interpolation and atypical selection improve performance, and using both components in combination yields the best results. (Top) Experiments on CIFAR-10 dataset (Bottom) Experiments on MNIST dataset. The performance measures are AUROC scores, higher the better.

suggests that our anomaly metric might be related to LLR metric.

## E SEMI-SUPERVISED TRAINING

One of the main advantages of our model is that it can be easily extended to semi-supervised scenario. What we mean by semi-supervised setting here is, a few tagged anomalies are available to us during train time along with many normal samples. For the objective from Equation (2) in the main paper and use real anomalies instead of synthetic anomalies chosen by Atypical Selection.

In Table 7, we consider the case of Anomaly Detection where anomalies arise within the dataset, which is CIFAR-10. We present the results of various cases of training setting, for varying availability of true anomlies. The Anom % column shows the ratio of *true* anomalies available during training time to the number of normal samples available in training. And each column represents the case where that particular class of CIFAR-10 is considered normal and rest of the 9 classes are considered anomalous. For eg, in 1% anomalies in training for dog experiment has - 5000 dog samples (normal), and 50 images randomly chosen for rest

| Anom % | airplane | automobile | bird | cat | deer | dog | frog | horse | ship | truck | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.752 | 0.634 | 0.696 | 0.603 | 0.733 | 0.650 | 0.658 | 0.582 | 0.754 | 0.632 | 0.669 |
| 0.1 | 0.776 | 0.582 | 0.699 | 0.556 | 0.743 | 0.599 | 0.723 | 0.553 | 0.781 | 0.473 | 0.648 |
| 1 | 0.809 | 0.732 | 0.729 | 0.613 | 0.749 | 0.667 | 0.759 | 0.716 | 0.834 | 0.665 | 0.727 |
| 5 | 0.847 | 0.814 | 0.744 | 0.747 | 0.788 | 0.741 | 0.827 | 0.784 | 0.873 | 0.798 | 0.796 |
| 10 | 0.870 | 0.892 | 0.775 | 0.760 | 0.823 | 0.810 | 0.855 | 0.851 | 0.887 | 0.831 | 0.835 |
| 20 | 0.870 | 0.910 | 0.814 | 0.785 | 0.813 | 0.832 | 0.859 | 0.879 | 0.895 | 0.867 | 0.852 |

Table 7: Semi-supervised Anomaly Detection performance on CIFAR-10 dataset. Each column denotes the normal class and the rest 9 classes from CIFAR-10 are considered as anomalies. Each row represents the case where a given percentage of *true* anomalies are available during training. Only for 0% case, we sample *synthetic anomalies* using atypical selection and use them in training. Performance reported are AUC scores. (Higher the better)

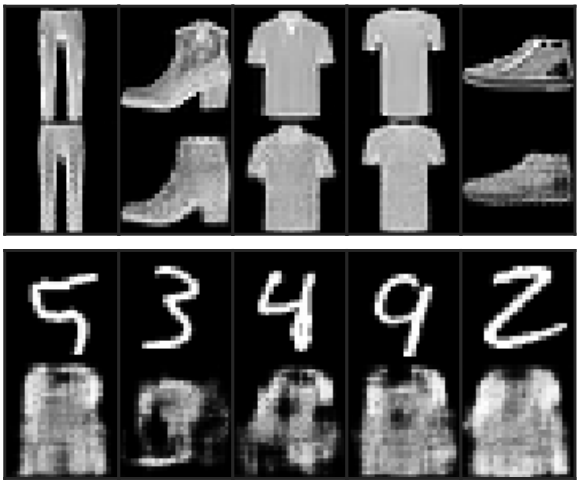


Figure 6: **AMA trained on Fashion MNIST data**: Here we show the performance of the model on test time Fashion MNIST images vs anomalies coming from MNIST dataset. Odd rows represent ground truths, and the even rows show the reconstructions. In this case, we can recognize anomalies not only quantitatively, but also qualitatively based on the reconstructions.

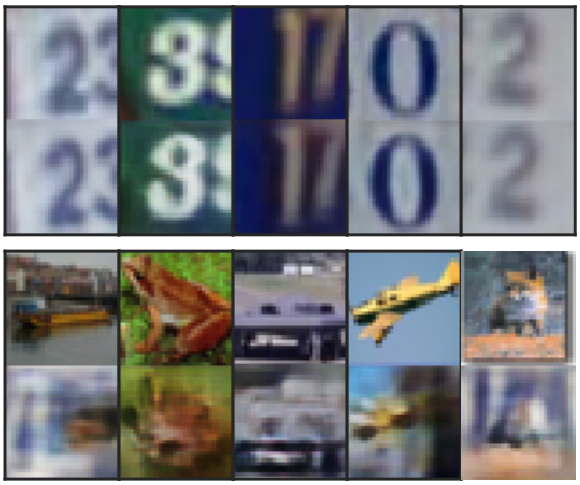


Figure 7: **AMA trained on SVHN data**: Here we show the performance of the model on test time SVHN images vs anomalies coming from CIFAR-10 dataset. Odd rows represent ground truths, and the even rows show the reconstructions. Similar to 6, the model performs well on normal test time samples, but reconstructs anomalies badly.

of the 9 classes (anomalies) are available during the training time. For 0% anomalies during training, we take the default setting of AMA and generate *synthetic anomalies* and use them during training. We can infer from the table, intuitively, as the percentage of anomalies available during training increases, the model performance increases.

One interesting case is when we have 0.1% of real anomalies available during the training. While in few classes like airplane;ship, using real anomalies certainly improves the performance compared to 0% real anomalies case, it did not improve in rest of the classes. This could be attributed to the extremely small size of real anomalies set (5 in this case). Hence in scenarios where very few real anomalies are available, we can improve the model by coupling real anomalies with some atypical sampled anomalies.

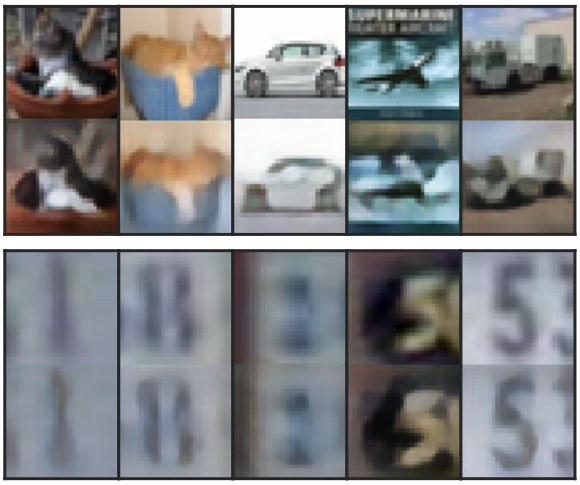


Figure 8: **AMA trained on CIFAR-10**: Here we show the performance of the model on test time CIFAR-10 images vs anomalies coming from SVHN dataset. Odd rows represent ground truths, and the even rows show the reconstructions. In this case, we cannot qualitatively distinguish anomalies from normal images. In such scenarios, we have to rely on quantitative A-scores as mentioned in the main text.

## References


David Berthelot, Colin Raffel, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *arXiv preprint arXiv:1807.07543*, 2018.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *NeurIPS*, pages 5767–5777, 2017.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *NeurIPS*, pages 14707–14718, 2019.

John Sipple. Interpretable, multidimensional, multimodal anomaly detection with negative sampling for detection of device failure. *arXiv preprint arXiv:2007.10088*, 2020.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

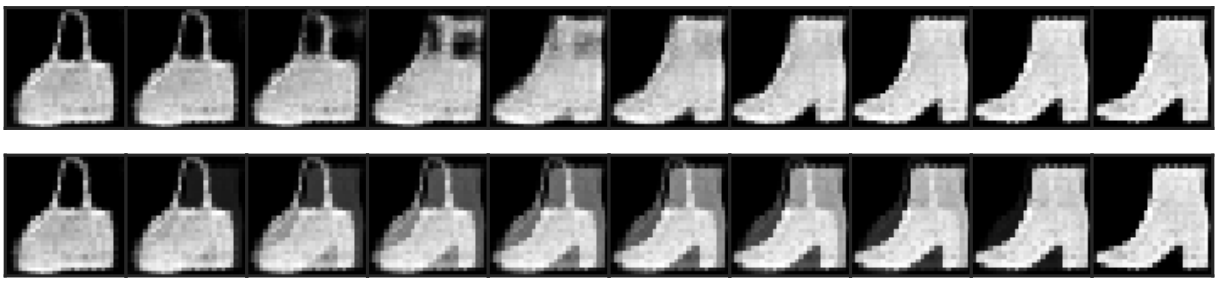


Figure 9: **Interpolations** of FashionMNIST images in latent and image spaces. Top panel shows how the reconstructions of interpolations in latent space. Bottom panel shows results when images are interpolated in image space. Clearly the latent interpolations look much better. The first and last images are from FashionMNIST dataset in each row.

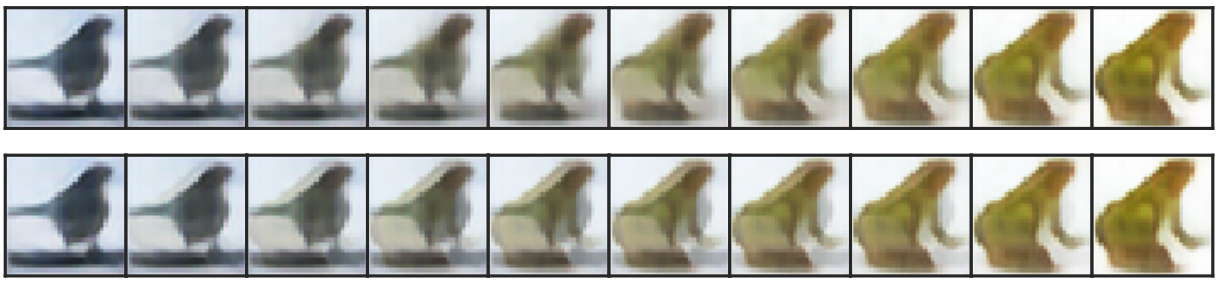
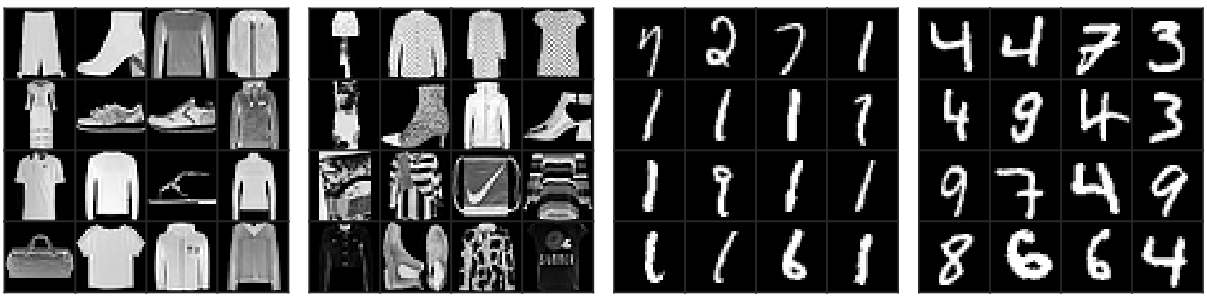
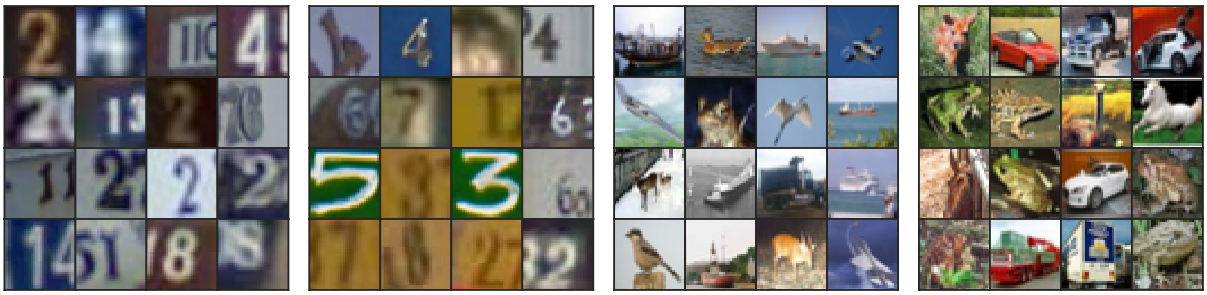


Figure 10: **Interpolations** of CIFAR10 images. Similar to Fig. 9, the top panels shows interpolation in latent space, while bottom panels shows interpolation in image space.



(a) True Positives (b) False Negatives (c) False Positives (d) True Negatives

Figure 11: We show here the plots with highest and lowest A-scores, $A(x)$ (Main paper - equation 6) from both FashionMNIST (normal) and MNIST (anomaly) classes. (a) True positives are from Normal class and have high $A(x)$ scores (b) False Negatives are from Normal class and have low $A(x)$ scores (c) False Positives are from Anomaly class and have high $A(x)$ scores. (d) True Negatives are from Anomaly class and have low $A(x)$ scores.



(a) True Positives (b) False Negatives (c) False Positives (d) True Negatives

Figure 12: Similar to Fig. 11, here we show the plots with highest and lowest $A(x)$ (Main paper - equation 6) from both SVHN (normal) and CIFAR-10 (anomaly) classes. (a) True positives are from Normal class and have high $A(x)$ scores (b) False Negatives are from Normal class and have low $A(x)$ scores (c) False Positives are from Anomaly class and have high $A(x)$ scores. (d) True Negatives are from Anomaly class and have low $A(x)$ scores.

(a) True Positives (b) False Negatives (c) False Positives (d) True Negatives

Figure 13: Similar to Fig. 11, here we show the plots with highest and lowest $A(x)$ (Main paper - equation 6) from both CIFAR10 (normal) and SVHN (anomaly) classes. (a) True positives are from Normal class and have high $A(x)$ scores (b) False Negatives are from Normal class and have low $A(x)$ scores (c) False Positives are from Anomaly class and have high $A(x)$ scores. (d) True Negatives are from Anomaly class and have low $A(x)$ scores.