A Counter-examples

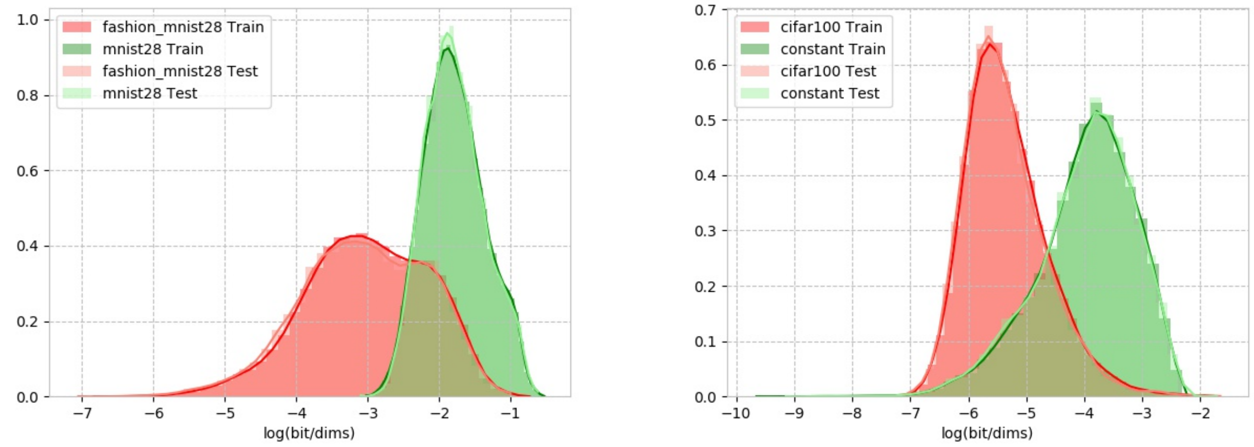

Figure 1: Counter example of $\log p_\theta(x)$ with AUROC 0.0609 and 0.1104

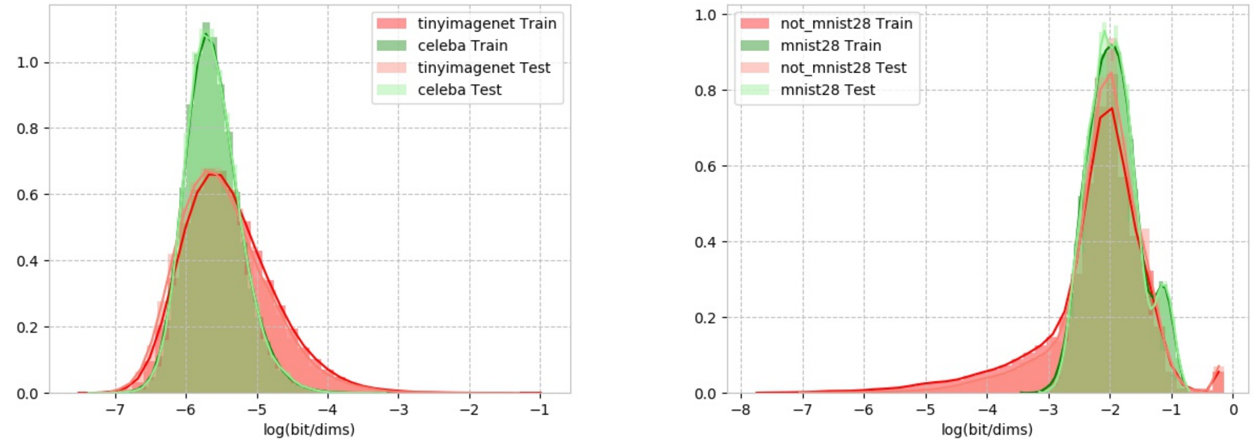

Figure 2: Counter example of $T_{perm}(x)$ with AUROC 0.3933 and 0.4692

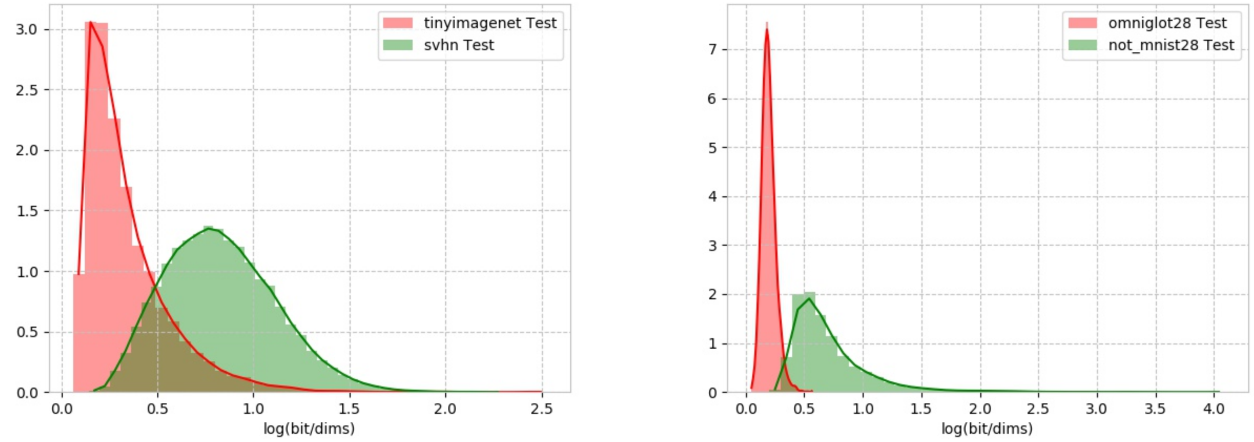

Figure 3: Counter example of $\|\nabla_x \log p_\theta(x)\|$ with AUROC 0.0196 and 0.0023

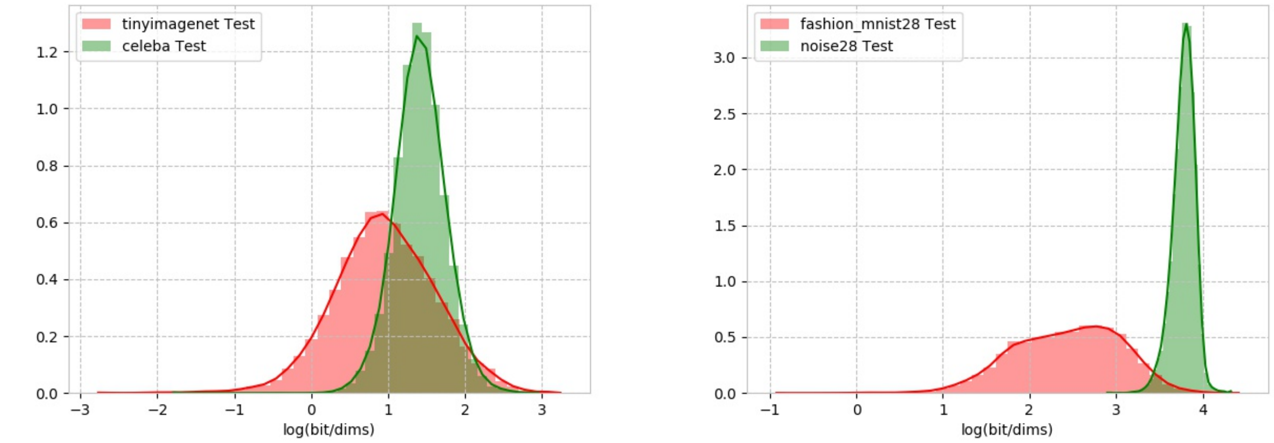

Figure 4: Counter example of $S(x)$ with AUROC 0.2636 and 0.0062

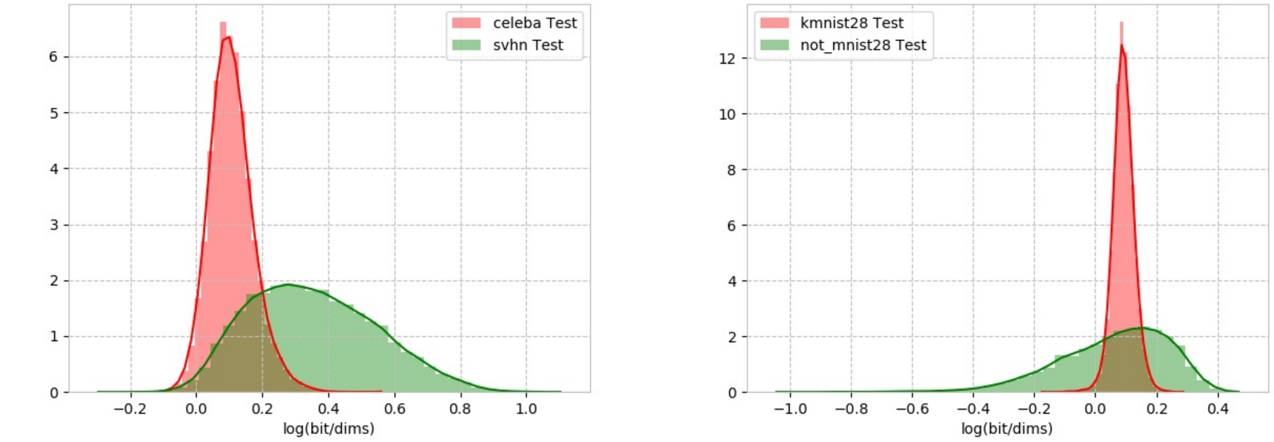

Figure 5: Counter example of $LLR(x)$ with AUROC 0.1054 and 0.5143

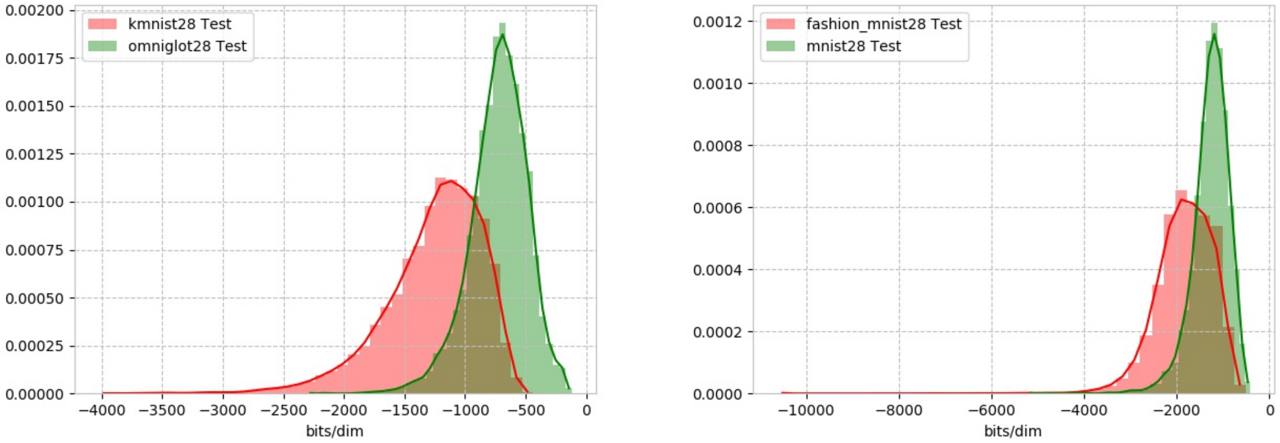

Figure 6: Counter example of $WAIC(x)$ with AUROC 0.1041 and 0.2135

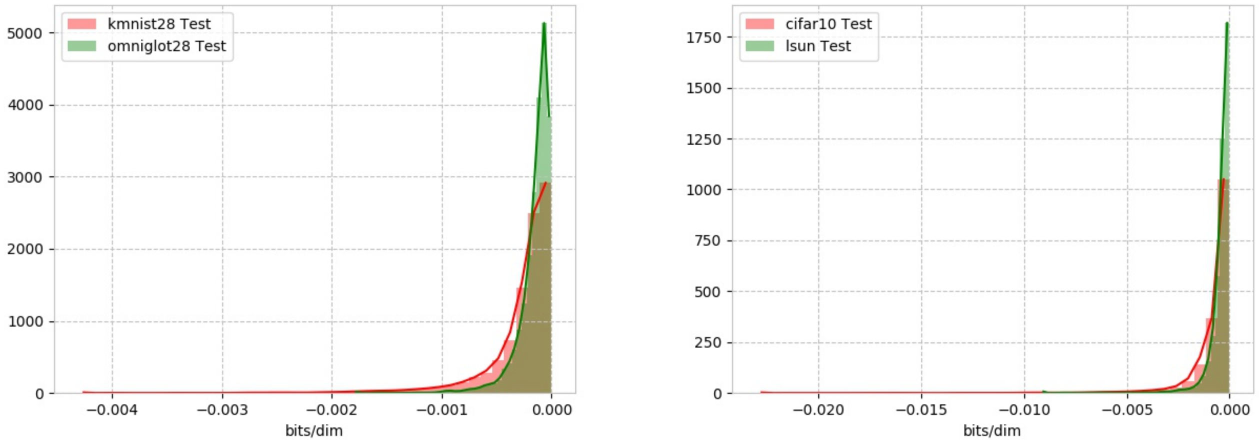

Figure 7: Counter example of $VAR_\theta \log p_\theta(x)$ with AUROC 0.5358 and 0.3756

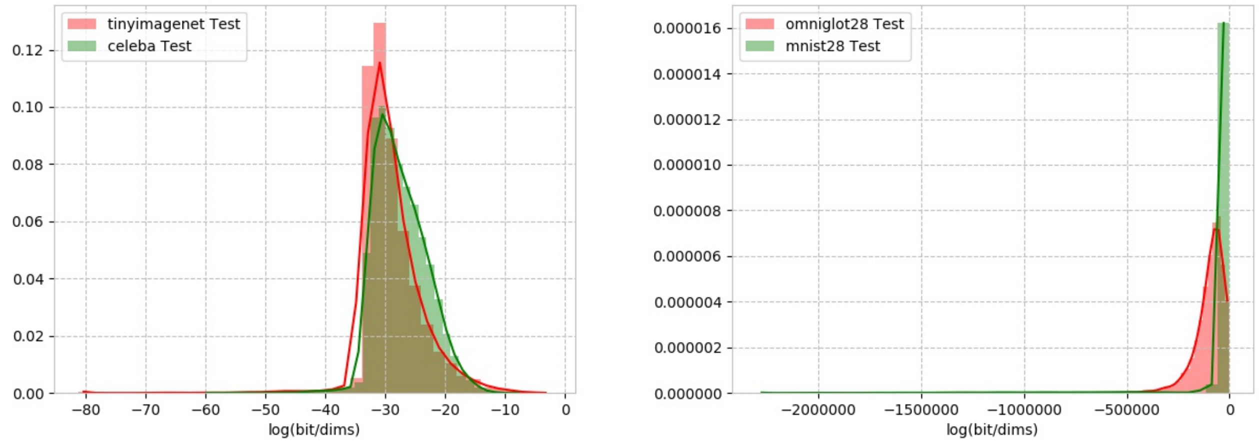

Figure 8: Counter example of $\log p_\theta(x|y)$ with AUROC 0.3822 and 0.1150

B Experiments

B.1 Datasets

The following datasets are considered: MNIST [6], FashionMNIST [10], KMNIST [1], NOT-MNIST, Omniglot [5], CIFAR-10 [4], CIFAR-100 [4], TinyImagenet [3], SVHN [8], iSUN [11], CelebA [7], LSUN [12], 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. LSUN and iSUN are only used as out-of-distribution. CelebA, Noise and Constant have no labels, and we set random labels from 0 to 9 on them. We end up with 92 dataset pairs.

B.2 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 [2] 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 [9].

B.3 Setups

For fair comparison, all indicators are based on common models with standard training as their proposers suggest, including ResNet34, VAE, PixelCNN, RealNVP. ResNet34 validates whether two datasets are simply-classified and serves for the OoD indicators based on classifier.

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.

Our experiments are running on 18 Nvidia GTX 2080Ti of 3 GPU servers. The detailed setup is described in 'supplemental/code/README'.

B.4 Architecture

The architecture of VAE is:

Layer	Channel	Stride	Kernel	Activation
Resnet	16	1x1	3x3	Leaky ReLU
Resnet	32	1x1	3x3	Leaky ReLU
Resnet	64	1x1	3x3	Leaky ReLU
Resnet	128	2x2	3x3	Leaky ReLU
Resnet	128	2x2	3x3	Leaky ReLU
Resnet	128	2x2	3x3	Leaky ReLU
Flatten				
Dense	256			None

Table 1: Encoder architecture of VAE

Before the decoder, we will use dense layer to map z to a tensor with shape $8hw$ and reshape $(h/4, w/4, 128)$ where h, w is the height and width of data.

Layer	Channel	Stride	Kernel	Activation
Resnet Deconv	128	2x2	3x3	Leaky ReLU
Resnet Deconv	128	2x2	3x3	Leaky ReLU
Resnet Deconv	128	2x2	3x3	Leaky ReLU
Resnet Deconv	64	1x1	3x3	Leaky ReLU
Resnet Deconv	32	1x1	3x3	Leaky ReLU
Resnet Deconv	16	1x1	3x3	Leaky ReLU
Conv2d	3 or 1	1x1	1x1	None

Table 2: Decoder architecture of VAE

We apply a Discretized Logistic Distribution as $p_\theta(x|z)$ and gaussian as $q_\phi(z|x)$.
The architecture of PixelCNN is:

Layer	Channel	Vertical Kernel	Horizontal Kernel	Activation	Dropout
Resnet	64	2x3	2x2	Leaky ReLU	0.2
Resnet	64	2x3	2x2	Leaky ReLU	0.2
Resnet	64	2x3	2x2	Leaky ReLU	0.2
Resnet	64	2x3	2x2	Leaky ReLU	0.2
Resnet	64	2x3	2x2	Leaky ReLU	0.2
Resnet	256	2x3	2x2	Leaky ReLU	0.2
Flatten					
Dense	256			None	

Table 3: Architecture of PixelCNN

The architecture of Wasserstein is:

Data-augmentation	AUROC
Basic	95.2470
+Flip	95.3480
+Crop	95.1940
+Blur	95.2464
+Noise	95.4416
+Contrast	95.2464
+Light	95.2680
+Scale	95.3076

Table 6: Ablation study about data-augmentations.

Layer	Channel	Stride	Kernel	Activation
Resnet Deconv	128	2x2	3x3	Leaky ReLU
Resnet Deconv	128	2x2	3x3	Leaky ReLU
Resnet Deconv	128	1x1	3x3	Leaky ReLU
Resnet Deconv	64	1x1	3x3	Leaky ReLU
Resnet Deconv	32	1x1	3x3	Leaky ReLU
Resnet Deconv	16	1x1	3x3	Leaky ReLU
Conv2d	1	1x1	1x1	None

Table 4: Generator architecture of WGAN

Layer	Channel	Stride	Kernel	Activation
Resnet	16	1x1	3x3	Leaky ReLU
Resnet	32	1x1	3x3	Leaky ReLU
Resnet	64	1x1	3x3	Leaky ReLU
Resnet	128	1x1	3x3	Leaky ReLU
Resnet	128	2x2	3x3	Leaky ReLU
Resnet	128	2x2	3x3	Leaky ReLU

Table 5: Discriminator architecture of WGAN

RealNVP has 3 level and 6 blocks at each level. Channel is 128 and activation is ReLU in each block. The average runtime memory of VAE, PixelCNN, WGAN and Glow are 2.949GB, 2.851GB, 2.826GB and 2.917GB.

B.5 Data-augmentation and steps

In single-shot fine-tune, the image is randomly shifted in -10% to 10% vertically and horizontally, and rotated in -60 to 60 degree, as basic data-augmentation.

The other data-augmentations, containing shifting, rotation, blur, noise, scale, cropping, flipping and modifying contrast and lightness are shown in the following. As can be seen, their performance has no significant difference ($\leq 0.3\%$) to the basic data-augmentation containing rotation and shift. If do not use data-augmentation, fine-tuning is easy to fail (*e.g.*, some parameters becomes NaN during fine-tuning).

In single-shot fine-tune, the number of steps influence the performance, as shown in the following.

C Theorem

We will derive the theorems in our papers by the assumptions:

1. $D_{in}^{train}, D_{in}^{test}$ are i.i.d. from distribution p_{in} and $D_{out}^{test}, D_{out}^{train}$ are i.i.d. from distribution p_{out}.

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 most 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.

Lemma 1. *Indicators f_0 and f_1 satisfying that if $f_0(x_1) < f_0(x_2)$ then $f_1(x_1) < f_1(x_2)$ and if $f_0(x_1) > f_0(x_2)$ then $f_1(x_1) > f_1(x_2)$ for x_1, x_2 in mixture distribution, have same performance for OoD. We call $f_0 \triangleq f_1$.*

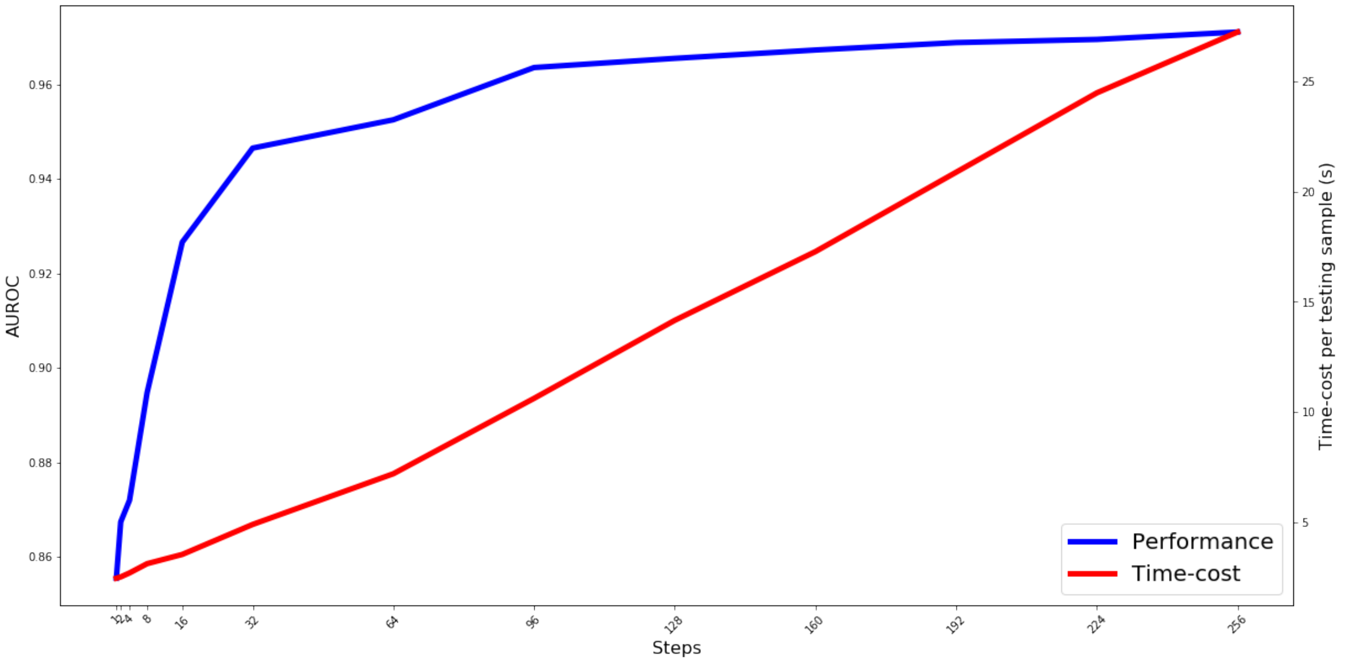

Figure 9: The AUROC when step is from 1 to 256. When step is less than 64, step significant influence the AUROC, and when step is larger than 64, AUROC is nearly same. Therefore, we set step = 64 as the default parameter of single-shot fine-tune. Additional, when step =64, the time-cost is only 7s per testing sample.

Proof. Assume we get data $x_0, x_1, \ldots, x_n$ from mixture distribution. The metrics AUROC, AUPR, FPR@TPR95 and AP for indicator f are all dependent on the order of $f(x_0), f(x_1), \ldots, f(x_n)$. The array $f_0(x_0), f_0(x_1), \ldots, f_0(x_n)$ and $f_1(x_0), f_1(x_1), \ldots, f_1(x_n)$ have the same order and then they have same performance for all metrics considered in OoD.

Theorem 1 *$\log p_{in}(x) - \log p_{out}(x)$ and $\log p_\theta(x) - \log p_\omega(x)$ and are effective symmetric indicators, i.e., it reaches same performance in experiment A vs B and B vs A, with threshold 0, where $p_\theta \to p_{in}$ and $p_\omega \to p_{out}$. $\log p_\theta(x)$ maps p_{in} into a gaussian distribution.*

Proof. We select KL as div. By assumption 3 and 4, we have

$$\log p_{in}(x_1) - \log p_{out}(x_1) \gg 0 \quad \textit{and} \quad \log p_{in}(x_2) - \log p_{out}(x_2) \ll 0$$

where $x_1 \sim p_{in}$ and $x_2 \sim p_{out}$. By assumption 5, above equation holds for $\log p_\theta(x) - \log p_\omega(x)$. Therefore, $\log p_{in}(x) - \log p_{out}(x)$ and $\log p_\theta(x) - \log p_\omega(x)$ can detect most of OoD, with threshold zero.

In experiment A vs B and B vs A, the indicator is $\log p_A(x) - \log p_B(x)$ and $\log p_B(x) - \log p_A(x)$. These two experiments have same mixture distribution. For any testing data x_1, x_2 in mixture distribution, if $\log p_A(x_1) - \log p_B(x_1) > \log p_A(x_2) - \log p_B(x_2)$, then $\log p_B(x_1) - \log p_A(x_1) < \log p_B(x_2) - \log p_A(x_2)$. Conversely, if $\log p_A(x_1) - \log p_B(x_1) < \log p_A(x_2) - \log p_B(x_2)$, then $\log p_B(x_1) - \log p_A(x_1) > \log p_B(x_2) - \log p_A(x_2)$. Thus, they have inverse order. Noting that the positive and negative labels in experiment A vs B and B vs A are also inverse, $\log p_{in}(x) - \log p_{out}(x)$ reaches same performance in experiment A vs B and B vs A. The proof for $\log p_\theta(x) - \log p_\omega$ is the same as above.

Let $f(x) = \log p_{in}(x)$ and $\hat{f}(x) = \log p_\theta(x)$. Since $p_\theta \to p_{in}$, $\hat{f} \to f$. By assumption 4,5, we know $\hat{f} = \log p_\theta(x)$ maps p_{in} into a gaussian distribution.

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.*

Proof. For any $x1, x2$ satisfying $\log p_{in}(x_1) - \log p_{out}(x_1) < \log p_{in}(x_2) - \log p_{out}(x_2)$, we have $\log p_{in}(x_1) - \log p_{mix}(x_1) <$

$\log p_{in}(x_2) - \log p_{mix}(x_2)$ by

$$\log p_{in}(x) - \log p_{mix}(x) = -\log \frac{\alpha p_{in}(x) + \beta p_{out}(x)}{p_{in}(x)} = -\log\left(\alpha + \beta \frac{p_{out}(x)}{p_{in}(x)}\right)$$
$$\log \frac{p_{in}(x_1)}{p_{out}(x_1)} < \log \frac{p_{in}(x_2)}{p_{out}(x_2)} \Rightarrow \frac{p_{in}(x_1)}{p_{out}(x_1)} < \frac{p_{in}(x_2)}{p_{out}(x_2)} \Rightarrow \frac{p_{out}(x_1)}{p_{in}(x_1)} > \frac{p_{out}(x_2)}{p_{in}(x_2)}$$
$$\Rightarrow \log\left(\alpha + \beta\frac{p_{out}(x_1)}{p_{in}(x_1)}\right) > \log\left(\alpha + \beta\frac{p_{out}(x_2)}{p_{in}(x_2)}\right)$$
$$\Rightarrow -(\log p_{in}(x_1) - \log p_{mix}(x_1)) > -(\log p_{in}(x_2) - \log p_{mix}(x_2))$$
$$\Rightarrow \log \frac{p_{in}(x_1)}{p_{mix}(x_1)} < \log \frac{p_{in}(x_2)}{p_{mix}(x_2)}$$

By the same way, for any $x1, x2$ satisfying $\log p_{in}(x_1) - \log p_{out}(x_1) > \log p_{in}(x_2) - \log p_{out}(x_2)$, we have $\log p_{in}(x_1) - \log p_{mix}(x_1) > \log p_{in}(x_2) - \log p_{mix}(x_2)$. By **??**, 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.
Remark. In the proof, we do not use the condition that $\alpha > 0$ but only $\beta > 0$ and $p_{mix}(x) > 0$. It means that theorem 2 holds even when $\alpha < 0, \beta > 0$ and $p_{mix}(x) > 0$.

Theorem 3 *On any dataset pair that log-likelihood works well, i.e.,* $\log p_{in}(x_1) > \log p_{in}(x_2)$ *for most* $x_1 \sim p_{in}, x_2 \sim p_{out}$*, KL-based indicator can reach better performance.*

Proof. Noting that the AUROC is related to the Mann–Whitney U, which tests whether positives are ranked higher than negatives, whose number called distortion. For each $x_1 \sim p_{in}, x_2 \sim p_{out}$, assume $\log p_{out}(x_1) < \log p_{out}(x_2)$, then $\log p_{in}(x_1) - \log p_{out}(x_1) > \log p_{in}(x_2) - \log p_{out}(x_2)$. KL-based indicator will get same performance than log-likelihood indicator.

Assume $\log p_{out}(x_1) > \log p_{out}(x_2)$, then likelihood indicator will make mistake in experiment B vs A (if current experiment is A vs B). Meanwhile KL-based indicator might detect OoD in both A vs B and B vs A by theorem 1. In conclusion, likelihood leads to more distortion than KL-based indicator, in experiment B vs A and experiment A vs B.

It means KL-based indicator is always better than log-likelihood indicator.

For the proofing of following theorems, we need an additional mathematical simplification: by assumption 4 and 5, the indicators for OoD maps the distribution p_{in}, p_{out} into two Gaussian distribution with constant variance, therefore, the AUROC between these two Gaussian distribution is dependent on the difference of their mean, *i.e.*, $\mathbb{E}_{p_{in}(x)} f(x) - \mathbb{E}_{p_{out}(x)} f(x)$ is our simplified metric for mathematical inference.

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.*

Proof. Consider the following optimization with subsidiary conditions for searching the optimal g for likelihood ratio:

$$\max_g \left\{ \mathbb{E}_{p_{in}(x)} \left[\log p_{in}(x) - \log g(x)\right] - \mathbb{E}_{p_{out}(x)} \left[\log p_{in}(x) - \log g(x)\right] \right\}$$
$$\textbf{s.t.} KL(p_{in}, g) = L \text{ and } \int g(x) \mathrm{d}x = 1$$

where L is a constraint to the optimization domain of g. It could be solved by Lagrange multiplier method introduced by calculus of variation. Let $\lambda, \psi > 0$ be Lagrange Multiplier for the two constraint. The Lagrange function is

$$\begin{aligned} J[g] &= \mathbb{E}_{p_{in}(x)} \left[\log p_{in}(x) - \log g(x)\right] - \mathbb{E}_{p_{out}(x)} \left[\log p_{in}(x) - \log g(x)\right] \\ &- \lambda KL(p_{in}, g) - \psi \int g(x)\mathrm{d}x \\ &= \int p_{in}(x) \log p_{in}(x) - p_{in}(x) \log g(x) - p_{out}(x) \log p_{in}(x) \\ &+ p_{out}(x) \log g(x) - \lambda p_{in}(x) \log \frac{p_{in}(x)}{g} - \psi g(x) \mathrm{d}x = \int F(g, x)\mathrm{d}x \end{aligned}$$

By calculus of variation, the extremal point for $J[g]$ satisfies that $\delta J = 0$, which is equal to

$$\begin{aligned} \frac{\partial F(g,x)}{\partial g} = 0 &\Rightarrow -\frac{p_{in}(x)}{g(x)} + \frac{p_{out}(x)}{g(x)} + \frac{\lambda p_{in}(x)}{g(x)} - \psi = 0 \\ &\Rightarrow g(x) = \frac{1}{\psi}((\lambda - 1)p_{in}(x) + p_{out}(x)) \end{aligned}$$

Considering the subsidiary condition $\int g(x)\mathrm{d}x = 1$, we have

$$\int g(x)\mathrm{d}x = \frac{1}{\psi}\int (\lambda-1)p_{in}(x) + p_{out}(x)\mathrm{d}x = \frac{1}{\psi}((\lambda-1)+1) = \frac{\lambda}{\psi} = 1$$

Therefore, $\lambda = \psi$ and $g^*(x) = \frac{\lambda-1}{\lambda}p_{in}(x) + \frac{1}{\lambda}p_{out}(x)$ is the extremal point. By the remark of theorem 1, $\log p_{in}(x) - \log g^*(x)$ gets same performance as $\log p_{in}(x) - \log p_{out}(x)$ in OoD detection. For any L, we can get $g^*(x)$, which has same performance as $\log p_{in}(x) - \log p_{out}(x)$. It means that KL-based indicator is the optimal indicator among all likelihood-ratio indicators.

Theorem 5 $\log\frac{p_\theta(x)}{p_\gamma(x)}$ *can reach better AUROC than KL-based indicator, when p_γ is well-trained, i.e., p_γ reaches better likelihood on p_{mix} than $p_{\hat{\gamma}} = \alpha p_\theta + \beta p_\omega$.*

Proof. Let $p_{\hat{\gamma}}(x) = \alpha p_\theta(x) + \beta p_\omega(x)$. $\hat{\gamma}$ is the parameters that can be obtained by optimizer. Considering the indicator $\log p_\theta(x) - \log p_{\hat{\gamma}}(x) = -\log(\alpha + \beta\frac{p_\omega(x)}{p_\theta(x)})$, for $x_1 \sim p_{in}$ and $x_2 \sim p_{out}$, we have

$$\log\frac{p_\theta(x_1)}{p_{\hat{\gamma}}(x_1)} - \log\frac{p_\theta(x_2)}{p_{\hat{\gamma}}(x_2)} = \log(\alpha + \beta\frac{p_\omega(x_2)}{p_\theta(x_2)}) - \log(\alpha + \beta\frac{p_\omega(x_1)}{p_\theta(x_1)})$$

We have known that $\log\frac{p_\theta(x_1)}{p_\omega(x_1)} \gg 0$ and $\log\frac{p_\theta(x_2)}{p_\omega(x_2)} \ll 0$ by theorem 1. Then, we have $\log\frac{p_\theta(x_1)}{p_{\hat{\gamma}}(x_1)} - \log\frac{p_\theta(x_2)}{p_{\hat{\gamma}}(x_2)} > 0$. It means that $\log\frac{p_\theta(x)}{p_{\hat{\gamma}}(x)}$ can be used for detecting OoD.

Next, consider the γ is optimized better than $\hat{\gamma}$, *i.e.*, $\mathbb{E}_{p_{mix}}\log p_\gamma(x) \geq \mathbb{E}_{p_{mix}}\log p_{\hat{\gamma}}(x)$.

$$\mathbb{E}_{p_{mix}}\log p_\gamma(x) = \alpha\,\mathbb{E}_{p_{in}}\log p_\gamma(x) + \beta\,\mathbb{E}_{p_{out}}\log p_\gamma(x)$$

By $\mathbb{E}_{p_{mix}}\log p_\gamma(x) \geq \mathbb{E}_{p_{mix}}\log p_{\hat{\gamma}}(x)$, there are 3 case:

1. $\mathbb{E}_{p_{in}}\log p_\gamma(x) \leq \mathbb{E}_{p_{in}}\log p_{\hat{\gamma}}(x)$ and $\mathbb{E}_{p_{out}}\log p_\gamma(x) \geq \mathbb{E}_{p_{out}}\log p_{\hat{\gamma}}(x)$.

2. $\mathbb{E}_{p_{in}}\log p_\gamma(x) \geq \mathbb{E}_{p_{in}}\log p_{\hat{\gamma}}(x)$ and $\mathbb{E}_{p_{out}}\log p_\gamma(x) \geq \mathbb{E}_{p_{out}}\log p_{\hat{\gamma}}(x)$.

3. $\mathbb{E}_{p_{in}}\log p_\gamma(x) \geq \mathbb{E}_{p_{in}}\log p_{\hat{\gamma}}(x)$ and $\mathbb{E}_{p_{out}}\log p_\gamma(x) \leq \mathbb{E}_{p_{out}}\log p_{\hat{\gamma}}(x)$.

By assumption 4, 5 and simplified metric, we know that if $\mathbb{E}_{p_{in}}\log p_\gamma(x)$ increase, the KL-based indicator $\log\frac{p_\theta(x)}{p_\gamma(x)}$ will be worse and if $\mathbb{E}_{p_{out}}\log p_\gamma(x)$ increase, the KL-based indicator $\log\frac{p_\theta(x)}{p_\gamma(x)}$ will be better.

Therefore, in case 1, the performance of $\log\frac{p_\theta(x_1)}{p_\gamma(x_1)}$ will be better than $\log\frac{p_\theta(x_1)}{p_{\hat{\gamma}}(x_1)}$. In case 2 and 3, we know $\mathbb{E}_{p_{in}}\log p_{\hat{\gamma}}(x) \leq \mathbb{E}_{p_{in}}\log p_\theta(x)$ and $\mathbb{E}_{p_{in}}\log p_\gamma(x) \leq \mathbb{E}_{p_{in}}\log p_\theta(x)$ since θ is well-trained for such loss function.

$$\begin{aligned}\beta\,\mathbb{E}_{p_{out}}\log p_\gamma(x) &= \mathbb{E}_{p_{mix}}\log p_\gamma(x) - \alpha\,\mathbb{E}_{p_{in}}\log p_\gamma(x)\\ &\geq \mathbb{E}_{p_{mix}}\log p_{\hat{\gamma}}(x) - \alpha\,\mathbb{E}_{p_{in}}\log p_\theta(x)\\ &= \alpha\,\mathbb{E}_{p_{in}}\log\frac{p_{\hat{\gamma}}(x)}{p_\theta(x)} + \beta\,\mathbb{E}_{p_{out}}\log p_{\hat{\gamma}}(x)\\ &\Rightarrow \mathbb{E}_{p_{out}}\log\frac{p_\gamma(x)}{p_{\hat{\gamma}}(x)} \geq \frac{\alpha}{\beta}\,\mathbb{E}_{p_{in}}\log\frac{p_{\hat{\gamma}}(x)}{p_\theta(x)}\end{aligned}$$

And we have

$$\mathbb{E}_{p_{in}}\log\frac{p_\gamma(x)}{p_{\hat{\gamma}}(x)} \leq \mathbb{E}_{p_{in}}\log\frac{p_\theta(x)}{p_{\hat{\gamma}}(x)}$$

Therefore, by our simplified metric, we have

$$\begin{aligned}&(\mathbb{E}_{p_{in}(x)}\log\frac{p_\theta(x)}{p_\gamma(x)} - \mathbb{E}_{p_{out}(x)}\log\frac{p_\theta(x)}{p_\gamma(x)}) - (\mathbb{E}_{p_{in}(x)}\log\frac{p_\theta(x)}{p_{\hat{\gamma}}(x)} - \mathbb{E}_{p_{out}(x)}\log\frac{p_\theta(x)}{p_{\hat{\gamma}}(x)})\\ &= \mathbb{E}_{p_{out}}\log\frac{p_\gamma(x)}{p_{\hat{\gamma}}(x)} - \mathbb{E}_{p_{in}}\log\frac{p_\gamma(x)}{p_{\hat{\gamma}}(x)} \geq \frac{\alpha}{\beta}\,\mathbb{E}_{p_{in}}\log\frac{p_{\hat{\gamma}}(x)}{p_\theta(x)} - \mathbb{E}_{p_{in}}\log\frac{p_\theta(x)}{p_{\hat{\gamma}}(x)}\\ &= \frac{\alpha+\beta}{\beta}\,\mathbb{E}_{p_{in}}\log\frac{p_\theta(x)}{p_{\hat{\gamma}}(x)} = \frac{1}{\beta}\,\mathbb{E}_{p_{in}}\log\frac{p_\theta(x)}{p_{\hat{\gamma}}(x)} \geq 0\end{aligned}$$

It means that $\frac{p_\theta(x)}{p_\gamma(x)}$ is a better OoD indicator than $\frac{p_\theta(x)}{p_{\hat{\gamma}}(x)}$ in case 2 and 3. In conclusion, $\frac{p_\theta(x)}{p_\gamma(x)}$ will be a better OoD indicator and thus they can be used to detect OoD if γ is trained better than $\hat{\gamma}$.

D Detailed Experiments

See the 'detailed_experiment' in supplemental file.

References

[1] Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., Ha, D.: Deep learning for classical japanese literature (2018)

[2] Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: Proceedings of the 23rd international conference on Machine learning. pp. 233–240 (2006)

[3] Deng, J., Dong, W., Socher, R., Li, L.j., Li, K., Li, F.f.: Imagenet: A large-scale hierarchical image database. CVPR pp. 248–255 (2009)

[4] Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images. Tech. rep., Citeseer (2009)

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

[6] LeCun, Y., Bottou, L., et al.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)

[7] Liu, Z., Luo, P., et al.: Deep learning face attributes in the wild. In: Proceedings of the IEEE international conference on computer vision. pp. 3730–3738 (2015)

[8] Netzer, Y., Wang, T., et al.: Reading digits in natural images with unsupervised feature learning (2011)

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

[10] Xiao, H., Rasul, K., et al.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017)

[11] Xu, P., Ehinger, A.K., Zhang, Y., Finkelstein, A., Kulkarni, R.S., Xiao, J.: Turkergaze: Crowdsourcing saliency with webcam based eye tracking. CoRR (2015)

[12] yu, f., zhang, y., song, s., seff, a., xiao, j.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. CoRR (2015)