AN ALGORITHM FOR OUT-OF-DISTRIBUTION ATTACK TO NEURAL NETWORK ENCODER

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural network (DNN), especially convolutional neural network, has achieved superior performance on image classification tasks. However, such performance is only guaranteed if the input to a trained model is similar to the training samples, i.e., the input follows the probability distribution of the training set. Out-Of-Distribution (OOD) samples do not follow the distribution of training set, and therefore the predicted class labels on OOD samples become meaningless. Classification-based methods have been proposed for OOD detection; however, in this study we show that this type of method has no theoretical guarantee and is practically breakable because of dimensionality reduction in the DNN model. We also show that Glow likelihood-based OOD detection is ineffective as well. Our analysis is demonstrated on five open datasets, including a COVID-19 CT dataset. At last, we present a simple theoretical solution with guaranteed performance for OOD detection.

1 INTRODUCTION

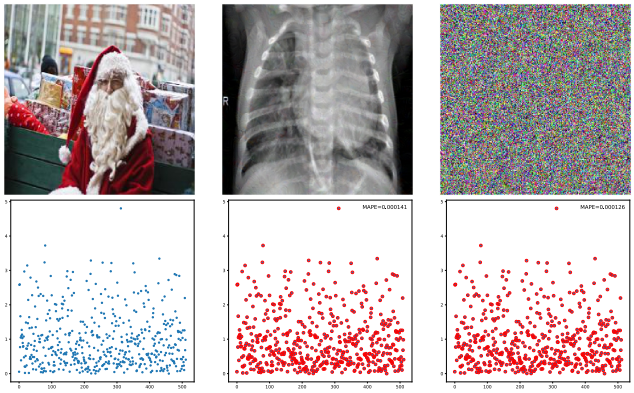
Deep neural network (DNN), especially convolutional neural network (CNN), has become the dominant technique for image classification. Under the i.i.d. (independent and identically distributed) assumption, a high-performance DNN model can correctly-classify an input sample as long as the sample is "generated" from the distribution of training data. If an input sample is not from this distribution, which is called Out-Of-Distribution (OOD), then the predicted class label from the model is meaningless. It would be great if the model has the ability to distinguish OOD samples from in-distribution samples. OOD detection is needed especially when applying DNN models in life-critical applications, e.g., vision-based self-driving or image-based medical diagnosis.

It was shown by Nguyen et al. in 2015 (Nguyen et al., 2015) that DNN classifiers can be easily fooled by OOD data, and evolutionary algorithm was used to generate OOD samples such that DNN classifiers had high output confidence on these samples. Since then, many methods are proposed for OOD detection using classifiers or encoders (Hendrycks & Gimpel, 2016)(Hendrycks et al., 2018)(Liang et al., 2017) (Lee et al., 2018) (Lee et al., 2017) (Alemi et al., 2018). For instance, Hendrycks et al. (Hendrycks & Gimpel, 2016) shows that a classifier's prediction probability of OOD examples tends to be lower than the prediction probability of in-distribution samples, and therefore the maximum predicted class probability from the softmax layer was used for OOD detection. Regardless of the details of these methods, every method needs a classifier or an encoder, which takes an image x as input and compresses it into a vector z in the laten space; after some further transform, z is converted to an OOD detection score τ. This computing process can be expressed as: $z = f(x)$ and $\tau = d(z)$. To perform OOD detection, a detection threshold needs to be specified, and then x is OOD if τ is smaller/larger than the threshold. For OOD detection method evaluation (Hendrycks & Gimpel, 2016), usually, an OOD detector is trained on a dataset (e.g. Fashion-MNIST as in-distribution) and then it is tested on another dataset (e.g. MNIST as OOD).

As will be shown in this study, the above mentioned classification-based OOD detection is theoretically almost ineffective and practically breakable. As an example (more details in section 3), we used the Resnet-18 model (He et al., 2016) pre-trained on ImageNet dataset. Let x_{in} denote a 224×224×3 image (in-distribution sample) in ImageNet and x_{out} denote an OOD sample which could be any kind of images (even random noises) not belonging to any category in ImageNet. Let z denote the 512-dimensional feature vector in Resnet-18, which is the input to the last fully-connected linear

layer before softmax operation. Thus, we have $z_{in} = f(x_{in})$ and $z_{out} = f(x_{out})$. As shown in Fig. 1, x_{in} is the image of Santa Claus, and x_{out} could be a chest x-ray image or a random-noise image, and "surprisingly", $z_{out} \cong z_{in}$ which renders OOD detection score to be useless: $d(z_{out}) \cong d(z_{in})$.

In section 2, we will introduce an algorithm to generate OOD samples such that $z_{out} \cong z_{in}$. In section 3, we will show the evaluation results on publicly available datasets, including ImageNet subset, GTSRB, OCT, and COVID-19 CT. Since some generative models (e.g. Glow (Kingma &



(a) (b) (c)

Figure 1: The 1st column shows the image of Santa Claus x_{in} and the scatter plot of z_{in} using blue dots. The 2nd column shows a chest x-ray image x_{out} and the scatter plot of z_{out} (red circles) and z_{in} (blue). The 3rd column shows a random image x_{out}, and the scatter plot of z_{out} (red) and z_{in} (blue).

Dhariwal, 2018)) can approximate the distribution of training samples (i.e. $p(x_{in})$), likelihood-based generative models were utilized for OOD detection (Nalisnick et al., 2018). It has been shown that likelihoods derived from generative models may not distinguish between OOD and training samples (Nalisnick et al., 2018) (Ren et al., 2019) (Choi et al., 2018), and the fix to the problem could be using likelihood ratio instead of raw likelihood score (Serrà et al., 2019). Although not the main focus of this study, we will show that OOD sample's likelihood score from the Glow model (Kingma & Dhariwal, 2018) (Serrà et al., 2019) can be arbitrarily manipulated by our algorithm (section 2.1) such that the output probability $p(x_{in}) \cong p(x_{out})$, which further diminishes the effectiveness of any Glow likelihood-based detection methods.

2 METHODOLOGY

2.1 OOD ATTACK ON DNN ENCODER

In this section, we will introduce an algorithm to perform OOD attack on DNN encoder $z = f(x)$ which takes an image x as input and transforms it into a feature vector z in a latent space. Preprocessing on x can be considered as the very first layer inside of the model $f(x)$. The algorithm needs a weak assumption that $f(x)$ is differentiable. A CNN classifier can be considered a composition of a feature encoder $z = f(x)$ and a feature classifier $p = g(z)$ where p is the softmax probability distribution over multiple classes.

Let's consider an in-distribution sample x_{in} and an OOD sample x'_{out}, and apply the model: $z_{in} = f(x_{in})$ and $z'_{out} = f(x'_{out})$. Usually, $z'_{out} \neq z_{in}$. However, if we add a relatively small amount of noise δ to x'_{out}, then it could be possible that $f(x'_{out} + \delta) = z_{in}$ and $x'_{out} + \delta$ is still OOD. This idea is realized in Algorithm 1, OOD attack on DNN Encoder.

The clip operation in the above algorithm is very important: it can ensure that x_{out} is OOD after a small modification to x'_{out}. The algorithm is inspired by the method called projected gradient descent (PGD) (Kurakin et al., 2016) (Madry et al., 2017) which is used for adversarial attack. We note that the term "adversarial attack" and "adversarial noise" usually refer to adding a small perturbation to a clean sample x in a dataset such that a classifier will incorrectly-classify the noisy

Algorithm 1 OOD attack on DNN Encoder

Input: An in-distribution sample x_{in} in a dataset. An OOD sample x'_{out} not similar to anyone in the dataset. f, the neural network feature encoder. ϵ, the maximum perturbation measured by Lp norm. N, the total number of iterations. α the learning rate of optimizer.

Output: an OOD sample x_{out} s.t. $f(x_{out}) \cong f(x_{in})$

Process:

1: Generate a random noise ξ with $||\xi|| \leq \epsilon$
2: Initialize $x_{out} = x'_{out} + \xi$
3: Setup loss $J(x_{out}) = ||f(x_{out}) - f(x_{in})||^2$ (L2 norm)
4: **for** n from 1 to N **do**
5: $\quad x_{out} \leftarrow clip(x_{out} - \alpha \cdot h(J'(x_{out})))$
6: **end for**

Note: The clip operation ensures that $||x_{out} - x'_{out}||_p \leq \epsilon$. The clip operation also ensures that pixel values stay within the feasible range (e.g. 0 to 1). If L-inf norm is used, $h(J')$ is the sign function; and if L2 norm is used, $h(J')$ normalizes J' by its L2 norm. Adamax optimizer is used in the implementation

sample while being able to correctly-classify the original clean sample x. Thus, OOD attack and adversarial attack are completely different things.

In practice, the Algorithm 1 can repeat many times to find the best solution. Random initialization is performed in line-1 and line-2 of the algorithm process. By adding initial random noise ξ to x'_{out}, the algorithm will have a better chance to avoid local minimum caused by a bad initialization.

2.2 Dimensionality Reduction and OOD Attack

Recall that in a classification-based OOD Detection approach, a DNN encoder transforms the input to a feature vector, i.e., $z = f(x)$, and an OOD detection score is computed by another transform on z, i.e., and $\tau = d(z)$. If $z_{out} \cong z_{in}$, then $d(z_{out}) \cong d(z_{in})$ which breaks the OOD detector regardless of the transform d. Usually, a DNN feature encoder makes dimensionality reduction: the dimension of z is significantly smaller than the dimension of x. In the example shown in Fig. 1, z is a 512-dimensional feature vector $(dim(z) = 512)$ in Resnet-18, and the dimension of x is 150528 $(224 \times 224 \times 3)$.

Dimensionality reduction in an encoder provides the opportunity for the existence of the mapping of OOD and in-distribution samples to the same locations in the latent space. This simply because the vectors in a lower-dimensional space cannot represent all of the vectors/objects in a higher-dimensional space, which is the Pigeonhole Principle. Let's do an analysis on the Resnet-18 example in Fig. 1. A pixel of the color image x has 8-bits. In the 150528-dimension discrete input space, there are $8^{224\times224\times3}$ different images/vectors, which defines the size of the input space. float32 data type is usually used in computation, a float32 variable can roughly represent 2^{32} unique real numbers. Thus, in the 512-dimensional latent space, there are $2^{32\times512}$ unique vectors/objects, which defines the size of the latent space. The ratio is $\left(\frac{2^{32\times512}}{8^{224\times224\times3}}\right) \ll 1$, and it shows that the latent space is significantly smaller than the input space. Thus, for some sample x in the dataset, we can find another sample x' such that $f(x') = f(x)$ as long as $dim(z) < dim(x)$. A question arises: will the x' be in-distribution or OOD? To answer this question, let's partition the input discrete space Ω into two disjoint regions ($\Omega = \Omega_{in} \cup \Omega_{out}$), Ω_{in} of in-distribution samples and Ω_{out} of OOD samples. $|\Omega|$ denotes the size of Ω. Usually, training set is only a subset of Ω_{in}, and the size of Ω_{out} is significantly larger than the size of Ω_{in}. For example, if Ω_{in} is ImageNet, then Ω_{out} contains medical images, noise images, and other weird images. If Ω_{in} contains human face images, then Ω_{out} contains non-face images and then $|\Omega_{in}| \ll |\Omega_{out}|$. The latent space (z-space) is denoted by $\mathcal{F}$ and partitioned into two subspaces: $\mathcal{F} = \mathcal{F}_{in} \cup \mathcal{F}_{out}$. An encoder is applied such that $\Omega_{in} \rightarrow \mathcal{F}_{in}$ and $\Omega_{out} \rightarrow \mathcal{F}_{out}$. If there is overlap $\mathcal{F}_{in} \cap \mathcal{F}_{out} \neq \emptyset$, then the encoder is **vulnerable** to OOD attack. Usually, the encoder is a part of a classifier trained to classify in-distribution samples into different classes, and therefore the encoder **cannot guarantee that** there is no overlap between $\mathcal{F}_{in}$ and $\mathcal{F}_{out}$. What is the size of $\mathcal{F}_{in} \cap \mathcal{F}_{out}$ or what is the probability $P(|\mathcal{F}_{in} \cap \mathcal{F}_{out}| \geq a)$? While it is hard to calculate it for an arbitrary encoder and dataset, we can do a worst-case-scenario analysis.

The intuition is that if $|\Omega_{out}| \gg |\mathcal{F}|$, then it is highly possible that the entire latent space is crawling with the shadows of OOD samples. Assuming that every OOD sample is i.i.d. mapped to the latent space with a uniform distribution over a number of $|\mathcal{F}|$ spots, then the probability of OOD samples covering the entire latent space is $P(\mathcal{F}_{out} = \mathcal{F}) = |\mathcal{F}|! \times Stirling(|\Omega_{out}|, |\mathcal{F}|) / |\mathcal{F}|^{|\Omega_{out}|} \rightarrow 1$ as $|\mathcal{F}| / |\Omega_{out}| \rightarrow 0$, where $Stirling$ is the Stirling number of the second kind. Noting that $|\mathcal{F}| / |\Omega_{out}| = \frac{2^{32\times512}}{8^{224\times224\times3} - 1.4\times10^7} \approx 0$ and 1.4×10^7 being the number of samples in ImageNet, then it could be true that almost (with probability close to 1) the entire latent space of Resnet-18 is covered by the z vectors of OOD samples.

Next, we discuss how to construct OOD samples to fool neural networks. First, let's take a look at one-layer linear network: $z = Wx$, and make notations: an in-distribution input $x \in \mathcal{R}^M$, laten code $z \in \mathcal{R}^K$ and $K \ll M$. W is a $K \times M$ matrix, and $rank(W) \leq K$. The null space of W is $\Omega_{null} = \{\eta;\ W\eta = 0\}$. Now, let's take out the basis vectors of this space, η_1, η_2, ..., η_{M-K}, and compute $x' = \sum_i \lambda_i \eta_i + x$ where λ_i is a non-zero scalar. Obviously, $z' = Wx' = z$. We can set the magnitude of the "noise" $\sum_i \lambda_i \eta_i$ to be arbitrarily large such that x' will look like garbage and become OOD, which is another explanation of the existence of OOD samples. Then, we can try to apply this attack method to multi-layer neural network. If the neural network only uses ReLU activation, then the input-output relationship can be exactly expressed as a linear mapping (Ding et al., 2018), a similar approach can be applied layer by layer. If ReLU is not used, a new method is needed. We note that the filter bank of a convolution layer can be converted to a weight matrix. We have examined the state-of-art CNN models that are pre-trained on ImageNet and available in Pytorch, and dimensionality reduction is performed in most of the layers (except 1 or 2 layers near the input), i.e. $|\mathcal{F}| \leq |\Omega_{in}| \ll |\Omega_{out}|$. Instead of constructing an OOD sample by adding perturbations to an in-distribution sample, in Algorithm-1, we construct OOD samples paired with in-distribution samples by starting from a single initial sample that is OOD.

Could an encoder be made robust to the OOD attack by including OOD samples in training set for supervised binary classification: in vs out? Usually $|\Omega_{in}| \ll |\Omega_{out}|$ and we will have to collect and label "enough" samples in Ω_{out}, which is infeasible considering the large size of $\Omega_{out} \approx \Omega$. As a comparison, to enhance DNN classifier robustness against adversarial noises, it is very effective to include noisy samples in the training set, i.e. $\Omega_{in} = \Omega_{in_clean} \cup \Omega_{in_noisy}$. It is known as adversarial training (Goodfellow et al., 2018) and computationally feasible as $|\Omega_{in_noisy}| \ll |\Omega_{out}|$.

2.3 Problem of Glow likelihood-based OOD Detection

Generative models have been developed to approximate the training data distribution. Glow (Kingma & Dhariwal, 2018) is one of these models, and it has a very special property: it is bijective and the latent space dimension is the same as the input space dimension, i.e., no dimensionality reduction, which is the reason that we studied this model.

Several studies have found the problem of Glow-based OOD detection: likelihoods derived from Glow may not distinguish between OOD and training samples (Ren et al., 2019) (Choi et al., 2018), and a possible fix to the issue could be using likelihood ratio (Serrà et al., 2019). In this study, we further show that OOD sample's negative log-likelihood (NLL) from the Glow model can be arbitrarily manipulated by our algorithm in which $f(x)$ denotes NLL. The results on CelebA face image dataset are in Section 3. We think the major reason causing Glow's vulnerability to OOD attack is that we do not have enough training data in high dimensional space. Glow is a mapping: $x_{in} \rightarrow z_{in} \rightarrow p(z_{in}) \rightarrow p(x_{in})$, the probability of x_{in}. For an OOD sample x_{out}, the mapping is $x_{out} \rightarrow z_{out} \rightarrow p(z_{out}) \rightarrow p(x_{out})$. Since the number of training samples is significantly smaller than the size of the space, there are a huge number of "holes" in the space, and it is easy to put z_{out} in one of these "holes" close to z_{in} such that $p(z_{out}) \cong p(z_{in})$.

2.4 Reconstruction-based OOD Detection

Auto-encoder style OOD detection has been developed for anomaly detection (Chalapathy & Chawla, 2019)(Cohen et al., 2019) based on reconstruction error. The data flow of an auto-encoder is $x \rightarrow z \rightarrow \hat{x}$ where $\hat{x}$ is the reconstruction of x. The OOD detection score can be the difference between x and $\hat{x}$, e.g., the Lp distance $||x - \hat{x}||_p$ or Mahalanobis Distance. This type of method has two known issues. The first issue is that auto-encoder may well reconstruct OOD samples, i.e.,

$x_{out} \approx \hat{x}_{out}$. Thus, one needs to make sure it has large reconstruction errors on OOD samples, which can be done by limiting the capacity of auto-encoder or saturating it with in-distribution samples. The second issue is that pixel-to-pixel distance is not a good measurement of image dissimilarity, especially for medical images. For example, x could be a CT image of a heart and $\hat{x}$ could be the image of the same heart that deforms a little bit, but the pixel-to-pixel distance between x and $\hat{x}$ can be very large. Thus, a robust image similarity measurement is needed.

Interestingly, the proposed OOD attack algorithm has no effect on this type of method. Let's consider the data flow: $x_{in} \rightarrow z_{in} \rightarrow \hat{x}_{in}$ and $x_{out} \rightarrow z_{out} \rightarrow \hat{x}_{out}$. If $z_{out} = z_{in}$, then $\hat{x}_{out} = \hat{x}_{in}$. Then, it is easy to find out that x_{out} is OOD because $||x_{out} - \hat{x}_{out}||_p = ||x_{out} - \hat{x}_{in}||_p$ which is very large. Ironically, in this case, the attack algorithm helps to identify the OOD sample. In future work, we will evaluate the effectiveness of combining the proposed algorithm and auto-encoder for OOD detection.

3 EXPERIMENT

We applied the proposed algorithm to attack state-of-art DNN models on open image datasets. For each in-distribution sample x_{in} in our evaluation, an OOD sample x_{out} is generated by the algorithm. To measure attack strength, mean absolute percentage error is calculated by $MAPE(z_{out}) = mean(|z_{out} - z_{in}|)/max(|z_{in}|)$. Here, $z_{out} = f(x_{out})$ and $z_{in} = f(x_{in})$. $|z_{out} - z_{in}|$ is an error vector, and $mean(|z_{out} - z_{in}|)$ is the average error. $max(|z_{in}|)$ is the maximum absolute value in the vector z_{in}. We also applied the algorithm to attack the Glow model on CelebA dataset. In all of the evaluations, L2 norm was used in the proposed algorithm. Pytorch was used to implement the algorithm. Nvidia Titan V GPU was used for model training and testing.

3.1 EVALUATION ON A SUBSET OF IMAGENET

ILSVRC2012 ImageNet has over 1 million images in 1000 classes. Given the limited computing power, it is impractical to test the algorithm on the whole dataset. Instead, we used a subset of 1000 images in 200 categories. The size of each image is 224×224×3. Two CNN models pretrained on the ImageNet were evaluated, which are Resnet-18 and Densenet-121 available in Pytorch.

Resnet-18 latent space has 512 dimensions. Since ImageNet covers a variety of natural and artificial objects, we choose medical images and random-noise images to make sure that x'_{out} is indeed OOD. Using each of the three initial OOD samples (chest x-ray, lung-CT, and random noise to be x'_{out}), we generated 1000 OOD samples paired with the 1000 (in-distribution) samples in the dataset and calculated MAPE values. The three MAPE histograms are shown in Fig. 3. Most of the MAPE values are less than 0.1%.

We also evaluated another CNN, named Densenet-121, and obtained similar results. The latent space has 1024 dimensions. Again, using each of the three initial OOD samples, 1000 OOD samples are generated for the samples in the dataset, and then MAPE values are calculated. The three MAPE histograms are shown in Fig. 4. Most of the MAPE values are less than 0.1%, indicating strong OOD attack.

From the results in Fig.2 to Fig. 4, it can be seen that each of the two CNN models mapped significantly different OOD samples and in-distribution samples to the same locations in the latent space. Dimensionality reduction leads to the existence of such mapping, and our algorithm can find such OOD samples out. In other words, the mapping from input space to the latent space is many-to-one, not bijective. And therefore, it is almost guaranteed that such OOD samples exist and they can break any OOD detector d that computes a detection score $d(z)$ only from the latent space (z-space). Given the results of $z_{out} \cong z_{in}$ ensuring $d(z_{out}) \cong d(z_{in})$, it is unnecessary to re-implement and test OOD detectors based on $d(z)$. For illustration purpose, we tested a classical OOD detection method using the maximum of softmax output as detection score (Hendrycks & Gimpel, 2016). The results are shown in Table-1, and the AUROC scores are close to 0.5, showing that the method is unable to tell the difference between the 1000 OOD samples and 1000 in-distribution samples.

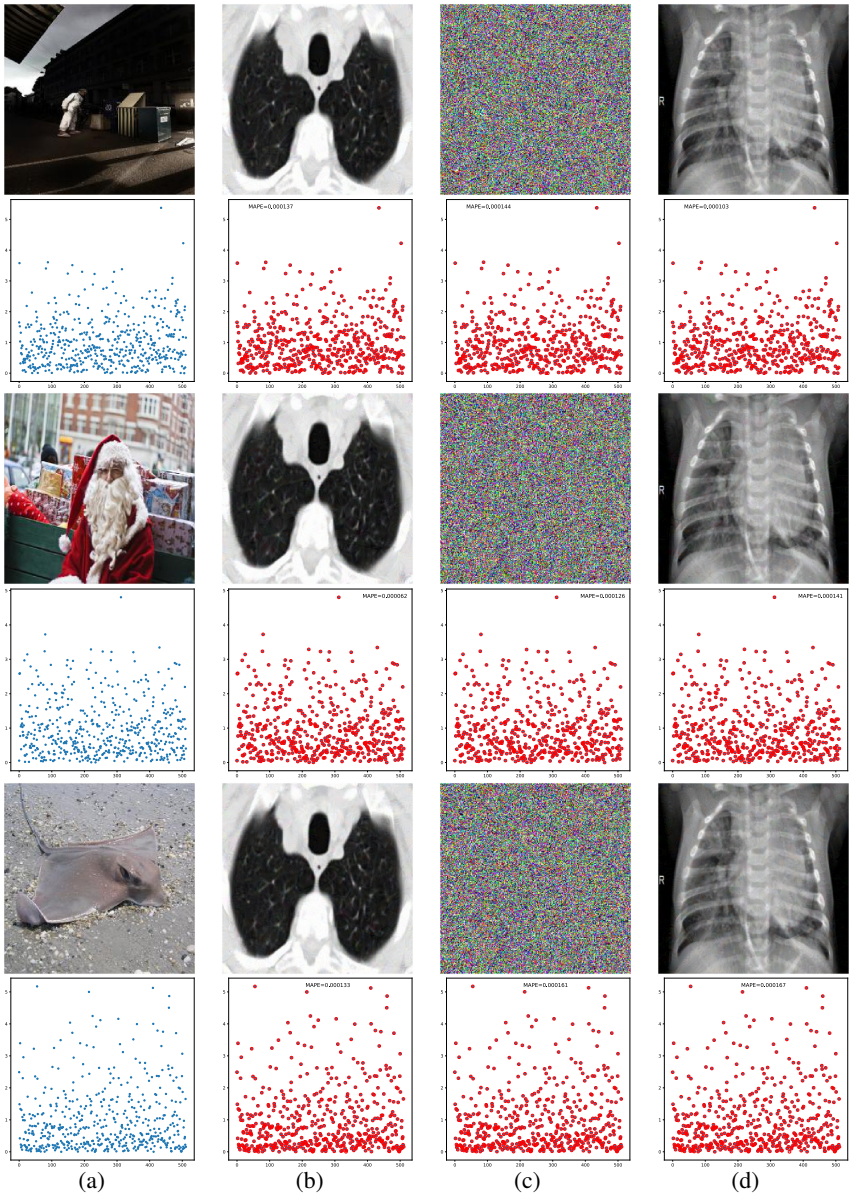


Figure 2: The 1st column shows three in-distribution samples (e.g. Santa Claus) x_{in}, and the corresponding scatter plots of z_{in} (blue dots). The 2nd column shows OOD samples x_{out} generated from a CT image x'_{out}, and the corresponding scatter plots of z_{out} (red) and z_{in} (blue). The 3rd column shows OOD samples x_{out} generated from a random image x'_{out}, and the corresponding scatter plots of z_{out} (red) and z_{in} (blue). The 4th column shows OOD samples x_{out} generated from a x-ray image x'_{out}, and the corresponding scatter plots of z_{out} (red) and z_{in} (blue). MAPE values are embedded in these scatter-plots. Please zoom-in for better visualization.

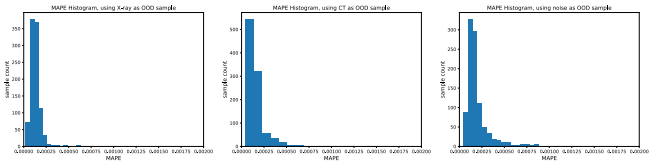


Figure 3: Left: the MAPE histogram using a chest x-ray as the initial OOD sample. Middle: the MAPE histogram using a lung CT image as the initial OOD sample. Right: the MAPE histogram using a random-noise image as the initial OOD sample. Please zoom-in for better visualization.

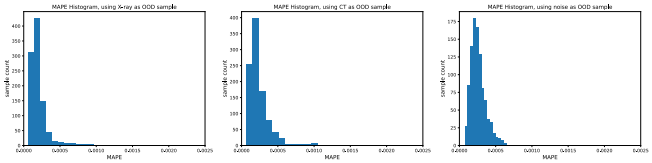


Figure 4: Left: MAPE histogram using a chest x-ray as the initial OOD sample. Middle: MAPE histogram using a lung CT image as the initial OOD sample. Right: MAPE histogram using a random-noise image as the initial OOD sample. Please zoom-in for better visualization.

Table 1: AUROC of two networks under OOD attack with each of x-ray, CT and noise as the initial OOD sample

	x-ray	**CT**	**random-noise**
Resnet-18	0.643	0.633	0.500
Densenet-121	0.638	0.651	0.500

3.2 Evaluation on CelebA Dataset

We tested the algorithm and the Glow model (Kingma & Dhariwal, 2018) on the CelebA dataset (human face images). The size of each image is 64×64×3. After training, the model was able to generate realistic face images. The model also outputs the negative log-likelihood (NLL) of the input sample, i.e., $NNL(x) = -log(p(x))$. By setting $f(x) = NNL(x)$, our algorithm can make $f(x_{out})$ to be close to 0 or very large to match any $f(x_{in})$, which renders NLL score useless for OOD detection. To demonstrate the effectiveness of our algorithm, we randomly selected 160 (in-distribution) samples in the dataset. We used a color spiral image as the initial OOD sample x'_{out}, and $NNL(x'_{out}) = 3.5268$. The distributions of $NLL(x_{in})$ from 160 in-distribution samples and $NLL(x_{out})$ from 160 corresponding OOD samples, as well as OOD sample images are shown in Fig. 5. The two distributions are almost identical. More examples of OOD samples are shown in Fig. 6. In each row of Fig. 6, although the images have different NLL scores, they look like each other.

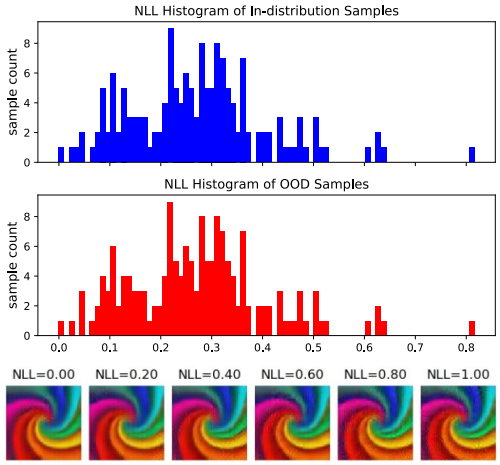


Figure 5: Top: NLL histogram (blue bars) of the in-distribution; samples Middle: NLL histogram (red bars) of OOD samples; Bottom: some OOD samples with NLL from 0 to 1. The initial OOD sample is a spiral image.

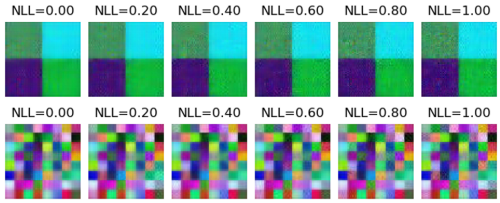


Figure 6: Top: OOD samples generated by using one 2×2 checkerbox image for initialization; Bottom: OOD samples generated by using one 8×8 checkerbox image for initialization.

Other three evaluations are shown in Appendix.

4 DICUSSION

For a DNN model that makes dimensionality reduction and therefore is not bijective, our algorithm can find OOD samples that will be mapped by the model to the locations of the in-distribution samples in the latent space, which invalidates any OOD detector that computes a detection score only from the latent space. We performed extensive tests to demonstrate the analysis and our algorithm on different datasets and applications.

The only assumption of the algorithm is that the model is differentiable. Non-differentiable preprocessing on the input cannot be incorporated into the model as the very first layer, and therefore it is ignored in this study. Such preprocessing has little effect on the OOD sample space and only makes it harder to find OOD samples. To handle such cases, our algorithm can be enhanced by using a sampling-based gradient estimator, as done in SPSA (Maryak & Chin, 2001)(Uesato et al., 2018).

We also show that the output NLL score of the generative model Glow is easily manipulated by the algorithm, and therefore Glow NLL score is not suitable for OOD detection. We note that there are other generative methods for OOD detection, which have shown promising results. Evaluation of these methods is out of the scope of this study.

We would like to point out that it is difficulty to evaluate an OOD detector to "prove" that it can detect, say 90% of the OOD samples by experimentally testing it on Ω_{out} because Ω_{out} is too large to be tested on: $|\Omega_{in}| \ll |\Omega_{out}| \approx |\Omega|$. For example, if Fashion-MNIST is used as in-distribution, then MINST and Omniglot are usually as OOD, which is the "standard" approach in the literature. Clearly, MINST and Omniglot cannot cover Ω_{out} the space of OOD samples. If the image size is larger, then $|\Omega_{out}|$ becomes much larger. Could we design an evaluation method (experimental or analytical) that does not rely on OOD samples?

At last, we present a simple method for OOD detection with theoretical guarantee: subspace saturation training of bijective DNN for OOD detection. The method does not need any OOD samples. In a bijective mapping, $x \rightarrow f(x) \rightarrow z \rightarrow f^{-1}(x) \rightarrow x$, the dimension of output z is equal to the dimension of input x, i.e., no dimensionality reduction. We "manually" partition the latent space (z-space) into two disjoint subspaces $\mathcal{F} = A \cup B$, $A \cap B = empty$, and then the network will be trained to map a huge number of in-distribution samples into the subspace A such that the subspace A is saturated by in-distribution samples (i.e. no 'holes'). Thus, for any OOD sample x_{out}, it is guaranteed that $z_{out} \in B$ because there is no empty spot in the subspace A and the mapping is bijective. As long as the subspace A is saturated by in-distribution samples, then the network can detect all of the OOD samples, which means to evaluate the performance of the network, we only need to check the saturation rate and do not need OOD samples. We provide two implementations of this method in the Appendix.

Before the OOD issue is fully resolved, for life-critical applications, any machine learning system that uses DNN classifiers should not make decisions independently and can only serve as assistants to humans.

We will release the code on GitHub when the paper is accepted. All figures are in high-resolution, please zoom in.

REFERENCES

Alexander A Alemi, Ian Fischer, and Joshua V Dillon. Uncertainty in the variational information bottleneck. *arXiv preprint arXiv:1807.00906*, 2018.

Alvaro Arcos-Garcia, Juan A Alvarez-Garcia, and Luis M Soria-Morillo. Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods. *Neural Networks*, 99:158–165, 2018.

Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International Conference on Machine Learning*, pp. 573–582, 2019.

Raghavendra Chalapathy and Sanjay Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

Hyunsun Choi, Eric Jang, and Alexander A Alemi. Waic, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018.

Joseph Paul Cohen, Paul Bertin, and Vincent Frappier. Chester: A web delivered locally computed chest x-ray disease prediction system. *arXiv preprint arXiv:1901.11210*, 2019.

Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Max-margin adversarial (mma) training: Direct input space margin maximization through adversarial training. *arXiv preprint arXiv:1812.02637*, 2018.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1625–1634, 2018.

Ian Goodfellow, Patrick McDaniel, and Nicolas Papernot. Making machine learning robust against adversarial inputs. *Communications of the ACM*, 61(7):56–66, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016.

Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606*, 2018.

Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *Cell*, 172(5):1122–1131, 2018.

Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in neural information processing systems*, pp. 10215–10224, 2018.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Kimin Lee, Honglak Lee, Kibok Lee, and Jinwoo Shin. Training confidence-calibrated classifiers for detecting out-of-distribution samples. *arXiv preprint arXiv:1711.09325*, 2017.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, pp. 7167–7177, 2018.

Shiyu Liang, Yixuan Li, and Rayadurgam Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. *arXiv preprint arXiv:1706.02690*, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

John L Maryak and Daniel C Chin. Global random optimization by simultaneous perturbation stochastic approximation. In *Proceedings of the 2001 American Control Conference.(Cat. No. 01CH37148)*, volume 2, pp. 756–762. IEEE, 2001.

Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*, 2018.

Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015.

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 *Advances in Neural Information Processing Systems*, pp. 14707–14718, 2019.

Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F Núñez, and Jordi Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. *arXiv preprint arXiv:1909.11480*, 2019.

Feng Shi, Jun Wang, Jun Shi, Ziyan Wu, Qian Wang, Zhenyu Tang, Kelei He, Yinghuan Shi, and Dinggang Shen. Review of artificial intelligence techniques in imaging data acquisition, segmentation and diagnosis for covid-19. *IEEE reviews in biomedical engineering*, 2020.

Eduardo Soares, Plamen Angelov, Sarah Biaso, Michele Higa Froes, and Daniel Kanda Abe. Sars-cov-2 ct-scan dataset: A large dataset of real patients ct scans for sars-cov-2 identification. *medRxiv*, 2020.

Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial risk and the dangers of evaluating against weak attacks. *arXiv preprint arXiv:1802.05666*, 2018.

Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, 2018.

A Appendix

If we think OOD detection is a binary classification problem, then we need to find a decision boundary in the input space, which can separate in-distribution samples from OOD samples. However, OOD samples are way too many, and therefore it is infeasible to do a supervised classification by collecting an "enough" number of OOD samples (how many is "enough"?). Instead of learning a decision boundary from data, we can fix/pre-define a decision boundary, which can be achieved by partitioning the latent space (z-space) into two disjoint subspaces, A and B. Then, we use a large number of in-distribution samples to saturate the subspace A. Given a bijective DNN, then each OOD sample has nowhere to go but to the subspace B. Here are two implementations of this idea.

Implementation-1 using an auto-encoder (encoder is bijective DNN, e.g., (Behrmann et al., 2019)): enforce $z_{in}[j] = 0 for\ j \geq J$, where $z_{in}[j]$ is the j-th element in the vector z_{in}, by using the loss: $L = \sum_{j=J}^{|F|} |z_{in}[j]| + ||x - \hat{x}||_p$. Thus, $z_{out}[j] \neq 0 for\ j \geq J$. OOD detection score is $\sum_{j=J}^{|F|} |z[j]|$. For example, z has $|\mathcal{F}| = 512$ elements and set $J = 10$. Assuming float32 precision, then at least a total number of $2^{32\times 10}$ samples are need for training. To generate so many samples, we could use a GAN or simply add noises to image data (which can also improve robustness against noises).

Implementation-2 using a GAN (generator is bijective DNN, e.g. Glow): it is easy to enforce $z_{in}[j] = 0 for\ j \geq J$ because the dataflow in a GAN is from z to x. OOD detection score is $\sum_{j=J}^{|F|} |z[j]|$.

The rationale is that we want to classify in-distribution samples and OOD samples but do not want to use any OOD samples (too many). As of today, we have not completed the testing of the two implementations because our humble research lab does not have huge computing power. Using current hardware in our lab, it may take several months to complete the experiments. Nevertheless, the method is implementable and has theoretical guarantee, and the only limit is computing power.

B Appendix

In this section, the parameters in each evaluation are shown.

1. Parameters for evaluation on ImageNet subset Attacking resnet-18: $\varepsilon = 5$, $N = 1e4$, $\alpha = \varepsilon/100$ with x-ray and CT; $\varepsilon = 20$, $N = 1e4$, $\alpha = \varepsilon/100$ with random noise. Attacking densnet-121: $\varepsilon = 5$, $N = 1e4$, $\alpha = \varepsilon/100$ with x-ray and CT; $\varepsilon = 30$, $N = 1e4$, $\alpha = \varepsilon/100$ with random noise. We inspected the OOD images from random noise: they are not recognizable to human vision.

2. Parameters for evaluation on OCT dataset $\varepsilon = 10$, $N = 1e4$, $\alpha = \varepsilon/100$ with retinal fundus photography image $\varepsilon = 20$, $N = 1e4$, $\alpha = \varepsilon/100$ with random noise.

3. Parameters for evaluation on COVID-19 CT dataset $\varepsilon = 20$, $N = 1e4$, $\alpha = \varepsilon/100$

4. Parameters for evaluation on GTSRB dataset $\varepsilon = 10$, $N = 1e4$, $\alpha = \varepsilon/100$

5. Parameters for Evaluation on CelebA Dataset $\varepsilon = 10$, $N = 1e4$, $\alpha = \varepsilon/100$

C APPENDIX

C.1 EVALUATION ON OCT DATASET

We tested our algorithm and Resnet-18 on a retinal optical coherence tomography (OCT) dataset (Kermany et al., 2018), which has four classes. Each image is resized to 224×224. 1000 samples per class were randomly selected to obtain a training set of 4000 samples. The test set has 968 images. We modified Resnet-18 for this four-class classification task. The latent space has 512 dimensions. After training, the Resnet-18 model achieved a classification accuracy $> 95\%$ on the test set.

We used two references images as the initial OOD sample x'_{out}. The first reference image is a grayscale retinal image converted from an RGB color retinal fundus photography image. Compared to this retinal fundus photography image, the OCT images have unique patterns of horizontal "white bands". We selected this OOD image by purpose: there may be a chance that both types of images are needed for retinal diagnosis. The second reference image is generated from random noises. Examples are shown in Fig. 7, and the two MAPE histograms are shown in Fig. 8. The results confirm that the algorithm can generate OOD samples (968) which are mapped by the DNN model to the locations of the in-distribution samples (968) in the latent space, i.e., $z_{out} \cong z_{in}$.

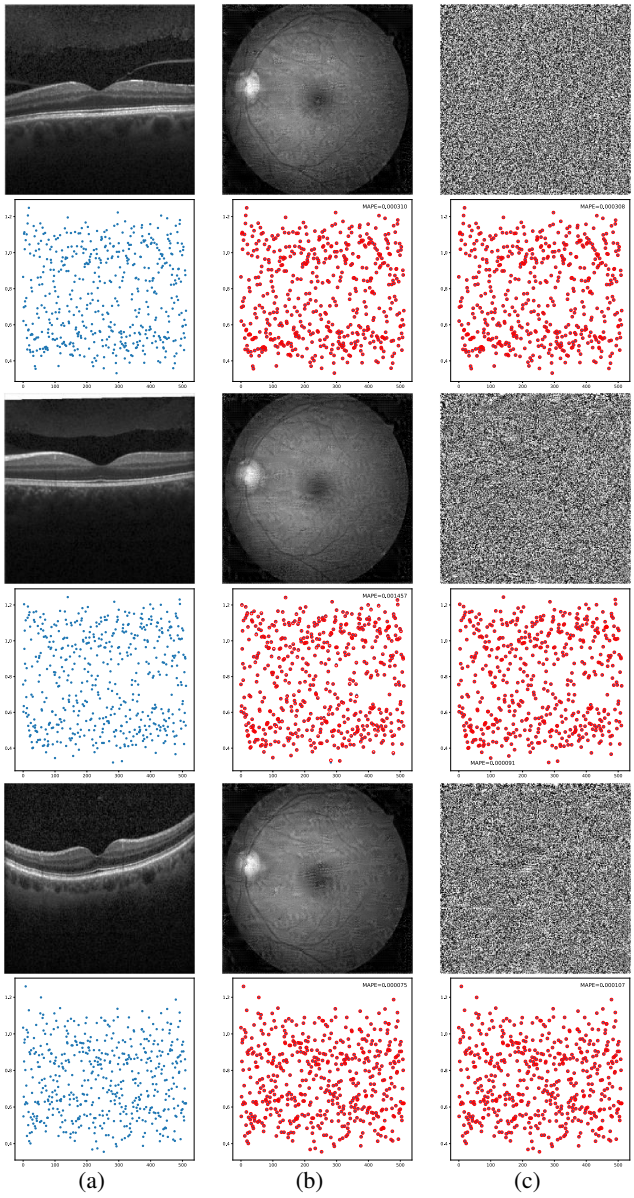


Figure 7: The 1st column shows in-distribution samples x_{in} in the OCT dataset, and the scatter plots of z_{in} (blue dots). The 2nd column shows OOD samples x_{out} generated from a retinal fundus photography image x'_{out}, and the scatter plots of z_{out} (red) and z_{in} (blue). The 3rd column shows OOD samples x_{out} generated from a random image x'_{out}, and the scatter plots of z_{out} (red) and z_{in} (blue). MAPE values are embedded in these scatter-plots. Please zoom-in for better visualization.

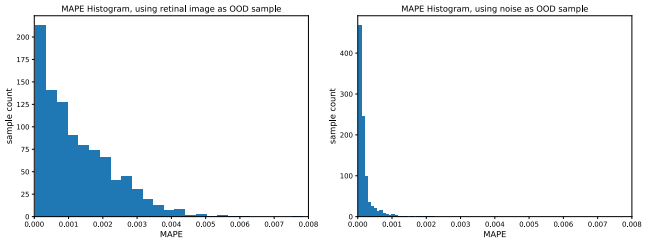


Figure 8: Left: MAPE histogram using a retinal fundus photography image as the initial OOD sample. Right: MAPE histogram using a random-noise image as the initial OOD sample. Please zoom-in for better visualization.

C.2 Evaluation on COVID-19 CT Dataset

We also tested our algorithm and Resnet-18 on a public COVID-19 lung CT (2D) image dataset (Soares et al., 2020). It contains 1252 CT scans (2D images) that are positive for COVID-19 infection and 1230 CT scans (2D images) for patients non-infected by COVID-19, 2482 CT scans in total. From infected cases, we randomly selected 200 samples for testing, 30 for validation, and 1022 for training. From the uninfected cases, we randomly selected 200 for testing, 30 for validation and 1000 for training. Each image is resized to 224×224.

We modified the last layer of Resnet-18 for this binary classification task, infected vs uninfected. We also replaced batch normalization with instance normalization because it is known that batch normalization is not stable for small batch-size (Wu & He, 2018). The latent space still has 512 dimensions. We set batch-size to 32, the number of training epochs to 100, and used AdamW optimizer with the default parameters. After training, the model achieved a classification accuracy $> 95\%$ on test set.

We used two reference images as the initial OOD sample x'_{out}, a chest x-ray image, and a random-noise image. The two MAPE histograms are shown in Fig. 9 that most of the MAPE values are less than 0.1%. The results also confirm that the algorithm can generate OOD samples (400) which are mapped by the DNN model to the locations of the in-distribution samples (400) in the latent space, i.e., $z_{out} \cong z_{in}$.

Examples are shown in Fig. 10. As reported in the previous studies (Shi et al., 2020), infected regions in the images have a unique pattern called ground-glass opacity. The CT images in the 1st and 3rd rows show COVID-19 infections with ground-glass opacity on the upper-left area. The CT image in the 5th row does not show any signs of infection. It can be seen that the random-noise images and the COVID-19 CT images have the same feature vectors in the latent space, which is astonishing.

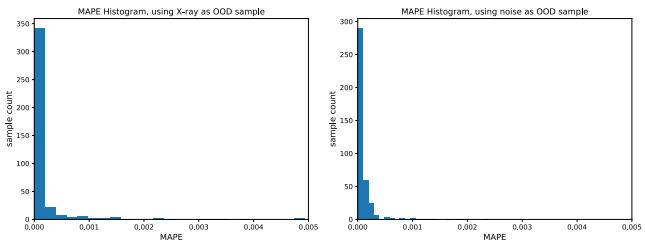


Figure 9: Left: MAPE histogram using chest x-ray image as the initial OOD sample. Right: MAPE histogram using a random-noise image as the initial OOD sample. Please zoom-in for better visualization.

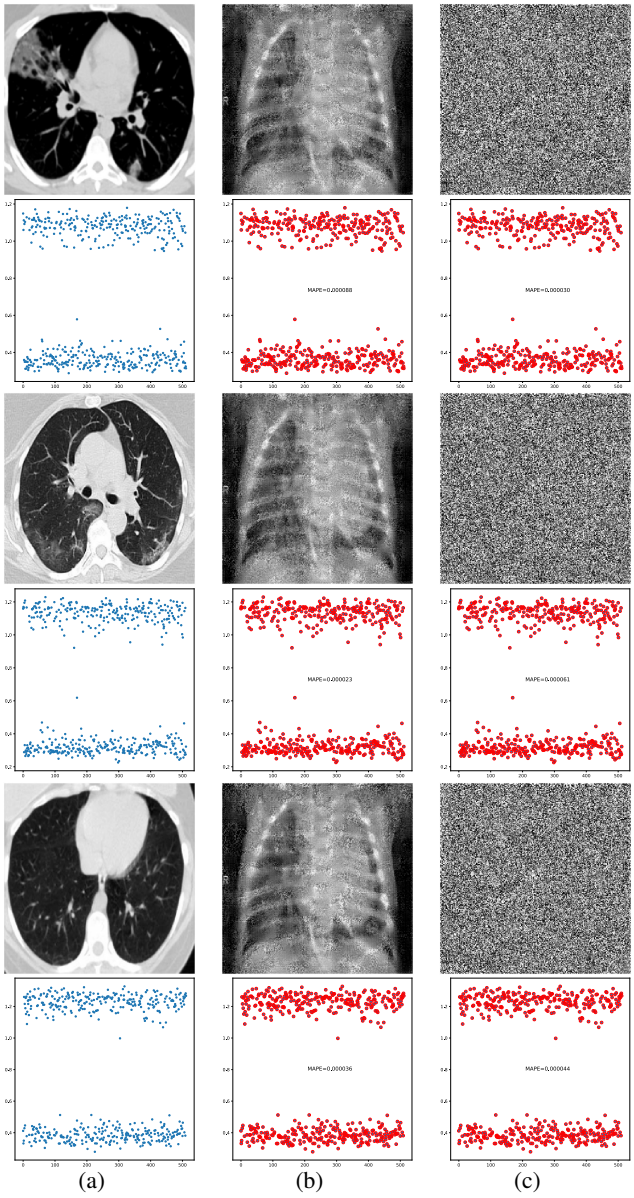


Figure 10: The 1st column shows in-distribution samples x_{in} in the COVID-19 dataset, and the scatter plots of z_{in} (blue dots). The 2nd column shows OOD samples x_{out} generated from a chest x-ray image x'_{out}, and the scatter plots of z_{out} (red) and z_{in} (blue). The 3rd column shows OOD samples x_{out} generated from a random image x'_{out}, and the scatter plots of z_{out} (red) and z_{in} (blue). MAPE values are embedded in these scatter-plots. Please zoom-in for better visualization.

C.3 Evaluation on GTSRB Traffic Sign Dataset

We tested our algorithm and a start-of-art traffic sign classifier on the GTSRB dataset. The classifier is similar to the one in (Arcos-Garcia et al., 2018), which has a spatial-transformer network. The size of each image is 32×32×3. The latent space has 128 dimensions. After training, the classifier achieved over 99% accuracy on the test set. We used a random-noise image as the initial OOD sample x'_{out} to generate 12630 OOD samples paired with the 12630 in-distribution samples in the test set. The MAPE histogram is shown in Fig. 11, in which most of the MAPE values are less than 0.1%. Examples are shown in Fig. 12.

It can be seen that z_{out} of random-noise images are almost the same as z_{in} of the stop sign, the speed limit sign, and the turning signs. Not only the classifier cannot tell the difference between a real traffic sign and a generated noise image, but also any detectors based on z_{in} for OOD detection will fail. We note that adversarial robustness of traffic sign classifiers has been studied (Eykholt et al., 2018), and after adding adversarial noises to the traffic sign images, the noisy images are still recognizable. OOD noises and adversarial noises are very different (discussed in sections 2.1 and 2.2). Thus, it would be wise to disable any vision-based auto-pilot in your self-driving cars today until this issue is resolved.

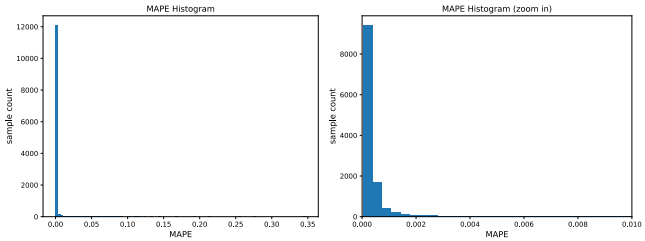
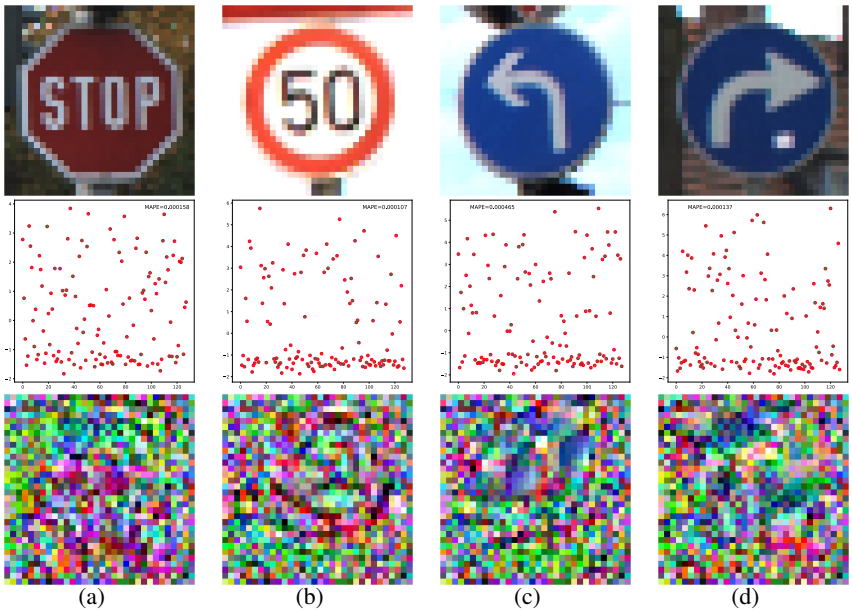


Figure 11: Left: MAPE histogram. Right: zoom-in view of the histogram



(a) (b) (c) (d)

Figure 12: The 1st row shows four traffic sign images. The 3rd row shows the generated OOD images. The 2nd row shows the scatter-plots of z_{out} (red) and z_{in} (blue). MAPE values are embedded in these scatter-plots. Please zoom-in for better visualization.