Generalized Out-of-distribution Indicator via Deep Generative Models

Abstract

We consider the problem of detecting out-of-distribution images by deep generative models. We reviewed the recent researches about out-of-distribution based on deep generative models and found a critical weakness that the number of datasets in these works is not enough to support the generality of their indicators. Therefore, we do a large-scale research for these indicators on more datasets and find that the performance of these indicators is not well. We propose the native indicators based on KullbackLeibler divergence and Wasserstein distance for detecting out-of-distribution. Our indicators are based on the theoretical derivation that the mixture distribution can be used to replace the out-distribution while the former is available. Our indicators outperform past works by a large margin. Moreover, we test our indicators under other limitations to show that our indicators are not data-specific and can be deployed on online system.

1 Introduction

Machine learning has achieved impressive success in classification domain through deep neural network classifiers. However, these classifiers can be easily fooled by the data from other datasets (called out-of-distribution OoD) (Nguyen, Yosinski, and Clune 2015), which might lead to potential risks in practice. It is necessary to detect OoD data to ensure that applications based on classifiers are robust and stable. Ostensibly, it is intuitive to train a density model $p_\theta(x)$ to approximate the empirical distribution of training data, and refuse the sample x when $p_\theta(x)$ is sufficiently low **??**. Recently, deep generative models are quickly developed and have been able to generate realistic samples, which indicates that deep generative models assign high density to the samples of the training data, and density can be used for detecting out-of-distribution.

However, contrary to popular belief, recent works show that density estimates by deep generative models assign higher density to the most samples from out-of-distribution and fail to detect OoD.

2 Background

Likelihood-based generative models are widely viewed to be robust to detect out-of-distribution samples by the model density intuitively. However, the densities of common likelihood-based models, *e.g.*, RealNVP (Dinh, Sohl-Dickstein et al. 2016), VAE (Tomczak and Welling 2018; Takahashi, Iwata et al. 2019) and PixelCNN (Van den Oord, Kalchbrenner et al. 2016), have been shown to be problematic for detecting out-of-distributions (Nalisnick, Matsukawa et al. 2018). To solve this problem, some researches proposed some variants of these models for detecting out-of-distribution (Che et al. 2019) and some researches proposed improved indicator to replace log-likelihood (Serrà et al. 2019). In our paper, we focus on how to improve the indicators based on common models.

(Song, Kim et al. 2017) considered using permutation tests statistics $T_{perm}(x)$ to determine whether an input x is in out-of-distribution. They use the rank of $p_\theta(x)$ in the training set as out-of-distribution indicators. Both low-likelihood and high-likelihood samples are identified as OoD. It is significantly useful to solve the counterexample of CIFAR-10 vs SVHN in (Nalisnick, Matsukawa et al. 2018).

(Choi, Jang et al. 2018) used Watanabe Akaike Information Criterion (WAIC) based on independently trained model ensembles.

$$\mathrm{WAIC}(x) = \mathbb{E}_\theta[\log p_\theta(x)] - \mathrm{Var}_\theta[\log p_\theta(x)] \tag{1}$$

(Ren et al. 2019) proposed a likelihood ratio indicator for deep generative models. They proposed a background model $p_{\theta_0}(x)$ to capture the general background statistics and a likelihood ratio indicator $LLR(x)$ to captures the significance of the semantics compared to the background model.

$$LLR(x) = \log p_\theta(x) - \log p_{\theta_0}(x) \tag{2}$$

(Nalisnick, Matsukawa et al. 2018) observed that input complexity excessively affects the generative models' likelihoods. Based on this observation, they proposed an estimation for input complexity $L(x)$, to derive a parameter-free OoD indicator $S(x)$:

$$S(x) = -\log p_\theta(x) - L(x) \tag{3}$$

(Song, Song et al. 2019) observed that on generative models with batch normalization, the estimated likelihood of a batch of OoD samples is much lower than of of in-distribution samples. Meanwhile, the corresponding log-likelihood decreases dramatically for OoD samples, but is relatively stable for in-distribution samples, as the ratio of test samples in a batch increases. Based on the observation, they proposed a permutation test that statics the change of log-likelihood when the ratio of test samples in a batch changes.

Researches also proposed to use labels (for classification task) to solve OoD. (Che et al. 2019) proposed to use conditional deep generative models to verify the predictions of classifier, when the labels of samples are selected by the classifier. (Alemi, Fischer, and Dillon 2018) models the bottleneck $I(Z;Y) - \beta I(Z;X)$ where I is the mutual information. (Hendrycks and Gimpel 2016; Hendrycks, Mazeika, and Dietterich 2018; Hsu et al. 2020; Lee et al. 2018) proposed to only use classifier for detecting OoD.

3 Problem Statement

Let p_{in} and p_{out} denote two distinct data distribution define on image space $\mathcal{X}$ where p_{in} is called in-distribution and p_{out} is called out-of-distribution. Let p_{mix} denote a mixture distribution $p_{mix}(x) = \alpha p_{in}(x) + \beta p_{out}(x)$ where $\alpha + \beta = 1$ and $\alpha, \beta > 0$. For samples generated by p_{mix}, they are generated by p_{in} or p_{out} and we want to know which of them are generated by p_{in} while we do not know any information about p_{out}.

It is important to decide a whether two datasets are distinct when the number of datasets is large. We call two datasets are **simply-classified** when a common classifier which is trained on the training data of the two datasets for 2-class classification,*i.e.*, in-distribution as label 0 and out-of-distribution as label 1, can simply distinguishing the testing data of the two datasets. If two dataset A, B are simply-classified, we will call their distribution is distinct and A vs B (A as in-distribution and B as out-of-distribution) will be an experiment.

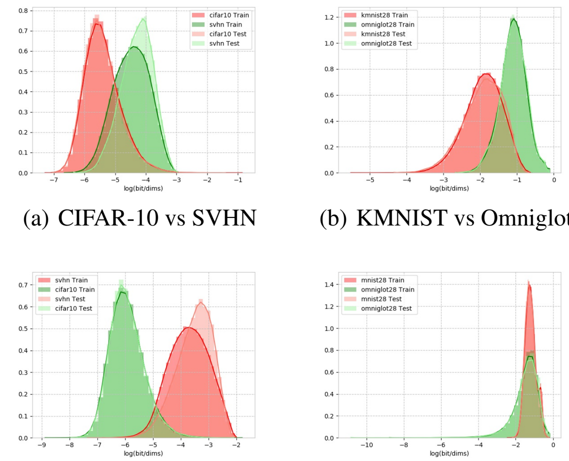
In this paper, we consider the problem of distinguishing the in-distribution and out-of-distribution images on the common deep generative models, *i.e.*, VAE, PixelCNN, flow-based models and GANs. To validate the generality of indicators, large-scale common datasets is used in our experiments, including MNIST, FASHION-MNIST, KMNIST, NOT-MNIST, Omniglot, CIFAR-10, CIFAR-100, TinyImagenet, SVHN, iSUN, CelebA, LSUN, Noise and Constant.

We consider all common metrics, including AUROC, AUPR, AP, FPR@TPR95. AUROC is selected as the major metrics in our paper and other metrics are shown in appendix B. AUROC is a threshold-independent metric (Davis and Goadrich 2006) and is widely used in OoD domain. A perfect detector can get AUROC score of 100%.

4 Motivating Observations

Counterexamples

The log-likelihood of likelihood-based models is expected to be lower in the out-of-distribution and be higher in the in-distribution intuitively, because the models are trained at in-distribution. However, the observation of (Nalisnick, Matsukawa et al. 2018) shows that all of models give the higher log-likelihood at out-of-distribution in experiments CIFAR-10 vs SVHN and NotMNIST vs MNIST. We observed that the number of datasets in (Nalisnick, Matsukawa et al. 2018) is quite small and we suspect that there might be more counterexamples at large-scale datasets.



(a) CIFAR-10 vs SVHN

(b) KMNIST vs Omniglot

(c) SVHN vs CIFAR-10

(d) MNIST vs Omniglot

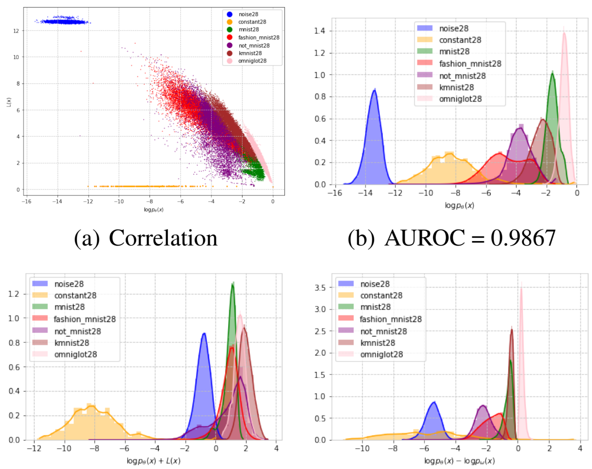
Figure 1: The log-likelihood of VAE. The AUROC of log-likelihood in (a) (b) (c) (d) are 0.08, 0.09, 0.99, 0.59. The AUROC of T-perm in (a) (b) (c) (d) are 0.84, 0.82, 0.98, 0.66. These experiments show that log-likelihood of out-of-distribution might be higher, lower or nearly same to the log-likelihood of in-distribution.

Therefore, we reproduce the experiments at more datasets and find more counterexamples shown in 1 and appendix A. These experiments show that log-likelihood is unpredictable at out-of-distribution, *i.e.*, it might be lower, higher or same. Moreover, the methods based on the log-likelihood might have counterexamples at large-scale datasets. We reproduce the indicators (Alemi, Fischer, and Dillon 2018; Song, Kim et al. 2017; Ren et al. 2019; Song and Ermon 2019; Nalisnick, Matsukawa et al. 2018; Che et al. 2019; Alemi, Fischer, and Dillon 2018) on common generative models and find counterexamples at large-scale datasets, shown in appendix A. (Nalisnick, Matsukawa et al. 2018) observed that there is a clear negative correlation between likelihoods trained on CIFAR-10 and FashionMNIST and their complexity estimates over large-scale datasets. We validate their observation over large-scale datasets. However, such correlation is depend on the in-distribution. It matters the performance of indictors as shown in fig. 2.

Performance on Large-scale Datasets

Based on the following reasons, we use the large-scale datasets instead few datasets to check the generality of indicators:

- The critical reason is that their indictors are supported by the motivating observation on few datasets, instead of the theoretical proof, however, the datasets in the past works are too less to validate the generality of motivating observation.
- The performance on few datasets might be early im-



(a) Correlation (b) AUROC = 0.9867

(c) AUROC = 0.7770 (d) AUROC = 0.9999

Figure 2: We select Omniglot as in-distribution and other grey datasets as out-of-distribution. (a) shows the correlation between likelihoods trained on Omniglot and complexity estimate. (b) shows the histogram of log-likelihood. The average AUROC over all testing datasets is 0.9867. (c) shows that indictor $S(x)$ might reach lower the performance than log-likelihood. It means that $L(x)$ is rough and we need a more precise, stable and interpretable estimate to assist log-likelihood for detecting OoD. (d) shows $\log p_\theta(x) - \log p_\omega(x)$ might be a good choice where p_ω is trained on out-of-distribution.

proved by fine-tuning hyper-parameters and architectures of models but not on large-scale datasets.

- By the definition of OoD problem in section 3, models should distinguish any p_{out} that is distinct to p_{in} instead of few datasets. Therefore, we should check more out-of-distribution as much as possible.
- Average performance on large-scale datasets is better for assessing indictors. In CelebA vs LSUN, log-likelihood reach 0.98 AUROC (1.0 is best), however, our counterexamples show that it gets 0.02 in CelebA vs SVHN. Average performance will consider such experiments with lower AUROC. We think that it is more meaningful to improve the average performance of indictors rather than improve little in a single experiment.

Our experiments in **??** show that on large-scale datasets the performance of the indictors of past work via common deep generative models are not good as they claimed.

Observation of KL indicator

As shown in fig. 2, complexity estimate is unstable and sometimes it might lower the performance. Therefore, we try to find another function to replace the term $L(x)$ in $S(x)$. In our experiments over large-scale datasets, we observe a phenomenon that $p_\theta(x) < p_\omega(x)$ for almost $x \sim p_{out}$ and $p_\theta(x) > p_\omega(x)$ for almost $x \sim p_{in}$ where $p_\omega(x)$ is a model trained on out-of-distribution. The average AUROC of $p_\theta(x) - p_\omega(x)$ reaches nearly 1.0 over all datasets, as shown in **??**. From the view of complexity estimate, $L(x) = -\log_2 p(x|\mathcal{M}_0)$ can be regard as a log-likelihood of a universal model $\mathcal{M}_0$ (Serrà et al. 2019). In our paper, $p_\omega(x)$ is used to replace $L(x)$ for assisting log-likelihood.

However, in OoD problem, we are not allowed to know any information of the out-of-distribution. Therefore, we will explore how to develop an indictor that are not depend on a model trained on out-of-distribution and why $p_\theta(x) - p_\omega(x)$ is almost useful for OoD theoretically.

5 Native Indicators

Basic Native Indicators

In the definition of OoD problem, we do not give any assumption about the probability distribution, since we try to make the definition more general. Our theory are based on following assumption about the probability distribution for OoD problem:

1. The training data and testing data in in-distribution and out-of-distribution are i.i.d.
2. There exists a divergence div, satisfying that $div(p_{in}, p_{out}) \gg 0$ and $div(p_{out}, p_{in}) \gg 0$.
3. If $div(p_{in}, p_{out})$ and $div(p_{out}, p_{in})$ can be represented by sampling formula, *i.e.*, $div(p_{in}, p_{out}) \approx \sum_i f(x_i)$, then $f(x_i) \gg 0$ for almost x_i.
4. $f : \mathcal{X} \to \mathcal{R}$ maps the distribution p_{in}, p_{out} into two Gaussian distribution with constant variance.

It is important to notice that assumption 3 can not be deduced from assumption 2 obviously. For example, when $p_{in} = \frac{1}{2}(\mathcal{N}(0,1) + \mathcal{N}(-10,1))$ and $p_{out} = \frac{1}{2}(\mathcal{N}(0,1) + \mathcal{N}(10,1))$, $KL(p_{in}, p_{out}) \approx 25.05 \gg 0$ but $\log p_{in}(x) - \log p_{out}(x) \approx 0$. In such example, p_{in} and p_{out} is distinguishing by KL-divergence but is not simply classified because no classier can detect whether a sample in $\mathcal{N}(0,1)$ is from p_{in} or p_{out}. We assume that assumption 3 holds when p_{in} and p_{out} is simply classified.

Above assumption means that if D can distinguishing the probability distribution in-distribution and out-of-distribution, then f can be used as an indicator for detecting whether a sample is OoD. f is called the Native Indictor based on D.

For detailed research, *e.g.*, how to use native indictor when we do not know out-of-distribution, we need more property of div. We will introduce two kinds of Native Indictors: Native KL Indictor and Native Wasserstein Indictor.

Native KL Indicator

We select KL-divergence as D, then $KL(p_{in}, p_{out}) = \mathbb{E}_{p_{in}(x)}[\log p_{in}(x) - \log p_{out}(x)]$ and Native KL Indictor is $\log p_{in}(x) - \log p_{out}(x)$. fig. 3 interprets the motivation of native KL indictor.

Theorem 1 *$\log p_{in}(x) - \log p_{out}(x)$ is a symmetric indictor, i.e., it reaches same performance in experiment A vs B and B vs A.*

Likelihood is not a symmetric indictor since it uses p_A as indictor in experiment A vs B and p_B as indictor in experiment A vs B. An interesting phenomenon is that if likelihood

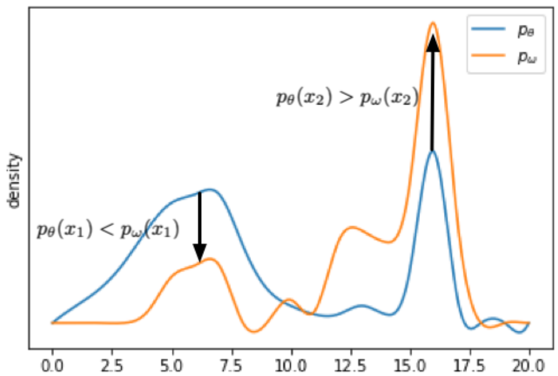


Figure 3: Diagrammatic sketch for the motivation of Native KL Indictor. The in-distribution is in [0, 10] and out-of-distribution in [10, 20]. Intuitively, p_θ has a lower density than p_ω for samples from in-distribution and higher density for samples from out-of-distribution. Different to the assumption of log-likelihood indictor, we do not assume that $p_\theta(x_2)$ or $p_\in(x_2)$ is a very lower density for $x_2 \sim p_{out}$. By the assumption 1,2,3,4, we can induce that $p_{in}(x_1) < p_{out}(x_1)$ and $p_{in}(x_2) > p_{out}(x_2)$. We expect that p_θ and p_ω also have this property, *i.e.*, $p_\theta(x) - p_\omega(x)$ is an indictor for OoD detection.

indictor reaches a good performance on experiment A vs B, then it will usually fail in experiment B vs A, in appendix B.

However, p_{out} is unknown under the limitation of OoD problem. Therefore, we try to use another tractable term to replace p_{out}.

Theorem 2 *For any mixture distribution $p_{mix} = \alpha p_{in} + \beta p_{out}$ where $\alpha + \beta = 1$ and $\alpha, \beta > 0$, the performance of indictor $\log p_{in}(x) - \log p_{mix}(x)$ and indictor $\log p_{in}(x) - \log p_{out}(x)$ are equal for OoD detection. A special example is the experiment of Noise vs any dataset. The optimal model of Noise dataset is uniform distribution, however, it can not detect any OoD. In the experiment of any dataset vs Noise, likelihood reaches nearly best performance since the log-likelihood of noise is significantly lower than any datasets.*

If we can get enough data from the mixture distribution, By theorem 1, a model p_γ that is trained to approximate p_{mix} can be used to replace the term p_{out} in Native KL Indictor and ensure the same performance. It is also observed in our experiments, shown in **??**. Next, we try to explain why native KL indictor can reach outstanding performance.

Compared to log-likelihood indictor and likelihood-ratio indicator, Native KL Indictor can always get better performance.

Theorem 3 *By the assumption that log-likelihood can detect OoD, $\log p_{in}(x_1) > \log p_{in}(x_2)$ for almost $x_1 \sim p_{in}$ and $x_2 \sim p_{out}$. At such situation, assumption 3 can be deduced, i.e., Native KL Indictor can be also used to detect OoD.*

Theorem 4 *For any likelihood-ratio indictor in formula $\log p_{in}(x) - \log g(x)$ where g is a continuous differentiable probability distribution, Native KL Indictor outperforms them in OoD problem with suitable mathematical simplification.*

By above theorems, we show that Native KL Indictor can be obtained by mixture distribution. It is natural to use two modeled distribution to model in-distribution and mixture distribution.

Next, we will show the property of modeled distribution. Let p_θ, p_γ models p_{in}, p_{mix}, *i.e.*, $\theta = \arg\max_\theta \mathbb{E}_{p_{in}(x)} \log p_\theta$ and $\gamma = \arg\max_\gamma \mathbb{E}_{p_{mix}(x)} \log p_\omega$. $\log p_\gamma(x) - \log p_\theta(x)$ acts as the indictor.

Theorem 5 *$\log p_\theta(x) - \log p_\gamma(x)$ can be used for detecting OoD, only requiring that the training converges, without the condition that p_θ sufficiently approaches p_{in} and p_γ sufficiently approaches p_{out} with suitable mathematical simplification. Moreover, $\log p_\theta(x) - \log p_\gamma(x)$ reflects whether a sample x in mixture distribution have been optimized in the training process of θ.*

theorem 4 shows $\log p_\theta(x) - \log p_\gamma(x)$ reflects the optimizability: if a sample x is OoD, then $p_\theta(x)$ will be lower than $p_\gamma(x)$ since $x \sim p_{mix}$ and γ can optimize x more than θ. By theorem 4, we will not require that θ approach p_{in} sufficiently and γ approach p_{mix} sufficiently but only the training of θ and γ converges. It alleviates the requirements for training and then we can discuss how to train models on online system.

Native Wasserstein Indicator

In (Nalisnick, Matsukawa et al. 2018), they do not include GANs in the comparison since how to evaluate the density of GANs is an open problem. Some researches propose variants of GANs with tractable density (Kumar, Goyal et al. 2019). In the framework of Native Indictors, we can detect OoD by GANs without the density of GANs.

In vanilla GAN (Goodfellow, Pouget-Abadie et al. 2014), a discriminator is trained to distinguish generated samples and real samples and a generator is trained to generate samples for deceiving the discriminator.

However, vanilla GAN is unstable during the training process. To tackle this problem, Wasserstein distance is introduced by WGAN (Arjovsky, Chintala et al. 2017):

$$W^1(\mu, \nu) = \sup_{Lip(D) \leq 1} \{\mathbb{E}_{\mu(x)} D(x) - \mathbb{E}_{\nu(x)} D(x)\} \quad (4)$$

We select Wasserstein distance as div, then

$$W^1(p_{in}, p_{out}) \approx \sum_i \left[D(x_i^{in}) - \sum_j D(x_j^{out})\right] \quad (5)$$

where x_i^{in} is sampled from in-distribution and x_j^{out} is sampled from out-of-distribution and D is the optimal solution in $W^1(p_{in}, p_{out})$. Then the Native Wasserstein Indicator for sample x is $D(x) - \sum_j D(x_j^{out})$. Notice that $\sum_j D(x_j^{out})$ is a constant and it will not affect the performance of OoD detection. Therefore, the Native Wasserstein Indicator is simply $D(x)$. The generator of WGAN does not appear in our formula since we only consider how to measure the distance between p_{in}, p_{out} instead of generating.

Similar to Native KL Indicator, we will show some theorems to show the property of Native Wasserstein Indicator to ensure that it can be obtained without any information of p_{out}.

Theorem 6 *Assumption 2 is a corollary of definition of OoD problem when div is Wasserstein distance.*

Theorem 7 *$D(x)$ is a symmetric indictor.*

Theorem 8 *$\hat{D}$ that is optimal solution in $W^1(p_{in}, p_{mix})$ is same to the optimal solution D in $W^1(p_{in}, p_{out})$.*

Because the Native Wasserstein Indicator is only depend on D, by theorem 5 we could use $\hat{D}$ that is independent to p_{out} to replace D.

Theorem 9 *The discriminator in $W^1(p_{in}, p_{out})$ is the best indictor among all indictors that is 1-Lipschitz. Moreover, it is the best indictor who has limited gradient.*

Concerns

We have proposed a method to exploit the samples of mixture distribution to train a model for OoD detection. However, there are three major concerns about this idea:

- Is this method **data-specific**? *i.e.*, does this method only work on the data that it has seen in training?
- Can this method work on **online system**? *i.e.*, for a new testing data, model should give the indictor before the next texting data comes.
- Can this method works only with **a batch of samples**? *i.e.*, model must detect immediately while it only knows in-distribution data and a batch of mixture distribution.
- Is it better to train **pre-train** models (trained on in-distribution) or **initialized** models on on mixture distribution?

To solve the concerns, we design following experiments:

- In experiment A vs B, only 20% data in mixture distribution can be used for training.
- We simulate the online system — the data in mixture distribution is streaming and requires the indicator immediately.
- We splits the data from mixture distribution into several batches and model can only know the in-distribution and the batch that it is detecting.
- Experiments for pre-train models and initialized models.

6 Experiments

In this section, we will demonstrate the effectiveness of Native KL Indictors and Native Wasserstein Indictors on several computer vision benchmark datasets. We run all experiments with Pytorch and Tensorflow and we submit the code to reproduce all experimental results.

Datasets

All the datasets considered are listed below:

1. **CIFAR-10** is a natural image datasets with 10 classes including animal, ship, airplane and etc.
2. **CIFAR-100** is just like CIFAR-10, including 100 classes.
3. **SVHN** includes house numbers from 0 to 9.
4. **CelebA** is a large-scale face attributes dataset.
5. **TinyImageNet** consists of a subset of ImageNet images. It contains 200 different classes.
6. **LSUN** has a testing set of 10 different scenes.
7. **iSUN** is subset of SUN, including 8925 scene images in 899 different scenes.
8. **MNIST** consists of handwritten digits from 0 to 9.
9. **Fashion MNIST** includes 10 kinds of clothes and shoes.
10. **Not MNIST** includes letters from A to J on various typefaces.
11. **KMNIST** includes 10 kinds of Kanji characters.
12. **Omniglot** contains 1623 different handwritten characters from 50 different alphabets.
13. **Noise** is created by uniformly randomly sampling.
14. **Constant** includes images whose pixels have same color.

The natural images are resized into 32x32x3 and grey images are all 28x28x1. All pair in natural image datasets and all pair in grey image datasets are considered in our experiments. Only CIFAR-10, CIFAR-100 and TinyImageNet are not simply-classified and in fact they have similar classes. Noise and Constant includes both grey and natural images. LSUN (only including testing data) and iSUN are only used for out-of-distribution.

CelebA, Noise and Constant have no labels, thus we create random labels whose value in 0-9 on these datasets.

Metrics

Following metrics are adopted to measure the effectiveness of a method in out-of-distribution detection:

- **AUROC** is the Area Under the Receiver Operating Characteristic curve, which is a threshold-independent metric (Davis and Goadrich 2006). AUROC can be interpreted as the probability that a sample from in-distribution is assigned a lower detection score than a sample from out-of-distribution (Fawcett 2006). It is widely applied in OoD domain. We select AUROC as our major metrics.
- **AUPR** is the Area under the Precision-Recall curve, which is another threshold independent metric (Saito and Rehmsmeier 2015). AUPR-In and AUPR-Out denotes the AUPR where in-distribution and out-of-distribution are positive, respectively.
- **AP** is the Average Precision. AP summarizes a precision-recall curve as the weighted mean of precisions achieved at each threshold.
- **FPR@TPR95** is the False Positive Rate when True Positive Rate is over 95%. It means the probability that an out-of-distribution example is misclassified as in-distribution when over 95% in-distribution is detected accurately.

Setups

For fair comparison, all indictors are based on common models with standard training, including 34-layer ResNet, VAE, PixelCNN, Glow and Wasserstein GAN. ResNet34 is trained to classify the in-distribution and out-of-distribution for validate whether they are simply classified. ResNet34 is also trained for classification and serves for the OoD indictors based on classifier. Deep generative models are trained as their proposer suggests without any other tricks.

In our experiments, there are not validation set. For the indictors depending on hyper-parameters, we try grid-searching as their proposers suggest and report the performance with all hyper-parameters considered. To ensure the generality over all datasets, it is forbidden to specify the hyper-parameters or architectures on any dataset. The detailed architecture and parameters are shown in appendix B.

Major Results

The main results for indictors of past works are summarized in **??**. The main results for Native KL Indictor and Native Wasserstein Indictors are summarized in **??**.

Validation of Theorem

theorem 1 is supported by the detail experiments shown in appendix B. theorem 2 is supported by the performance of native KL indictor based on mixture distribution. **??** shows that native KL indictor can also get same performance when log-likelihood can reach outstanding performance, to support theorem 3.

We compare the precision-recall curve and ROC curve of native KL indictor, log-likelihood indictor, likelihood ratio indictor and log-likelihood with input complexity in **??**, to show native KL indictor is the best version of log-likelihood ratio indictor, to support theorem 4. **??** shows the change of AUROC during training on mixture distribution to support theorem 5.

theorem 7 is supported by the detail experiments shown in appendix B. The performance The detection ability of native Wasserstein indictor is shown in **??**. The performance of the native Wasserstein indictor trained on mixture distribution is near same as the native Wasserstein indictor trained on out-of-distribution, which validates theorem 8. The outstanding performance of the native Wasserstein indictor supports the theorem 9.

Concerns

The experiments about concerns are showing in **??**. **??** shows the change of AUROC while the size of batch varies. These experiments show that native KL indicator and native Wasserstein indictor are effective, robust, general, online and not data-specific. It also show the weakness of these native indictors that it can not work well when only a few data from mixture distribution is provided.

7 Limitations of the Study

In this section, we will discuss the limitation of our paper in datasets, models and native indictors.

Limitation of datasets

In our paper, we try large-scale datasets to show the generality of indictors. However, we only consider natural OoD datasets **??** and do not consider attacked OoD. An important reason is that it is quite hard to design a principle like simply classified, to measure these attacked OoD datasets.

Limitation of models

In our paper, for fair comparison and the research for generality, we only consider the common models, including ResNet, VAE, PixelCNN, Glow and WGAN. However, there are numerous models careful-designed for OoD detection. Limited by our computation resource, we can not give the performance of them over large-scale datasets. We have emphasized that it is unreliable to evaluate the models for OoD in a few datasets in this paper and we expect that it can become a common sense in OoD domain.

Limitation of native indictors

While native indictors show outstanding performance in our experiments, we have also observed some defects of them. They depend on the model (*e.g.*, Native KL Indicator based on Glow is data-specific) and optimizer (*e.g.*, when data from mixture distribution is not enough for training, optimizer can not give the expected p_γ expect). It is natural to design an indicator to approximate $\log p_\gamma(x) - \log p_\theta(x)$ without optimizer, *e.g.*, by Taylor Expansion. We leave it for future work. Native indictors will lead to many works since it rise simply, fundamental and practical indictors with solid experiments and theories.

8 Conclusion

References

Alemi, A. A.; Fischer, I.; and Dillon, J. V. 2018. Uncertainty in the variational information bottleneck. *arXiv preprint arXiv:1807.00906* .

Arjovsky, M.; Chintala, S.; et al. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, 214–223.

Che, T.; Liu, X.; Li, S.; Ge, Y.; Zhang, R.; Xiong, C.; and Bengio, Y. 2019. Deep verifier networks: Verification of deep discriminative models with deep generative models. *arXiv preprint arXiv:1911.07421* .

Choi, H.; Jang, E.; et al. 2018. Waic, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392* .

Davis, J.; and Goadrich, M. 2006. The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning*, 233–240.

Dinh, L.; Sohl-Dickstein, J.; et al. 2016. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803* .

Fawcett, T. 2006. An introduction to ROC analysis. *Pattern recognition letters* 27(8): 861–874.

Goodfellow, I.; Pouget-Abadie, J.; et al. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

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

Hendrycks, D.; Mazeika, M.; and Dietterich, T. 2018. Deep anomaly detection with outlier exposure. *arXiv preprint arXiv:1812.04606* .

Hsu, Y.-C.; Shen, Y.; Jin, H.; and Kira, Z. 2020. Generalized odin: Detecting out-of-distribution image without learning from out-of-distribution data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10951–10960.

Kumar, R.; Goyal, A.; et al. 2019. Maximum Entropy Generators for Energy-Based Models. *arXiv preprint arXiv:1901.08508* .

Lee, K.; Lee, K.; Lee, H.; and Shin, J. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems*, 7167–7177.

Nalisnick, E.; Matsukawa, A.; et al. 2018. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136* .

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

Ren, J.; Liu, P. J.; Fertig, E.; Snoek, J.; Poplin, R.; Depristo, M.; Dillon, J.; and Lakshminarayanan, B. 2019. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, 14707–14718.

Saito, T.; and Rehmsmeier, M. 2015. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS one* 10(3): e0118432.

Serrà, J.; Álvarez, D.; Gómez, V.; Slizovskaia, O.; Núñez, J. F.; and Luque, J. 2019. Input complexity and out-of-distribution detection with likelihood-based generative models. *arXiv preprint arXiv:1909.11480* .

Song, J.; Song, Y.; et al. 2019. Unsupervised Out-of-Distribution Detection with Batch Normalization. *arXiv preprint arXiv:1910.09115* .

Song, Y.; and Ermon, S. 2019. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, 11895–11907.

Song, Y.; Kim, T.; et al. 2017. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766* .

Takahashi, H.; Iwata, T.; et al. 2019. Variational Autoencoder with Implicit Optimal Priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 5066–5073.

Tomczak, J.; and Welling, M. 2018. VAE with a VampPrior. In *International Conference on Artificial Intelligence and Statistics*, 1214–1223.

Van den Oord, A.; Kalchbrenner, N.; et al. 2016. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, 4790–4798.

A Counterexamples

B Experiemtns

C Proof