A Framework of Robust Out-of-Distribution Indicators Based on Divergence in Deep Generative Models

Abstract

To ensure robust and reliable classification results, OoD (out-of-distribution) indicators based on deep generative models are proposed recently and are shown to work well on small scale experiments. In this paper, we conduct the first large-scale experiments (1 order of magnitude larger than previous ones) for existing OoD indicators and observe that none of them perform well. We thus advocate that large-scale experiments are mandatory for evaluating OoD indicators. We then propose a novel theoretical framework ROID for robust Out-of-Distribution indictors based on divergence (instead of the traditional likelihood assumption) in deep generative models. Following this framework, we further propose two effective and robust divergence-based indictors: KL-based and Wasserstein-based. These two indictors significantly outperform past works by 10% in AUROC and its performance is close to the optimal. Moreover, our systematic experiments show that our indicators are not data-specific and can be deployed online.

1 Introduction

Machine learning has achieved impressive success in the classification domain through deep neural network classifiers (Szegedy et al. 2016; He et al. 2016; Zagoruyko and Komodakis 2016). Knowing when a machine learning (ML) model is qualified to make predictions on an input is critical to safe deployment of ML technology in the real world (Choi, Jang et al. 2018). When training distribution (called in-distribution) differ from test distributions (called Out-of-Distribution), neural networks may provide (with high confidence) arbitrary predictions on inputs that they are unaccustomed to seeing. This is known as the Out-of-Distribution (OoD) problem(Choi, Jang et al. 2018). For example, a classifier trained on CIFAR-10 (Krizhevsky, Hinton et al. 2009) may recognize the house number in SVHN (Netzer, Wang et al. 2011) as a horse.

Therefore, it is crucial to develop **OoD indicators** for detecting OoD data, to ensure that applications based on classifiers are robust and reliable. The common belief (Bishop 1994) is that the OoD indicators can be based on density model: train a density model $p_\theta(x)$ (as an OoD indicator) to approximate the empirical distribution of training data, and refuse the sample x when $p_\theta(x)$ is sufficiently low. However, recent works (Nalisnick et al. 2019; Choi, Jang et al. 2018; Hendrycks, Mazeika, and Dietterich 2018) show that density estimates by *deep generative models* (Dinh, Sohl-Dickstein et al. 2016; Tomczak and Welling 2018; Takahashi, Iwata et al. 2019; Van den Oord, Kalchbrenner et al. 2016), which generate realistic samples, assign higher density to samples from out-of-distribution. For example, according to (Nalisnick et al. 2019), this phenomenon occurs in CIFAR-10 (as in-distribution) vs SVHN (as out-of-distribution) for different likelihood-based models, while the data in CIFAR-10 and SVHN have significant different semantics.

To to alleviate the aforementioned phenomenon, more advanced OoD indicators (Serrà et al. 2019; Song, Kim et al. 2017; Choi, Jang et al. 2018; Ren et al. 2019; Song, Song et al. 2019; Che et al. 2019) are proposed recently based on deep generative models and shown to perform well on small-scale datasets,*e.g.*, the number of dataset pairs (in-distribution dataset, out-of-distribution dataset) are only 1 to 10. However, a *robust* OoD indicator should detect samples from any out-of-distribution (Chen et al. 2020), thus should be evaluated through much-larger-scale experiments. In this paper, we first conduct large-scale experiments with 92 dataset pairs (based on 14 popular image datasets, including MNIST, FASHION-MNIST, KMNIST, NOT-MNIST, Omniglot, CIFAR-10, CIFAR-100, TinyImagenet, SVHN, iSUN, CelebA, LSUN, Noise and Constant), whose scale is 1 order of magnitude larger than all above works. We observe that none of the above OoD indicators perform well on our large-scale experiments (see later in table 1). Based on this observation, we advocate that *small-scale experiments are unreliable and large-scale experiments are mandatory for evaluating OoD indicators*.

Another very interesting observation that we discover by accident, as a result of the fact that we try to enumerate the dataset pairs when possible (*e.g.*, (CIFAR-10, SVHN) and (SVHN,CIFAR-10) are both in our experiment setting), is that, on all dataset pairs, *p_θ has a higher density for samples from in-distribution and lower density for samples from out-of-distribution than p_ω*, which is trained on the out-of-distribution. Inspired by this intuitive (in retrospect) observation, we propose a theoretical framework *ROID* for **R**obust **O**ut-of-Distribution **I**ndictors based on **D**ivergence in deep generative models. Following this framework, we

further propose two effective and robust divergence-based indictors in two mainstream deep generative models: 1) for likelihood-based models, we propose and prove a KL-based OoD indicator; 2) for WGAN (a popular version of GAN with more stable training), we propose and prove a Wasserstein-based OoD indicator. Our large-scale experiments show that our two indicators significantly outperform existing works by 10% in AUROC and its performance is close to the theoretical optimal results.

The main contributions of this paper are as follows:

- We conduct the first large-scale experiments (1 order of magnitude larger than previous ones) for existing OoD indicators and observe that none of them perform well. We thus advocate that *small-scale experiments is unreliable and large-scale experiments are mandatory for evaluating OoD indicators*.
- We propose a novel theoretical framework *ROID* for robust Out-of-Distribution indictors based on divergence (instead of the traditional likelihood assumption)in deep generative models. Following this framework, we further propose two effective and robust divergence-based indictors: KL-based and Wasserstein-based. We believe that more indictors can be proposed following *ROID* framework.
- Our KL-based indicator and Wasserstein-based indicators significantly outperform past works by 10% in AUROC and its performance is close to the optimal. Moreover, our systematic experiments show that our indicators are not data-specific and can be deployed online.

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 et al. 2019). These likelihood-based models assign higher likelihood for samples from SVHN (out-of-distribution) than samples from CIFAR-10 (in-distribution).

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 on common models (Serrà et al. 2019). Common models are widely applied in images domain and variants are not. Moreover, evaluation on numerous variants on large-scale datasets is more expensive while common models are easy for training and many indicators can share one well-trained model. Furthermore, it is necessary to check the generality of indicators on common models. By the above motivations, this paper focuses on the indicators based on common models.

(Song, Kim et al. 2017) considered using permutation tests statistics $T_{perm}(x)$ as the indicator to detect OoD. The rank of $p_\theta(x)$ in the training set is used as OoD 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 et al. 2019).

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

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

(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 capture the significance of the semantics compared to the background model.

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

(Serrà et al. 2019) 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 $S(x)$:

$$S(x) = -\log p_\theta(x) - L(x) \quad (3)$$

(Song, Song et al. 2019) observed that generative models with batch normalization assign much lower likelihood to OoD samples than in-distribution samples. Based on the insight, $T_{b,r_1,r_2}(x)$ is proposed for OoD detection.

Some researches also proposed to use labels (for classification tasks) to solve OoD. (Che et al. 2019) proposed $p(x|y)$ for OoD detection. It uses conditional deep generative models to verify the predictions of classifier. (Alemi, Fischer, and Dillon 2018) use VIB to model 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; Lakshminarayanan, Pritzel, and Blundell 2017) proposed some indictors based on classifier for detecting OoD.

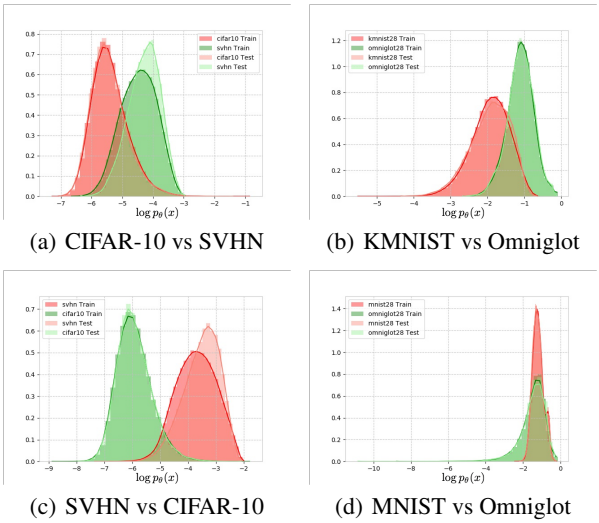
3 Problem Statement

p_{in} and p_{out} denote two distinct data distributions where p_{in} is called in-distribution and p_{out} is called out-of-distribution. p_{mix} denotes a mixture distribution $p_{mix}(x) = \alpha p_{in}(x) + \beta p_{out}(x)$ where $\alpha + \beta = 1$ and $\alpha, \beta > 0$. The samples from p_{mix}, are generated by p_{in} or p_{out} and *we wonder which of them are generated by p_{in} without knowing p_{out}*.

It is important to decide whether two datasets are distinct on large-scale datasets. Two datasets are **simply-classified** when a common classifier trained on the training data of two datasets for 2-class classification, *i.e.*, in-distribution as label 0 and OoD as label 1, can simply distinguish the testing data of the two datasets. If two dataset A, B are simply-classified, A vs B and B vs A dataset pairs will be considered in large-scale experiments.

In this paper, we consider the problem of distinguishing the in-distribution and out-of-distribution on the common deep generative models, *i.e.*, VAE, PixelCNN, flow-based models and GANs. Large-scale common datasets are used to validate the generality of indicators, as shown in Section 6.

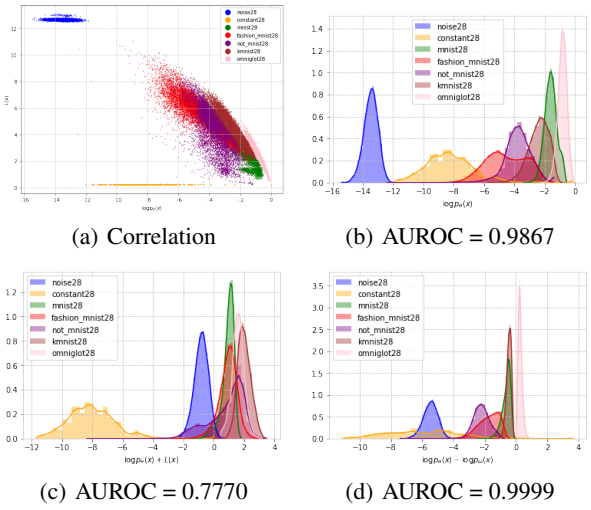
All common metrics including AUROC, AUPR, AP, FPR@TPR95 are considered in this paper. AUROC is selected as the major metric and other metrics are shown in appendix B. AUROC is a threshold-independent metric (Davis and Goadrich 2006) and is widely used in the OoD domain.



(a) CIFAR-10 vs SVHN (b) KMNIST vs Omniglot

(c) SVHN vs CIFAR-10 (d) MNIST vs Omniglot

Figure 1: The histogram of log-likelihood of VAE. Green and red parts denote the log-likelihood of out-of-distribution and in-distribution respectively. 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.



(a) Correlation (b) AUROC = 0.9867

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

Figure 2: We select Omniglot as in-distribution and other datasets as out-of-distribution. (a) shows the correlation between likelihoods trained on Omniglot and complexity estimate. (b) shows the histogram of log-likelihood, where the average AUROC is 0.9867. (c) shows that indicator $S(x)$ 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 $\log \frac{p_\theta(x)}{p_\omega(x)}$ might be a good choice where p_ω is trained on out-of-distribution.

4 Motivating Observations

Counterexamples

Generally, 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 on in-distribution. However, the observation of (Nalisnick et al. 2019) shows that VAE, PixelCNN and RealNVP all assign the higher log-likelihood to samples from out-of-distribution in experiments CIFAR-10 vs SVHN and NotMNIST vs MNIST. We observed that the number of datasets in (Nalisnick et al. 2019) is quite small and we suspect that there might be more counterexamples at large-scale datasets.

Therefore, we reproduce the experiments at more datasets and find more counterexamples shown in Figure 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 to in-distribution. 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 et al. 2019; 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 et al. 2019) observed that there is a clear negative correlation between likelihoods and complexity estimates, when the model is trained on CIFAR-10 and FashionMNIST. We validate their observation on large-scale datasets. However, since $L(x)$ is rough, it matters the performance of $S(x)$, as shown in Figure 2.

Furthermore, many counter-examples for OoD indicators not based on deep generative models, are shown in appendix A. *e.g.*, , (Lee et al. 2018) reaches 98.24% AUROC on SVHN vs CIFAR-10, but only 38.22% AUROC on Omniglot vs FashionMNIST. These counter-examples indicate the critical generality problem in the OoD domain. An important reason of this phenomenon is that *OoD indicators are always designed based on motivating observations only on few datasets, however, nothing can ensure these observations hold on large-scale datasets*. These counter-examples encourage the evaluation on large-scale datasets.

Performance on Large-scale Datasets

For the following reasons, large-scale datasets is used to check the generality of indicators:

1) Check observations OoD indicators based on the motivating observation on few datasets, are unreliable. It is necessary to validate the generality of motivating observation.

2) Avoid over fine-tuning The performance on few datasets might be easily improved by fine-tuning hyper-parameters and architectures of models, but hard on large-scale datasets.

3) Average Performance Average performance on large-scale datasets is better for assessing indicators. In CelebA vs LSUN, log-likelihood reaches 98% AUROC, however, it only reaches 0.02 in CelebA vs SVHN in appendix A. Average performance will consider such experiments with lower AUROC. It is more meaningful to improve the average performance of indicators than to improve little (*e.g.*, 99.1% to 99.2%) in a single experiment.

Indicators of previous works via common deep generative models do not perform well on large-scale datasets, as

indicator	Model	AUROC	AUPR
Recon	VAE	69.26	74.01
ELBO	VAE	69.44	74.14
ELBO - Recon	VAE	52.85	58.93
MCMC Recon	VAE	67.43	71.25
MCMC $\log p_\theta(x)$	VAE	67.45	71.36
Volume	RNVP	62.46	70.63
$\log p_\theta(z)$	RNVP	74.58	77.33
H	VIB	66.79	67.17
R	VIB	58.78	62.36
$D_\theta(x)$	WGAN	79.15	82.54
$\|\nabla_x D_\theta(x)\|$	WGAN	60.55	65.14
Disagreement	ResNet	69.25	69.75
Mahalanobis	ResNet	83.02	82.42
Entropy of $p(y\|x)$	ResNet	62.74	62.95
$\max_y p(y\|x)$	ResNet	61.59	64.90
ODIN	ResNet	60.68	60.98
DeConf-C	ResNet	68.59	71.81
DeConf-C*	ResNet	71.09	73.92
Perfect classifier	ResNet	99.85	99.87

indicator	VAE	PixelCNN	RNVP
$\log p_\theta(x)$	70.11	72.26	69.19
$T_{perm}(x)$	89.71	84.28	89.72
$\|\nabla_x \log p_\theta(x)\|$	53.95	NA	24.27
$S(x)$	81.88	88.33	80.11
$LLR(x)$	69.47	77.46	64.03
$WAIC(x)$	74.59	82.19	83.74
$Var_\theta[\log p_\theta(x)]$	83.11	82.21	86.06
$\log p(x\|y)$	53.13	69.22	71.27
$T_{b,r_1,r_2}(x)$	67.26	56.98	77.38
$\log p_\theta(x)$ with BN	85.15	64.10	82.00
$\log \frac{p_\theta(x)}{p_\omega(x)}$	**99.08**	**99.85**	**99.81**

Table 1: Top table shows the average AUROC and AUPR of other past works on large-scale datasets. Bottom table shows the average AUROC of past works on large-scale datasets. $\log \frac{p_\theta(x)}{p_\omega(x)}$ is always outstanding, where ω is trained on OoD.

shown in Table 1, where DeConf-C, MCMC Recon, MCMC $\log p_\theta(x)$, $D_\theta(x)$, $\|\nabla_x D_\theta(x)\|$, entropy, $\max_y p(y|x)$, Mahalanobis, ODIN and disagreement are proposed by (Hendrycks and Gimpel 2016; Hsu et al. 2020; Lee et al. 2018; Alemi, Fischer, and Dillon 2018; Liang, Li, and Srikant 2018; Kumar, Goyal et al. 2019; Xu et al. 2018; Chen et al. 2019; Lakshminarayanan, Pritzel, and Blundell 2017). Perfect classifier is the classifier used to detect simply-classified, which is the theoretical optimal result. Thanking to the assistance of $L(x)$, $S(x)$ reaches the best performance among the past works, which encourages us to develop a better assistance in the next.

Observation of KL-based indicator

As shown in Figure 2, complexity estimate is unstable and sometimes it might lower the performance. Therefore, we try to find another function to replace $L(x)$ in $S(x)$. In experiments on large-scale datasets, we observe a common 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 likelihood-based model trained on out-of-distribution. The average AUROC of $\log p_\theta(x) - \log p_\omega(x)$ reaches nearly 100% on all datasets in Table 1. From the view of complexity estimate, $L(x) = -\log_2 p(x|\mathcal{M}_0)$ is the log-likelihood of a universal model $\mathcal{M}_0$ (Serrà et al. 2019). In our paper, $p_\omega(x)$ is used to replace $L(x)$ to assist log-likelihood.

However, in OoD problem, the out-of-distribution is not known. Therefore, we will explore how to develop an indicator that are not dependent on OoD and why $\log \frac{p_\theta(x)}{p_\omega(x)}$ is almost useful for OoD theoretically.

5 Theorem

Divergence-based Indicators

We propose a theoretical framework for **R**obust **O**oD **I**ndicators based on **D**ivergence called *ROID*, with following assumptions (also observed in experiments):

1. The training data and testing data of in-distribution and out-of-distribution are i.i.d. .

2. div is a divergence , 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 in p_{in}.

4. $f : \mathcal{X} \to \mathcal{R}$ maps the distribution p_{in}, p_{out} into two Gaussian distribution with constant variance.

5. Assumption 3 and 4 hold for any $\hat{f}$ approximating f.

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 $\log p_{in}(x) - \log p_{out}(x) \approx 0$ for x near 0. In such example, p_{in} and p_{out} is distinguishing by KL-divergence but is not simply classified because no classifier can detect whether a sample in $[-1, 1]$ is from p_{in} or p_{out}. We assume that assumption 3 holds when p_{in} and p_{out} is simply classified.

The above assumptions mean that if div can distinguishing the probability distribution in-distribution and out-of-distribution, then f can be used as an OoD indicator, called the div-based indicator.

In the next, we will introduce two concrete divergence-based indicators: KL-based and Wasserstein-based indicator, and show how to obtain them without knowing OoD.

KL-based Indicator

We choose KL-divergence as div, then $KL(p_{in}, p_{out}) = \mathbb{E}_{p_{in}(x)}[\log p_{in}(x) - \log p_{out}(x)]$ and KL-based indicator is $\log p_{in}(x) - \log p_{out}(x)$, whose motivation is in Figure 3.

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

Likelihood is not a symmetric indicator since it uses p_A as indicator in experiment A vs B and p_B as indicator in experiment B vs A. An interesting phenomenon is that if likelihood indicator reaches a good performance on experiment A vs B, it will usually fail in experiment B vs A, in appendix

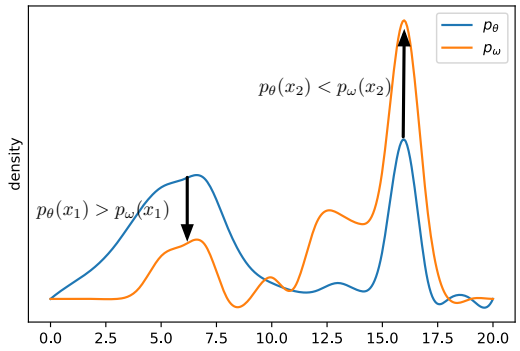


Figure 3: Diagrammatic sketch for KL-based indicator. The in-distribution is in [0, 10] and out-of-distribution in [10, 20]. Intuitively, p_{in} assign lower density to samples from in-distribution and higher density for samples from out-of-distribution than p_{out} . Different to the assumption of likelihood indicator, we do not assume that $p_\theta(x_2)$ very small for $x_2 \sim p_{out}$. By the assumptions in Section 5, we can induce that $p_\theta(x_1) > p_\omega(x_1)$ and $p_\theta(x_2) < p_\omega(x_2)$. Thus $\log p_\theta(x) - \log p_\omega(x)$ is an indicator for OoD detection.

B. Especially, in experiments of Noise vs any dataset, the optimal model for Noise dataset is uniform distribution, but it can not detect any OoD. In the experiment of any dataset vs Noise, likelihood reaches nearly 100% AUROC since the likelihood of noise is significantly lower than any datasets.

However, p_{out} is unknown in the 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 indicator $\log p_{in}(x) - \log p_{mix}(x)$ and indicator $\log p_{in}(x) - \log p_{out}(x)$ is equal for OoD detection.*

If we can get enough data from the mixture distribution, by Theorem 1, p_{mix} can be used to replace the term p_{out} in KL-based indicator and keep the same performance, which is shown in experiments in Table 2. Next, we try to explain why KL-based indicator can reach outstanding performance.

Compared to likelihood indicator and likelihood-ratio indicator, KL-based indicator can get better performance.

Theorem 3 *When log-likelihood can detect OoD, i.e., $\log p_{in}(x_1) > \log p_{in}(x_2)$ for almost $x_1 \sim p_{in}$ and $x_2 \sim p_{out}$, KL-based indicator can be also used to detect OoD.*

Theorem 4 *For any likelihood-ratio indicator $\log p_{in}(x) - \log g(x)$ where g is a continuous differentiable probability distribution, KL-based indicator outperforms them.*

Above theorems show that $\log \frac{p_{in}(x)}{p_{mix}(x)}$ is a good OoD indicator. It is a natural idea to model in-distribution and mixture distribution by p_θ, p_γ, *i.e.*, $\max_\theta \mathbb{E}_{p_{in}(x)} \log p_\theta(x)$ and $\max_\gamma \mathbb{E}_{p_{mix}(x)} \log p_\gamma(x)$ are used as loss to train model. Next, we will show the property of model distribution.

Theorem 5 $\log \frac{p_\theta(x)}{p_\gamma(x)}$ *can be used for detecting OoD. Moreover,* $\log \frac{p_\theta(x)}{p_\gamma(x)}$ *represents whether a sample x in mixture distribution have been optimized in the training process of θ.*

Theorem 5 shows that $\log \frac{p_\theta(x)}{p_\gamma(x)}$ represents the optimizability: if a sample x is OoD, $p_\theta(x)$ will be lower than $p_\gamma(x)$ since x is also in p_{mix} and $p_\gamma(x)$ can be optimized more than $p_\theta(x)$; if x is not OoD, $p_\theta(x)$ will not be lower than $p_\gamma(x)$ since $p_\theta(x)$ is well-trained to maximize log-likelihood. Theorem 5 indicates that training on mixture distribution does not need much time since we only need to observe whether the likelihood of a sample can be improved, which can be shown in few epochs. It alleviates the requirements for training and then we can train models online.

Wasserstein-based Indicator

From the view of traditional likelihood-indicator, GANs can not be used for detecting OoD since how to evaluate the density of GANs is an open problem(Nalisnick et al. 2019). Therefore, some researches propose variants of GANs with tractable density (Kumar, Goyal et al. 2019). In the framework of divergence-based indicators, we can detect OoD by discriminator 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 in 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, x_j^{out} is sampled from out-of-distribution and D is the optimal solution in $W^1(p_{in}, p_{out})$. Then the Wasserstein-based indicator for sample x is $D(x) - \sum_j D(x_j^{out})$. Note that $\sum_j D(x_j^{out})$ is constant, which do not affect the performance of OoD detection. Thus, the Wasserstein-based indicator is simply $D(x)$.

Next theorems are used to show why Wasserstein-based indicator can reach great performance ensure that D can be obtained without knowing 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 indicator.*

Theorem 8 *$\hat{D}$ that is optimal solution in $W^1(p_{in}, p_{mix})$ is same to the optimal solution D in $W^1(p_{in}, p_{out})$. Moreover, the neural networks trained by $W^1(p_{in}, p_{mix})$ and $W^1(p_{in}, p_{out})$ will share the same optimization process.*

By Theorem 8, $\hat{D}$ that is easy-obtained can replace D. Next theorem will ensure the performance of it.

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

Addressing concerns

The idea of this paper is training a model for OoD detection by exploiting the samples of mixture distribution. However, there are three major concerns about this idea:
1. Is our method **data-specific**? *i.e.*, does this method only work on the data that it has seen in training?
2. Can our method work **online**? *i.e.*, for new testing data, indicators should be given before the next testing data come.
3. Can our method work when p_{mix} has only **less samples**?
4. Is it better to train **pre-trained** models (trained on in-distribution) or **initialized** models on mixture distribution?

To address concerns, we design following experiments:
1. In experiment A vs B, only 20% data in mixture distribution can be used for training.
2. We simulate the online system — the data in mixture distribution is streaming and requires indicators immediately.
3. We splits the data from p_{mix} into several blocks and model can only use p_{in} and the block waiting for detection.
4. Experiments for pre-train models and initialized models.

6 Experiments

In this section, we will demonstrate the effectiveness of KL-based indicator and Wasserstein-based indicator on large-scale computer vision benchmark datasets.

Datasets

We consider following datasets: MNIST (LECUN et al. 1998), FashionMNIST (Xiao, Rasul et al. 2017), KMNIST (Clanuwat et al. 2018), NOT-MNIST, Omniglot (Lake, Salakhutdinov et al. 2015), CIFAR-10 (Krizhevsky, Hinton et al. 2009), CIFAR-100 (Krizhevsky, Hinton et al. 2009), TinyImagenet (Deng et al. 2009), SVHN (Netzer, Wang et al. 2011), iSUN (Xu et al. 2015), CelebA (Liu, Luo et al. 2015), LSUN (yu et al. 2015), Noise and Constant.

Natural images are resized to 32x32x3 and grey images are 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 intuitively they have similar classes. Noise and Constant includes both grey and natural images. LSUN and iSUN are only used for out-of-distribution. CelebA, Noise and Constant have no labels, and we set random labels from 0 to 9 on these datasets. We ended up with 92 dataset pairs.

Samples of p_{in}, p_{out} are from training set of p_{in}, p_{out}. Samples of p_{mix} are from the testing set of p_{in}, p_{out}.

Metrics

Following standard 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, a threshold-independent metric (Davis and Goadrich 2006) and widely used in OoD domain.
AP is the Average Precision, summarizing the precision-recall curve as the weighted mean of precisions.
FPR@TPR95 is the False Positive Rate when True Positive Rate is over 95%., which means the probability that an out-of-distribution example is misclassified as in-distribution when over 95% in-distribution is detected accurately.
AUPR is the Area under the Precision-Recall curve, which is also threshold independent (Saito and Rehmsmeier 2015).

Setups

For fair comparison, all indicators are based on common models with standard training, including 34-layer ResNet, VAE, PixelCNN, RealNVP and Wasserstein GAN. ResNet34 is trained for classification and serves for the OoD indicators based on classifier. ResNet34 is also trained to classify in-distribution and out-of-distribution for validating whether they are simply classified. Deep generative models are trained as their proposers suggest.

In our experiments, there is no validation set. For indicators 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 on all datasets, it is forbidden to specify the hyper-parameters or architectures on special dataset. The detailed architecture and parameters are shown in appendix B.

Major Results

indicator	Model	AUROC	AUPR
$\log \frac{p_\theta(x)}{p_\omega(x)}$	VAE	99.08	99.03
$\log \frac{p_\theta(x)}{p_\omega(x)}$	PixelCNN	99.85	99.82
$\log \frac{p_\theta(x)}{p_\omega(x)}$	RNVP	99.81	99.77
$D_\omega(x)$	WGAN	98.57	98.40
$\log \frac{p_\theta(x)}{p_\gamma(x)}$	VAE	98.44	98.31
$\log \frac{p_\theta(x)}{p_\gamma(x)}$	PixelCNN	97.23	95.91
$\log \frac{p_\theta(x)}{p_\gamma(x)}$	RNVP	97.99	97.40
$D_\gamma(x)$	WGAN	98.06	98.20

Table 2: The performance of divergence-based Indicators. γ, ω are trained on p_{mix} and p_{out} respectively.

The main results of the previous works are summarized in Table 1. The main results for KL-based indicator and Wasserstein-based indicator are summarized in Table 2.

Validation of Theorem

Theorem 1 is supported by the detailed experiments shown in appendix B. Theorem 2 is supported by Table 2, where the performance of $\log \frac{p_\theta(x)}{p_\gamma(x)}$ is nearly same as $\log \frac{p_\theta(x)}{p_\omega(x)}$.

Detailed experiments in appendix B show that KL-based indicator can get same performance when log-likelihood can reach outstanding performance, to support Theorem 3.

We compare the precision-recall curve and ROC curve of KL-based indicator, log-likelihood indicator, likelihood ratio indicator and log-likelihood with input complexity in Figure 4, which shows KL-based indicator is the best version of log-likelihood ratio indicator, to support Theorem 4.

Theorem 7 is supported by the detailed experiments in appendix B. In Table 2, the performance of $D_\omega(x)$ is nearly same as $D_\gamma(x)$, which supports Theorem 8. The outstanding performance of $D_\omega(x), D_\gamma(x)$ supports the Theorem 9.

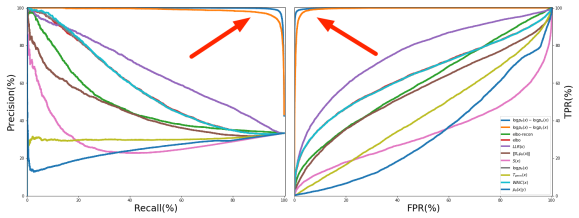


Figure 4: ROC and PRC on CIFAR-100 vs CelebA based on VAE model. The KL-based indicator outperforms others.

Concerns

Limit	indicator	Model	AUROC	AUPR
online	$\log \frac{p_\theta(x)}{p_\gamma(x)}$	VAE	97.24	95.92
online	$\log \frac{p_\theta(x)}{p_\gamma(x)}$	PixelCNN	91.77	90.14
online	$\log \frac{p_\theta(x)}{p_\gamma(x)}$	RNVP	90.92	90.89
online	$D_\gamma(x)$	WGAN	97.96	98.16
20%	$\log \frac{p_\theta(x)}{p_\theta(x)}$	VAE	96.50	94.97
20%	$\log \frac{p_\theta(x)}{p_\gamma(x)}$	PixelCNN	91.47	90.83
20%	$\log \frac{p_\theta(x)}{p_\gamma(x)}$	RNVP	88.16	88.84
20%	$D_\gamma(x)$	WGAN	98.09	98.22
block	$\log \frac{p_\theta(x)}{p_\gamma(x)}$	VAE	97.60	96.26
block	$\log \frac{p_\theta(x)}{p_\gamma(x)}$	PixelCNN	88.80	88.96
block	$\log \frac{p_\theta(x)}{p_\gamma(x)}$	RNVP	90.44	90.83
block	$D_\gamma(x)$	WGAN	97.66	97.89

Table 3: The average online testing time (including training time on p_{out}) per image for VAE, PixelCNN, RNVP and WGAN is 0.18s, 0.13s, 0.58s and 0.11s. 'online' indicates that data is streaming (4096 data is coming at same time), '20%' indicates that only 20% data in p_{mix} can be used for training and 'block' indicates that only a block (including 4096 data) can be used. VAE and WGAN are more robust since their performance is near to the value in table 2.

The experiments about concerns are shown in Table 3. Figure 5 shows the change of AUROC during training on p_{mix}. Pretrained model can easily get a good performance after few epochs, but it will fall into the local extremum (likelihood on in-distribution has been optimized quite well in pre-training, and any modification for parameters will reduce it) and the final performance is not well. Un-pretrained model need more epochs to get a good performance, but its final performance is better than pretrained model. Figure 5 shows the change of AUROC while the size of block varies. It shows the limitation: when block size is small enough, the optimization only on one block will lead to unexpected p_γ. Training on past data (online) can alleviate this issue. Block size of 'online' means the size of data streaming at same time. These experiments show that KL-based indicator and Wasserstein-based indicator are effective, robust, general, online and not data-specific. It also shows the weakness of that they can not work well in a small enough block.

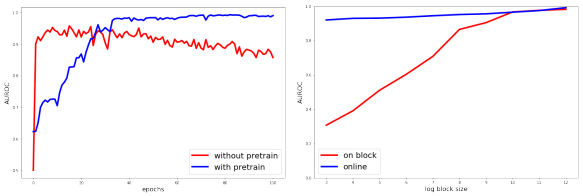


Figure 5: Left: the changing of average AUROC of pretrained model and un-pretrained model during the training on p_{mix}, on CIFAR-10 vs other datasets. Right: the changing of AUROC of KL-based indicator when the block size varies on CIFAR-10 vs SVHN.

Limitations of this Study

Limitation of datasets. In our paper, we use large-scale datasets to show the generality of indicators. However, we only consider natural OoD datasets and do not consider attacked OoD, which are categorized by (Chen et al. 2020). An important reason is that there is not a universal principle like simply classified, to measure these attacked OoD datasets.

Limitation of models. In our paper, for fair comparison and generality, we only consider the common models, including ResNet, VAE, PixelCNN, RealNVP and WGAN. However, there are numerous models careful-designed for OoD detection. Due to the resource limitation, we can not give the performance of them on large-scale datasets. We emphasize that small-scale experiments are unreliable and large-scale experiments are mandatory for evaluating OoD indicators.

Limitation of divergence-based indicators. Some defects of divergence-based indicators is shown in section 6. They rely on the model (*e.g.*, KL-based indicator on RNVP is more data-specific) and optimizer (*e.g.*, when data from p_{mix} is not enough, optimizer can not give the expected p_γ). It is a natural idea to approximate $\log p_\theta(x) - \log p_\gamma(x)$ without optimizer, *e.g.*, Taylor Expansion on pretrained model. We leave it as future work. Divergence-based indicators will guide future works since it can develop simply, fundamental and practical indicators from basic assumptions.

7 Conclusion and Future Work

In this paper, we first show through large-scale experiments that none of existing OoD indicators based on deep generative models is robust. We then propose a novel theoretical framework *ROID* for robust Out-of-Distribution indictors based on divergence in deep generative models (instead of the 10 traditional likelihood assumption), and propose and prove the KL-based indicator and the Wasserstein-based indictor, both of which significantly outperform past works by 10% in AUROC.

We believe our paper is an important step towards developing more robust OoD indicators based on deep generative models. For future work, it will be interesting to propose more OoD indicators following *ROID* framework. It would be also interesting to approximate KL-based indicator without optimizer or develop optimizers that only require few data to train models on mixture distribution, for solving the limitation of divergence-based indicators.

8 Ethical and societal impact

There is no any potential ethical impact of our work. Our work focus on pure model research. The datasets and model considered in our paper are open, clear and popular. Our work concentrate on developing new OoD detector based on deep generative models, which might play important role in anomaly detection or assisting the classification. Our paper does not have any negative societal implications.

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.

Bishop, C. M. 1994. Novelty detection and neural network validation. *IEE Proceedings-Vision, Image and Signal processing* 141(4): 217–222.

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* .

Chen, J.; Li, Y.; Wu, X.; Liang, Y.; and Jha, S. 2020. Robust Out-of-distribution Detection via Informative Outlier Mining. *arXiv preprint arXiv:2006.15207* .

Chen, W.; Xu, H.; Li, Z.; Peiy, D.; Chen, J.; Qiao, H.; Feng, Y.; and Wang, Z. 2019. Unsupervised anomaly detection for intricate kpis via adversarial training of vae. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, 1891–1899. IEEE.

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

Clanuwat, T.; Bober-Irizar, M.; Kitamoto, A.; Lamb, A.; Yamamoto, K.; and Ha, D. 2018. Deep Learning for Classical Japanese Literature.

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.

Deng, J.; Dong, W.; Socher, R.; Li, L.-j.; Li, K.; and Li, F.-f. 2009. ImageNet: A large-scale hierarchical image database. *CVPR* 248–255.

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

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

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

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.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Technical report, Citeseer.

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

Lake, B. M.; Salakhutdinov, R.; et al. 2015. Human-level concept learning through probabilistic program induction. *Science* 350(6266): 1332–1338.

Lakshminarayanan, B.; Pritzel, A.; and Blundell, C. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. *neural information processing systems* .

LECUN, Y.; BOTTOU, . E. L.; BENGIO, Y.; and HAFFNER, P. 1998. Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE* 2278–2324.

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.

Liang, S.; Li, Y.; and Srikant, R. 2018. Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks. *international conference on learning representations* .

Liu, Z.; Luo, P.; et al. 2015. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, 3730–3738.

Nalisnick, T. E.; Matsukawa, A.; Teh, W. Y.; Grr, D.; and Lakshminarayanan, B. 2019. Do Deep Generative Models Know What They Don't Know? *international conference on learning representations* .

Netzer, Y.; Wang, T.; et al. 2011. Reading digits in natural images with unsupervised feature learning .

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* .

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. 2016. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261* .

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.

Xiao, H.; Rasul, K.; et al. 2017. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms.

Xu, H.; Chen, W.; Zhao, N.; Li, Z.; Bu, J.; Li, Z.; Liu, Y.; Zhao, Y.; Pei, D.; Feng, Y.; et al. 2018. Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. In *Proceedings of the 2018 World Wide Web Conference*, 187–196.

Xu, P.; Ehinger, A. K.; Zhang, Y.; Finkelstein, A.; Kulkarni, R. S.; and Xiao, J. 2015. TurkerGaze: Crowdsourcing Saliency with Webcam based Eye Tracking. *CoRR* .

yu, f.; zhang, y.; song, s.; seff, a.; and xiao, j. 2015. LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop. *CoRR* .

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146* .