Towards Out-of-Distribution Detection with Divergence Guarantee in Deep Generative Models

Yufeng Zhang, Wanwei Liu, Zhenbang Chen, Ji Wang, Zhiming Liu, Kenli Li, Hongmei Wei

Abstract—Recent research has revealed that deep generative models including flow-based models and Variational autoencoders may assign higher likelihoods to out-of-distribution (OOD) data than in-distribution (ID) data. However, we cannot sample out OOD data from the model. This counterintuitive phenomenon has not been satisfactorily explained. In this paper, we prove theorems to investigate the divergences in flow-based model and give two explanations to the above phenomenon from divergence and geometric perspectives, respectively. Our theoretical analysis inspires us to detect OOD data by Kullback-Leibler divergence between the distribution of representations and prior. Furthermore, we decompose the KL divergence to improve our group-wise anomaly detection method and support point-wise anomaly detection as well. We have conducted experiments on prevalent benchmarks to evaluate our method. Experimental results demonstrate the superiority of our OOD detection method.

Index Terms—Out-of-distribution detection, flow-based model, Kullback-Leibler divergence.

1 INTRODUCTION

A NOMALY detection is the process of “finding patterns in data that do not conform to expected behavior” [1]. Anomaly detection can be classified into group anomaly detection (GAD) [2] and point-wise anomaly detection (PAD) [1]. In unsupervised learning setting, the model is trained on a set of unlabeled data \( \{x_1, \ldots, x_n\} \) which are drawn independently from an unknown distribution \( p^* \). GAD is to determine whether a group of test inputs \( \{\tilde{x}_1, \ldots, \tilde{x}_m\} (m > 1) \) are drawn independently from an unknown distribution \( p^* \). When \( m = 1 \), GAD becomes PAD. Examples of GAD include discovering high-energy particle physics [4], anomalous galaxy clusters in astronomy [5], [6], unusual vorticity in fluid dynamics [7], and stealthy attacks [2], [8]. Examples of PAD include detecting intrusion [1], fraud [9], malware [10], and medical anomalies [1]. In literature, the term anomaly is also referred to as outlier, peculiarity, out-of-distribution (OOD) data, etc. In the following, we mainly use the term OOD data as in the most related works.

In this paper, we focus on unsupervised OOD detection using deep generative models (DGM) including flow-based model and VAE. Recent research shows that deep generative models (DGMs) including flow-based models [11], [12], VAE [13], and auto-regressive models [14], [15] are not capable of distinguishing OOD data from training data (or in-distribution (ID) data) according to the model likelihood [16], [17], [18], [19], [20], [21]. For example, as shown in Figure 1(b) and 1(c), Glow [11] assigns higher likelihoods for SVHN (MNIST) when trained on CIFAR-10 (FashionMNIST). However, as pointed by Nalisnick et al. [20] we cannot sample out OOD data although they are assigned higher likelihood. Another similar phenomenon is observed in class conditional Glow (GlowGMM), which contains a Gaussian Mixture Model on the top layer with one Gaussian for each class [11], [22], [23]. GlowGMM does not achieve the same performance as prevalent discriminative models (e.g., ResNet [24]) on FashionMNIST. This means that one component may assign higher likelihoods for other classes. As shown in Figure 7 in the supplementary material, these centroids are closer than we imagine. However, we always sample out images of the correct class from the corresponding component.

Nalisnick et al. explain the above phenomenon by the discrepancy of the typical set and high probability density regions of the model distribution [20]. They propose using typicality test to detect OOD data. However, their explanation and method fail on problems where the likelihoods of ID and OOD data coincide (e.g., CIFAR-10 vs CIFAR-100, CelebA vs CIFARs).

In this paper, we try to answer the following two questions:

1. Why cannot sample out new data similar to OOD data set in flow-based model? We need a unified answer to this question whenever OOD data have lower, higher, or coinciding likelihoods.
2. How to detect OOD data using flow-based model and VAE without supervision?

We start our research from the sampling process. Flow-based model constructs diffeomorphism from visible space...
to latent space. Each input data point is mapped to a unique representation in latent space. So we should ask why we cannot sample out the representations of OOD data from prior. In this paper, we first answer Q1 and then propose a unified OOD detection method.

The contributions of this paper are:

1) We prove several theorems to investigate the divergences in flow-based model. Based on these theorems, we attempt to provide a theoretical guarantee for OOD detection method.

2) We give two answers to Q1 from two perspectives. The first answer reveals the large divergence between the distribution of representations of OOD data and the prior. The second answer states that the representations of OOD data locate in specific directions.

3) Our answer to Q1 prompts us to perform GAD according to the Kullback-Leibler (KL) divergence between the distribution of representations and prior. However, estimating KL divergence is hard when OOD data set is arbitrary. Luckily, we observe that, for a wide range of problems, the representations of OOD data set in flow-based model follow a Gaussian-like distribution. This allows us to use the fitted Gaussian in KL divergence estimation and makes the whole method easy to perform. We also find that the same criterion works even better when the representations of OOD data set do not follow Gaussian-like distribution.

4) Furthermore, we decompose the KL divergence between the distribution of representations and prior to improve GAD method and support PAD. We split representations into multiple groups to implement our improvement. We also devise a splitting strategy that can leverage the local pixel dependence of representations.

5) We conduct experiments to evaluate our method. The results demonstrate the effectiveness, robustness, and generality of our method. For GAD, our method achieves near 100% AUROC for almost all the problems encountered in the experiments and is robust against data manipulations. On the contrary, the state-of-the-art (SOTA) GAD method is not better than random guessing on challenging problems and can be attacked by data manipulation in almost all cases. For PAD, our method also outperforms the baseline.

The remaining part of this paper is organized as follows. Section 2 gives the background and proposes data manipulations to attack the SOTA methods. Section 3 presents our theoretical analysis to answer Q1. Section 4 shows the details of our OOD detection method. Section 5 reports evaluation results. Section 6 discusses more details of our method. Section 7 discusses related work. Finally, Section 8 concludes.

2 Problem Settings

2.1 Background

Flow-based generative model constructs diffeomorphism $f$ from visible space $X$ to latent space $Z$ [11, 12, 25, 26]. The model uses a series of diffeomorphisms implemented by multilayered neural networks

$$x \xrightarrow{f_1} h_1 \xrightarrow{f_2} h_2 \ldots \xrightarrow{f_n} z$$

like flow. The whole bijective transformation $f(x) = f_n \circ f_{n-1} \ldots f_1(x)$ can be seen as encoder, and the inverse function $f^{-1}(z)$ is used as decoder. According to the change of variable rule, the probability density function of the model can be formulated as

$$\log p_X(x) = \log p_Z(f(x)) + \log \left| \frac{\partial z}{\partial x_T} \right|$$

where $x = h_0, z = h_n, \frac{\partial h_i}{\partial h_{i-1}}$ is the Jacobian of $f_i$, $\partial$ is the determinant.

Here prior $p_Z(z)$ is chosen as tractable density function. For example, the most popular prior is standard Gaussian $N(0, I)$, which makes $\log p_Z(z) = -(1/2) \times \sum_i z_i^2 + C (C$ is a constant). After training, one can sample noise $\varepsilon$ from prior and generate new samples $f^{-1}(\varepsilon)$.

Variational Autoencoder (VAE) is directed graphical model approximating the data distribution $p(x)$ with encoder-decoder architecture. The probabilistic encoder $q\phi(x|z)$ approximates the unknown intractable posterior $p(z|x)$. The probabilistic decoder $p\theta(x|z)$ approximates $p(x|z)$. In VAE, the variational lower bound of the marginal likelihood of data points (ELBO)

$$\mathcal{L}(\theta, \phi) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}_{z \sim q_\phi} [\log p_\theta(x^i|z)] - KL(q_\phi(z|x^i)||p(z))$$

can be optimized using stochastic gradient descent. After training, one can sample $z$ from prior $p(z)$ and use the decoder $p\theta(x|z)$ to generate new samples.
2.2 Attacking Likelihood

In [20], Nalisnick et al. conjecture that the counterintuitive phenomena in Q1 stem from the distinction of high probability density regions and the typical set of the model distribution [18, 20]. For example, Figure 2(a) shows the typical set of d-dimensional standard Gaussian, which is an annulus with a radius of \( \sqrt{d} \) [27]. When sampling from the Gaussian, it is highly likely to get points in the typical set. Neither the highest density region (i.e., the center) nor the lowest density region far from the mean would be reached. Based on this explanation, Nalisnick et al. propose using typicality test (Ty-test in short) to detect OOD data and achieve SOTA on this explanation, Nalisnick et al. in [20]. However, their explanation and method do not apply to problems where OOD data reside in the typical set of model distribution.

In the following, we show how to manipulate OOD data set to make the likelihood distribution of ID and OOD dataset coincide.

M1: rescaling \( z \) to typical set of prior. We train Glow with 768-dimensional standard Gaussian prior on FashionMNIST. Figure 1(a) shows the histogram of log-likelihood of representations of ID and OOD data sets. The likelihoods of ID and OOD data sets coincide, Ty-test expectations of MNIST (notMNIST). Ty-test can handle problems where the likelihoods of ID and OOD data sets diverge (e.g., FashionMNIST vs MNIST/notMNIST) [20]. However, when the likelihoods of ID and OOD data sets coincide, Ty-test fails (e.g., CIFAR-10 vs CIFAR-100 on Glow, Figure 1(c)). Nalisnick et al. also find that the likelihood distribution can be manipulated by adjusting the variance of inputs [16].

As shown in Figure 1(d), SVHN with increased contrast by a factor of 2.0 has coinciding likelihood distribution with CIFAR-10 on Glow trained on CIFAR-10. So it is impossible to detect OOD data by \( p(x) \) or typicality test on the model distribution. In our experiments, we can manipulate the likelihoods of OOD data set in this way for almost all problems. See Figures 2, 3 in the supplementary material for details. Besides, some more complicated likelihood-based methods, e.g., likelihood ratio, can also be affected by such manipulation (see Section 5). Similarly, in VAE, we can also manipulate the likelihood distribution by adjusting the contrast of input images.

In summary, we can manipulate both \( p(x) \) and \( p(z) \) of OOD data without the knowledge of the parameters of the model. In Section 5 we will show that our method is robust to the above manipulations.

2.3 Problems

We use ID vs OOD to represent an OOD detection problem and use “ID (OOD) representations” to denote representations of ID (OOD) data. According to the behavior of OOD data set, we group OOD detection problems into two categories:

- **Category I**: OOD data set has smaller or similar variance with ID data set and tends to have higher or similar likelihoods;
- **Category II**: OOD data set has larger variance than ID data and tends to have lower likelihoods.

As shown in the subsection 2.2, it is easy to use M2 to convert one problem from one category to another. In this paper, we present a unified OOD detection method for both categories.

3 Theoretical Analysis

In subsection 3.1, we first give several theorems that help to investigate the divergence between the distributions in flow-based model. Then in subsection 3.2 we give answers to Q1 from two perspectives.

3.1 Theorems

Large Divergence Guaranteed. In our analysis, we use \((h, \phi)\)-divergence family which includes many commonly used divergence measures in machine learning fields [29] (e.g., the KL divergence, Jensen-Shannon divergence, and squared Hellinger distance, see Section A in the supplementary material for details).

**Theorem 1** Given a flow-based model \( z = f(x) \) with prior \( p_z \). Suppose that \( X_1 \sim p_X(x) \), \( X_2 \sim q_X(x) \), \( Z_1 = f(X_1) \sim p_Z(z) \) and \( Z_2 = f(X_2) \sim q_Z(z) \). Let \( D_h^{\phi} \) be a \((h, \phi)\)-divergence measure, \( D \) be a proper statistical distance metric belonging to the \((h, \phi)\)-divergence family, and \( R_D \) be the range of \( D \).

\[
D_h^{\phi}(p_X, q_X) = D_h^{\phi}(p_Z, q_z) \text{ holds} [30].
\]

(b) For any \( 0 < d < \sup(R_D) \), there are \( d' > 0 \) and \( \epsilon > 0 \) so that \( D(q_Z, p_Z) > d \) when \( D(p_X, q_X) > d' \) and \( D(p_Z, p_Z') < \epsilon \).

**Proof** (a) Since \( D_h^{\phi}(p, q) = h(D_\phi(p, q)) \), where \( D_\phi(p, q) \) is \( \phi \)-divergence between \( p \) and \( q \), it suffices to prove...
(b) Since D is a proper statistical distance metric and satisfies the triangle inequality, we have \( D(p_{yz}, p'_{yz}) + D(q_{z}, p'_{z}) \geq D(p_{yz}, q_{z}) \). For any \( d > 0 \) and \( \epsilon > 0 \), if \( D(p_{yz}, q_{z}) > d + \epsilon = d' \) and \( D(p_{yz}, p'_{y}) < \epsilon \), we have \( D(p_{yz}, q_{z}) > d' \). Since D belongs to the \((h, \phi)-\)divergence family, from Theorem 1 we know \( D(p_{x}, q_{x}) = D(p_{yz}, q_{z}) \). Thus we have Theorem 1. \( \square \)

Note. The proof of Theorem 1 relies on diffeomorphisms. According to the Brouwer Invariance of Domain Theorem [31], \( \mathbb{R}^n \) cannot be homeomorphic to \( \mathbb{R}^m \) if \( n \neq m \). So Theorem 1 does not apply to non-diffeomorphisms (e.g., vanilla VAE). The Brouwer Invariance of Domain Theorem also implies that there is no dead neuron in flow-based model. Otherwise, we can construct diffeomorphism from high to low-dimensional space.

Theorem 1 provides a general perspective for investigating the divergences between distributions in flow-based model. Currently, flow-based models are usually trained by maximum likelihood estimation, which is equivalent to minimizing the forward KL divergence \( KL(p_{x}(x)||p_{x}(x)) \) [26], [32], where \( p_{x}(x) \) is the distribution of training data and \( p'_{x}(x) \) is the model distribution. However, KL divergence is not symmetric. The difference between forward and reverse KL divergence of two distributions can be any large. We cannot apply Theorem 1 on KL divergence directly.

The following theorems 2-4 use a famous transcendental function, the Lambert W function.

**Definition 1 Lambert W Function** [33], [34]. The reverse function of function \( y = x^e \) is called Lambert W function \( y = W(x) \).

When \( x \in \mathbb{R}, W \) is a multivalued function with two branches \( W_0, W_{-1} \), where \( W_0 \) is the principal branch (also called branch 0) and \( W_{-1} \) is the branch \(-1\).

**Approximate symmetry of small KL divergence between Gaussians.** The following Theorem 2 guarantees that the KL divergence between two \( n \)-dimensional Gaussians \( KL(N_1||N_2) \) is small when \( KL(N_2||N_1) \) is small.

**Theorem 2** Let \( KL \) be the Kullback-Leibler divergence, \( \varepsilon \) be a positive real number, \( W_0 \) be the principal branch of the Lambert W Function. For any \( n \)-dimensional Gaussians \( N(\mu_1, \Sigma_1) \) and \( N(\mu_2, \Sigma_2) \), if \( KL(N(\mu_1, \Sigma_1)||N(\mu_2, \Sigma_2)) \leq \varepsilon \), then

\[
KL(N(\mu_2, \Sigma_2)||N(\mu_1, \Sigma_1)) \leq 1 + 2 \left\{ \frac{1}{W_0(-e^{-(1+2\varepsilon)})} - \log \frac{1}{W_0(-e^{-(1+2\varepsilon)})} - 1 \right\} \tag{4}
\]

**Proof** The proof is too long to be included in the same paper. The details of the proof are presented in our work [35], which is submitted independently to another venue.

**Notes.** The supremum in Inequation 4 is small when \( \varepsilon \) is small. See Table 10 in the supplementary material for some approximate values of the supremum. Besides, it needs strict conditions to make Inequality 4 tight. In machine learning practice, KL divergence is usually much smaller than the supremum. See our proof in [35] for details.

The following Theorem 3 gives the infimum of \( KL(N_1||N_2) \) when \( KL(N_1||N_2) \geq M \) for \( M > 0 \).

**Theorem 3** Let \( KL \) be the Kullback-Leibler divergence, \( M > 0 \) be a positive real number, \( W_0 \) be the branch \(-1\) of the Lambert W Function. For any two \( n \)-dimensional Gaussians \( N(\mu_1, \Sigma_1) \) and \( N(\mu_2, \Sigma_2) \), if \( KL(N(\mu_1, \Sigma_1)||N(\mu_2, \Sigma_2)) \geq M \), then

\[
KL(N(\mu_2, \Sigma_2)||N(\mu_1, \Sigma_1)) \geq \frac{1}{2} \left\{ -W_{-1}(-e^{-(1+2M)}) - \log \frac{1}{W_{-1}(-e^{-(1+2M)})} - 1 \right\} \tag{5}
\]

**Proof** Theorem 3 can be derived from Theorem 2 or proved independently. See our work [33] for the proof.

Relaxed triangle inequality of KL divergence between Gaussians. The following Theorem 4 shows that the KL divergence between Gaussians follows a relaxed triangle inequality.

**Theorem 4** Let \( KL \) be the Kullback-Leibler divergence, \( W_0 \) and \( W_{-1} \) be the branches \(-1\) of the Lambert W Function, respectively. For any \( n \)-dimensional Gaussians \( N(\mu_1, \Sigma_1) \), \( N(\mu_2, \Sigma_2) \) and \( N(\mu_3, \Sigma_3) \), if \( KL(N(\mu_1, \Sigma_1)||N(\mu_2, \Sigma_2)) \leq \varepsilon_1 \), \( KL(N(\mu_2, \Sigma_2)||N(\mu_3, \Sigma_3)) \leq \varepsilon_2 \), then

\[
KL(N(\mu_1, \Sigma_1)||N(\mu_3, \Sigma_3)) \leq \varepsilon_1 + \varepsilon_2 + \frac{1}{2} \left\{ W_{-1}(-e^{-(1+2\varepsilon_1)})W_{-1}(-e^{-(1+2\varepsilon_2)}) \right. \\
+ W_{-1}(-e^{-(1+2\varepsilon_1)}) + W_{-1}(-e^{-(1+2\varepsilon_2)}) + 1 \\
- W_{-1}(-e^{-(1+2\varepsilon_2)}) \left( \sqrt{2\varepsilon_1} + \sqrt{\frac{2\varepsilon_2}{W_0(-e^{-(1+2\varepsilon_2)})}} \right)^2 \right\} \tag{6}
\]

**Proof** The proof is too long to be included in the same paper. See our work [33] for details.

Most importantly, all the bounds in Theorem 2-4 and 4 are independent of the dimension \( n \). So these theorems can be applied to high-dimensional problems (e.g., flow-based model).

### 3.2 Why Cannot Sample Out OOD Data?

Based on the theorems we have proven, we can give two explanations on why cannot sample out OOD data from two perspectives.

#### 3.2.1 Divergence Perspective

Figure 3 illustrates how we can apply Theorem 1 to investigate the divergences between the following distributions: the distributions of ID data \( p_{X}(x) \) and OOD data \( q_{X}(x) \), the distributions of ID representations \( p_{Z}(z) \) and OOD representations \( q_{Z}(z) \), the prior \( p_{Z}(z) \), and the model induced distribution \( p'_{Z}(z) \) such that \( Z_{r} \sim p_{Z}(z) \) and \( X_{r} = f^{-1}(Z_{r}) \sim p'_{X}(x) \). In the following, we first discuss the general case. Then we conduct further analysis for one category of problems.

1. **General case.** Our analysis consists of the following steps.

   - **Step 1:** In practice, ID and OOD datasets are sampled from different distributions. Take KL divergence \( KL(p_{X}(x)||q_{X}(x)) = \int p_{X}(x) \log \frac{p_{X}(x)}{q_{X}(x)} \, dx \) as an example. When each input \( x \) belongs to only one dataset, \( KL(p_{X}(x)||q_{X}(x)) \) can be considered as any large.
In practice, in Step 4 we assume \( p_z \approx p'_z \). Here we do not require a precise approximation. This is because \( KL(q_z || p_z) \) is large enough in practice. Take SVHN vs CIFAR-10 for example, each data point belongs to only one dataset. So \( KL(q_z || p_z) \) can be assumed any large. Such precondition allows us to use an imprecise approximation in \( p_z \approx p'_z \).

### The Gaussian case

We also perform normality test on OOD representations. The results are surprising. As shown in Table 2 OOD representations in all Category I problems except for SVHN vs Constant have \( p \)-values greater than 0.05. As far as we know, we are the first to observe this phenomenon (More details are presented in section 4.1.1). In summary, we have the following key observation.

**Key observation:** ID representations of flow-based models follow Gaussian-like distribution. For Category I problems, the representations of OOD dataset tend to follow Gaussian-like distribution.

Using our theorems on the properties of KL divergence between Gaussians, we can conduct further analysis when \( q_z \) is Gaussian-like. We can use a Gaussian \( N_q \) to approximate \( q_z \) and have \( KL(q_z || p_z) \approx KL(N_q || p_z) \). Now we know that \( KL(p_z || q_z) \) is large and \( KL(p_z || p'_z) \) is small. According to the relaxed triangle inequality in Theorem 3, \( KL(p'_z || N_q) \) must not be small. Furthermore, we can apply Theorem 3 on \( KL(p'_z || N_q) \) and know that \( KL(p'_z || p'_z) \) is large. Finally, we know \( KL(q_z || p'_z) \) is large too.

#### 3.2.2 Geometric Perspective

Theorem 5 decomposes forward KL divergence into two parts. It provides a basis for further analysis.

**Theorem 5** Let \( X \sim p'_X(x) \) be an \( n \)-dimensional random vector, \( X_i \sim p'_X(x) \) be the \( i \)-th dimensional element of \( X \). Then

\[
KL(p'_X(x) || N(0, I_n)) = KL(p'_X(x) || \prod_{i=1}^{n} p'_X(x)) + \sum_{i=1}^{n} KL(p'_X(x) || N(0, 1))
\]

**Proof**

\[
KL(p'_X(x) || N(0, I_n))
= \mathbb{E}_{p'_X(x)} \left[ \log \left( \frac{p'_X(x)}{\prod_{i=1}^{n} p'_X(x)} \right) \right]
= \mathbb{E}_{p'_X(x)} \left[ \log \left( \prod_{i=1}^{n} \frac{p'_X(x)}{p'_X(x)} \right) \right] + \sum_{i=1}^{n} \mathbb{E}_{p'_X(x)} \left[ \log \left( \frac{p'_X(x)}{N(0, 1)} \right) \right]
= \sum_{i=1}^{n} \mathbb{E}_{p'_X(x)} \left[ \log \left( \frac{p'_X(x)}{N(0, 1)} \right) \right]
= KL(p'_X(x) || \prod_{i=1}^{n} p'_X(x)) + \sum_{i=1}^{n} KL(p'_X(x) || N(0, 1))
\]
We can see that OOD representations are more correlated. In diagonal elements in the correlation matrix of representations. MNIST/notMNIST. Figure 4 shows the histogram of the non-coefficient, to investigate the linear dependency.

\[ I_{d} \] is large \( KL \) in independent representations. On the contrary, for OOD data, containing the divergence between the marginal distribution of \( D \) dimensions \([37]\); information) measuring the mutual dependence between non-negative parts: \( J \) is Gaussian-like the distribution is. When \( p \) tends to be Gaussian-like.

From a geometric perspective, a high correlation between dimensions indicates the representations of OOD data set locate in specific directions \([38]\). In high dimensional space, it is hard to obtain data on specific directions by sampling from standard Gaussian. When OOD representations reside in the typical set of prior, we can treat the phenomenon in Q1 as a manifestation of the curse of dimensionality.

Furthermore, we scale the norm of OOD representations with different factors. The decoded images also vary from ID data to OOD data gradually. See Figure 13 in the supplementary material for details.

Overall, this leads to the second answer to Q1.

**Answer 2 to Q1:** OOD representations locate in specific directions with specific norms. In high dimensional space, it is hard to sample out data in specific directions from prior regardless of whether these data reside in the typical set or not.

\[ D_{j} \] is dimension-wise KL divergence containing the divergence between the marginal distribution of each dimension and prior. We use \( p^{*}_{X} \) to denote one term is computed from \( p_{X} \).

Theorem 5 decomposes forward KL divergence into two non-negative parts: \( I_{d} \) is total correlation (generalized mutual information) measuring the mutual dependence between dimensions \([37]\); \( D_{j} \) is dimension-wise KL divergence containing the divergence between the marginal distribution of each dimension and prior. We use \( p^{*}_{X} \) to denote one term is computed from \( p_{X} \).

Theorem 5 can help us to investigate the forward KL divergence further. For ID data, we have known that \( KL(p_{X}(z)||p^{*}_{X}(z)) \) is small. According to Theorem 5, \( I_{d} \) is also small. This indicates that ID data tends to have independent representations. On the contrary, for OOD data, a large \( KL(q_{Z}(z)||p^{*}_{X}(z)) \) allows a large \( I_{d} \). Although it is hard to estimate \( I_{d} \), we can use an alternative dependence measure, \( i.e., the most commonly used correlation coefficient, to investigate the linear dependency.

We train Glow on FashionMNIST and test on MNIST/notMNIST. Figure 4 shows the histogram of the non-diagonal elements in the correlation matrix of representations. We can see that OOD representations are more correlated. In fact, this happens for all the problems in our experiments. See Figure 18 to 23 in supplementary material for more details. We note that correlation completely characterizes dependence only when data follows Gaussian distribution. In last subsection, we have shown that for \( Category I \) problems, \( q_{z} \) tends to be Gaussian-like.

**4 Anomaly Detection Method**

In this section, we first propose a preliminary GAD method. Then we improve the method to support small batch size and PAD.

**4.1 A Preliminary GAD Method**

Answer 1 reminds us to detect OOD data by estimating \( KL(p||p_{z}) \), where \( p \) is the distribution of representations of inputs. However, when only samples are available, divergence estimation is provable hard and the estimation error decays slowly in high dimension space \([39],[40],[41]\). This brings difficulty in applying existing divergence estimation \([41],[42],[43],[44],[45]\) to high dimensional problems with small sample size. Luckily, as shown in Table, we observe that both ID data and OOD data of \( Category I \) problems follow Gaussian-like distribution. This provides us a facility to estimate the KL divergence for GAD.
4.1.1 Flow-based Model

**ID data.** As discussed in Subsection 3.2, we can use a Gaussian \(N_p\) to approximate \(p_z\). Here we use sample expectation \(\mu\) and covariance \(\Sigma\) of representations to estimate the parameters of \(N_p\) \[1\]. Experiments also show that we can generate high-quality images by sampling from \(N_p\) rather than the prior. Therefore, we have

\[
\begin{align*}
K(p_z||p_\phi) & \approx KL(N_p(\mu, \Sigma)||N(\mu, \Sigma)) \\
& = \frac{1}{2} \left\{ \log \frac{\Sigma}{\Sigma} + \text{Tr}(\Sigma^{-1} \Sigma) + (\mu - \hat{\mu})^T \Sigma^{-1}(\mu - \hat{\mu}) - n \right\} \\
& = \frac{1}{2} \left\{ - \log |\Sigma| + \text{Tr}(\Sigma) + \mu^T \hat{\mu} - n \right\} \\
& \quad \quad \quad \quad \text{for } q_\phi(z). \quad (9)
\end{align*}
\]

**OOD data in Category I problems.** At the very beginning, we observed that the fitted Gaussian from OOD representations contains style information of that dataset. We train Glow on CIFAR-10 and test on notMNIST (grayscale images are preprocessed for consistency, see Section 5.1). Then we replace the prior with fitted Gaussian from representations of notMNIST and generate new images. Surprisingly, as shown in Figure 5, we find that the generated images seem similar to notMNIST, although the images are blurred. In this way, using a single Glow model, we can generate images with the style of multiple OOD datasets, including (not)MNIST, SVHN, CelebA, etc. See Figure 14–16 in the supplementary material for details. Such a similar phenomenon is also reported by [46], which is released contemporaneously with the first edition of this paper.

Most importantly, these results remind us that \(q_z\) may be also Gaussian-like to some extent. To validate this intuition, we perform generalized Shapiro-Wilk test for multivariate normality [47] on the representations. For each dataset, we randomly select 2000 inputs for normality test. As shown in Table 1, all almost OOD data sets of Category I problems have \(p\)-values greater than 0.05. It seems that OOD data sets “sitting inside” of the training data are also “Gaussianized” along with the training data.

Based on this observation, we can use fitted Gaussian \(N_q\) to approximate \(q_z\). This allows us to use an expression to estimate \(KL(q_z||p_\phi)\).

**OOD data in Category II problems.** Table 1 shows that OOD representations in Category II problems do not follow Gaussian-like distribution. Nevertheless, we find that Equation (9) performs even better on Category II problems.

When \(p_\phi(z) = N(0, I)\), Equation (9) equals to

\[
\begin{align*}
& \frac{1}{2} \left\{ - \log |\Sigma| + \text{Tr}(\Sigma) + \hat{\mu}^T \hat{\mu} - n \right\} \\
& \quad \quad \quad \quad \text{for } q_\phi(z). \quad (10)
\end{align*}
\]

1. This is equal to using maximum likelihood estimation [52].
2. In [46], the authors only use the mean of fitted Gaussian, not including the covariance.

where generalized variance \(\tilde{\Sigma}\) and total variation \(\text{Tr}(\tilde{\Sigma})\) both measure the dispersion of all dimensions. So the first two items of Equation (10) compensate each other. For Category I problems, OOD representations are less dispersed than ID representations and have a larger \(-\log |\tilde{\Sigma}|\). For Category II problems, OOD representations tend to be more dispersed, so \(\text{Tr}(\tilde{\Sigma})\) is larger. Besides, we find OOD representations always have a larger \(\hat{\mu}^T \hat{\mu}\) than ID representations.

In fact, if OOD representations do not follow Gaussian-like distributions, Equation (10) is a conservative criterion. The reason is revealed by the following theorem.

**Theorem 6** (see [48]) Let \(N_1(\mu_1, \Sigma_1)\) and \(N_2(\mu_2, \Sigma_2)\) be two \(n\)-dimensional Gaussian distributions. Assume that \(Z \sim p_Z(z)\) is an arbitrary \(n\)-dimensional continuous random variable with mean vector \(\mu\) and covariance matrix \(\Sigma\), then

\[
KL(N_1(\mu_1, \Sigma_1)||N_2(\mu_2, \Sigma_2)) \leq KL(p_Z(z)||N_1(\mu_2, \Sigma_2)) \leq KL(p_Z(z)||N_2(\mu_1, \Sigma_1)) \quad (11)
\]

According to Theorem 6, Equation (10) computes a lower bound of \(KL(q_z||p_\phi)\). If the criterion is large, then \(KL(q_z||p_\phi)\) must be large.

Overall, we get a preliminary answer to Q2.

**A preliminary answer to Q2:** Equation (10) can be used as a unified conservative criterion for CACD due to the following reasons.

1. For ID data, Equation (10) approximates \(KL(p_z||p_\phi)\) and should be small;
2. For OOD data whose representations follow Gaussian-like distribution, Equation (10) approximates \(KL(q_z||p_\phi)\) and should be large;
3. For OOD data whose representations do not follow Gaussian-like distribution, Equation (10) computes the lower bound of \(KL(q_z||p_\phi)\). If the lower bound is large, then \(KL(q_z||p_\phi)\) must be large.

Besides, when batch size is too small to estimate the parameters, Equation (10) should be treated just as a statistic. Note that Equation (10) also applies to Gaussian prior with diagonal covariance \(\text{diag}(\sigma)\) and mean \(\mu\). In such a case, we only need to normalize the data by a linear operation \(Z' = (Z - \mu)/\sigma\) while keeping \(KL(p_Z(z)||N(\mu, \text{diag}(\sigma))) = KL(p_Z(z)||N(0, I))\) \((\text{by Theorem 1})\). This equals to using Equation (9) directly. We also emphasize that we are not pursuing precise divergence estimation or parameter estimation, which are proven to be hard with very small batch sizes in high-dimensional problems.

4.1.2 VAE

It is well-known that VAE and its variations learn independent representations \([49, 50, 51, 52, 53]\). In VAE, the probabilistic encoder \(q_{\phi}(z|x)\) is often chosen as Gaussian form \(N(\mu(z), \text{diag}(\sigma(z)^2))\), where \(z \sim q_{\phi}(z|x)\) is used as sampled representation, \(\mu(x)\) is used as mean representation.

The KL term in variational evidence lower bound objective (ELBO) can be rewritten as \(\mathcal{E}_{p(x)}[KL(q_z(z|x)||p(z))] = I(x; z) + KL(q(z)||p(z))\), where \(p(z)\) is the prior, \(q(z)\) the aggregated posterior, and \(I(x; z)\) the mutual information between \(x\) and \(z\) [54]. Here the term \(KL(q(z)||p(z))\) pulls \(p_z\)
to the Gaussian prior and encourages independent sampled representations. We also investigate the representations in VAE. The results show that:

1) ID representations in VAE do not always have p-value greater than 0.05;
2) the representations of all OOD datasets do not have p-value greater than 0.05;
3) the sampled (mean) representations of OOD datasets are more correlated (see Figure 23-26 in the supplementary material).

Furthermore, there is no theoretical guarantee that $KL(q_z||p^*_Z)$ is large enough because Theorem 1 does not apply to non-diffeomorphisms.

We also tried the SOTA $\phi$-divergence estimation method applicable for VAE, i.e. RAM-MC [41]. Results show RAM-MC can also be affected by data manipulation M2 (see Section 2.2). Finally, we find that Equation (10) also works for GAD with VAE.

4.2 Improvement

Up to now, we are still facing two challenges. Firstly, the performance of the preliminary GAD method tends to decrease when the batch size $m$ is small (e.g., $m = 5$). Secondly, it seems impossible to apply our theorems to PAD because we cannot estimate the parameters from one single data point. In this section, we improve our preliminary method to tackle these two challenges. The key idea is splitting representation into groups.

4.2.1 Splitting Dimensions into Groups

The factorizability of standard Gaussian allows us to investigate representations in groups. Intuitively, if $z \sim \mathcal{N}(0, I)$, then each dimension group of $z$ follows $\mathcal{N}(0, I)$. Otherwise, it is unlikely that each part of $z$ follows $\mathcal{N}(0, I)$. Thus, we can split one single $z$ into multiple subvectors and investigate these subvectors separately. This also generates multiple samples from one data point artificially. Formally, we split random vector $Z$ into $k$ $l$-dimensional ($k = n/l$) subvectors $\hat{Z}_1, \ldots, \hat{Z}_k$. We note the marginal distribution of $\hat{Z}_i$ as $p_{\hat{z}} (z)$ ($1 \leq i \leq k$). Then we can use the following Theorem 7 to decomposes $KL(p_{\hat{z}}(z)||\mathcal{N}(0, I_n))$ further.

**Theorem 7** Let $X \sim p^*_X(x)$ be an $n$-dimensional random vector. We note $X = \hat{X}_1, \ldots, \hat{X}_k$ where $\hat{X}_i \sim p^*_X(x)$ be the $i$-th $l$-dimensional ($k = n/l$) subvector of $X$, $\hat{X}_{ij} \sim p^*_{X_{ij}}(x)$ be the $j$-th element of $\hat{X}_i$. Then,

$$KL(p^*_X(x)||\mathcal{N}(0, I_n)) = \sum_{i=1}^k KL(p^*_{X_i}(x)||\mathcal{N}(0, I_n))$$

Proof We can use the similar deduction in Theorem 5 and get Equation (12).

$$KL(p^*_X(x)||\mathcal{N}(0, I_n)) = \sum_{i=1}^k KL(p^*_{X_i}(x)||\mathcal{N}(0, I_n))$$

Then we apply Theorem 5 on each $D_g[p^*_{X_i}]$ and have

$$KL(p^*_X(x)||\mathcal{N}(0, I_n)) = \sum_{i=1}^k KL(p^*_{X_{ij}}(x)||\mathcal{N}(0, I_n))$$

Finally, combining Equation (12) and (13) we can obtain Equation (14).

In Equation (12), $I_g$ is the generalized mutual information between dimension groups [37]. $D_g$ is group-wise KL divergence. Furthermore, in Equation (13) $D_g$ is decomposed as $I_g + D_d$, where $I_g$ is the generalized mutual information inside each group, $D_d$ is dimension-wise KL divergence which also occurs in Equation (7). Combining Equation (7) and (13) we have $I_g + I_d$ and $D_g = I_l + D_d$. Compared with Equation (7), Equation (12) distributes more divergence into the second part. When $k = n$, Equation (12) is equal to Equation (7).

Applying Theorem 7 on $p_z$ and $q_z$, we get

$$KL(p_z||p^*_Z) = I_g[p_z] + D_g[p_z] = I_g[p_z] + \sum_{i=1}^k D_g[p^*_{Z_{ji}}]$$

$$KL(q_z||p^*_Z) = I_g[q_z] + D_g[q_z] = I_g[q_z] + \sum_{i=1}^k D_g[q^*_{Z_{ji}}]$$

where $p^*_{Z_i}, q^*_{Z_i}$ are the marginal distributions of subvectors of ID and OOD representations, respectively. In Section 3, we have known

$$I_g[q_z] + D_g[q_z] > I_g[p_z] + D_g[p_z]$$

Since $I_g[p_z] + D_g[p_z]$ is small, we can assume that $I_g[p_z] < \varepsilon$. To make Equation (15) hold, it suffices that $D_g[q_z] > D_g[p_z] + \varepsilon$. If the splitting strategy (see Section 4.2.2) distributes more divergence to $D_g[q_z]$ in Equation (12), it is high likely that $D_g[q_z] > D_g[p_z]$. Therefore, we can use $D_g$ as the criterion to detect OOD data. The remaining problems are how to estimate $D_g$ and how to choose a splitting strategy.

**Estimating $D_g$**. For ID data, we treat each representation as $k$ data points sampled from a mixture of distributions $p^*_{Z_{ij}}(z) = (1/k) \sum_{i=1}^k p^*_{Z_{ij}}(z)$ where $p^*_{Z_i}(1 \leq i \leq k)$ is very close to $\mathcal{N}(0, I_k)$. Thus, we can use a single Gaussian $\mathcal{N}_{Z_i}$ to approximate each $p^*_{Z_i}$. Therefore, $D_g[p_z]$ can be approximated as

$$D_g[p_Z] = \sum_{i=0}^k KL(N_{Z_i}||\mathcal{N}(0, I_n)) = k \times D_g[p^*_{Z_i}]$$
Now we can plug Equation (10) in Equation (16) except that each representation $z$ is treated as $k$ samples of $p_{Z_m}$. For OOD data, we cannot use a single Gaussian to approximate $q_{Z_m}(z) = (1/k) \sum_{i=1}^{k} q_{Z_i}(z)$ when $q_{Z_i}(z)$ are far from each other or $q_{Z}$ is not Gaussian-like. Nevertheless, we still can use Equation (10) as a statistic measuring the dispersion and deviation.

Finally, splitting representations not only makes PAD possible but also improve GAD performance because it increases the batch size artificially.

4.2.2 Splitting Strategy: Leveraging Local Pixel Dependence

Our aim is to use $D_g$ (Equation (16)) for OOD detection. When choosing the splitting strategy, we have the following two principles.

1. We should retain enough intragroup dependence in $I_q$ to make $D_g[q_Z] > D_g[p_Z] + \epsilon$;
2. When the batch size is too small to estimate parameters, Equation (10) should still be a qualified statistic for OOD detection.

Take the Glow model for example, a representation $z$ has shape $(H \times W \times C)$ where $H, W, C$ are the height, width, and the number of channels of $z$, respectively. The most natural choices are:

1. S1: split $z$ as $H \times W \times C$-dimensional vectors;
2. S2: split $z$ as $(H \times W)$-dimensional vectors.

S1 retains inter-channel dependence into $D_g$ while S2 retains pixel dependence into $D_g$.

In Glow, the representation has shape $(4 \times 4 \times 48)$. The number of channels is much larger than the size of one channel. Our analysis of the correlation matrix also indicates that more divergence occurs between channels than that between pixels. So it seems that S1 tends to have a larger $I_q[q_Z]$.

However, we find S2 is better than S1 especially when the batch size is small. The reason is that representations have simpler inter-channel dependence than pixel dependence as like natural images [55]. We split a single $z$ into $k$ subvectors $z_1, \ldots, z_k$. Then we treat $z_1, \ldots, z_k$ as samples of one random vector $Z_m$ with multimodal distribution. If the $r$-th element $Z_{r,m}$ and $s$-th element $Z_{s,m}$ are strongly correlated for all $1 \leq i \leq k$, we can say that $Z_{m,r}$ and $Z_{m,s}$ are also strongly correlated. More generally, if $Z_1, \ldots, Z_k$ have the similar dependence structure, $Z_m$ would also has the similar dependence structure. Based on this intuition, we find that OOD representations manifest local pixel dependence. For example, we test CIFAR-10 and Imagenet32 on Glow trained on SVHN. For each OOD dataset, we visualize the correlation between pixels. As shown in Figure 27 in supplementary material, we find for almost all channels each pixel always has stronger correlation with its neighbors. Therefore, we can say that $Z_1, \ldots, Z_k$ tend to have a similar dependence structure. This means that using strategy S2 tends to have a larger result when calculating Equation (10).

On the contrary, when using strategy S1 we cannot observe a similar dependence structure between channels. Therefore, S2 is more suitable for PAD. The above analysis also applies to GAD with small batch size ($2 \sim 5$).

Besides, we have also tried other splitting strategies. Evaluation results show that S2 is the most stable and best one.

Algorithm 1 Anomaly Detection method (KLODS)
1: Input: $f(x)$: a well-trained flow-based model or the encoder of VAE using Gaussian prior $N(\mu, \text{diag}(\sigma))$; $X = \{x_1, \ldots, x_m\} (m \geq 1)$: a batch of inputs; $(H, W, C)$: the shape of representations. $t$: threshold
2: $\mathbb{Z} = \emptyset$
3: for $i = 1$ to $m$
4: $z_i = f(x_i)$
5: $z'_i = z_i - \mu$
6: split $z'_i \sigma$ as $C (H \times W)$-dimensional subvectors $z'_{i,1}, \ldots, z'_{i,C}$
7: $\mathbb{Z} = \mathbb{Z} \cup \{z'_{i,1}, \ldots, z'_{i,C}\}$
8: end for
9: compute sample covariance $\Sigma$ and sample mean $\bar{\mu}$ of $\mathbb{Z}$
10: $c = (1/2)\{-\log |\Sigma| + tr(\Sigma) + \bar{\mu}^\top \bar{\mu} - n\}$
11: if $c > t$ then
12: return “$X$ is OOD data”
13: else
14: return “$X$ is ID data”
15: end if

Overall, we get an answer to Q2.

Answer to Q2: We use $D_g$ (computed by Equation (16) and (10)) as a unified criterion for both GAD and PAD for flow-based models.

4.3 Algorithm

Algorithm 1 shows the details of our OOD detection method. Given inputs $X = \{x_1, \ldots, x_m\} (m \geq 1)$, we compute the representations of each $z_i = f(x_i)$ and split normalized $z'_i = (z_i - \mu)/\sigma$ as $C (H \times W)$-dimensional subvectors. Then we collect all the subvectors as $\mathbb{Z}$ and use Equation (10) as the criterion. If $c$ is greater than a threshold $t$, the input is determined as OOD data. Otherwise, the input is determined as ID data. Algorithm 1 becomes PAD when $m = 1$. We name our method as KLODS for KL divergence-based Out-of-Distribution Detection with Splitted representations.

Without splitting representations (line 6), Algorithm 1 can be used only for GAD. We call the algorithm without splitting representations as KLOD. Experimental results show that KLOD needs a larger batch size to achieve the same performance as KLODS for GAD.

5 Experiments

We conduct experiments to evaluate the effectiveness, robustness, and generality of our OOD detection method.

5.1 Experimental Setting

Benchmarks. We evaluate our method with prevalent benchmarks in deep anomaly detection research [15, 17, 20, 56, 57, 58], including MNIST [59], FashionMNIST [60], notMNIST [61], CIFAR-10/100 [62], SVHN [63], CelebA [64], TinyImageNet [65], and ImageNet32 [66]. We also construct unnatural images as OOD data. Constant consists of images with all pixels equal to the same constant $C \sim U \{0, 255\}$. 
Uniform consists of images with each pixel sampled independently from \( U \{0, 255\} \). We also use the mixtures of different data sets as OOD data in GAD problems.

We use different dataset compositions falling into Category I and II problems. For example, CIFAR-10 vs SVHN falls into Category I and SVHN vs CIFAR-10 falls into Category II. All datasets are resized to \( 32 \times 32 \times 3 \) for consistency. For grayscale datasets of size \( 28 \times 28 \times 1 \), we replicate channels and pad zeros around the image. We use \( S^{-C}(k) \) (\( k \geq 0 \)) to denote dataset \( S \) with adjusted contrast by a factor \( k \). See Figure 29 in the supplementary material for examples. The size of each test dataset is fixed to 10,000 for comparison.

**Models.** For flow-based model, we use OpenAI’s open-source implementation of Glow [67] with 768-dimensional standard Gaussian as prior except for CIFAR-10. For CIFAR-10, we use model checkpoint released by the authors of [20], [68] for fairness. The prior has learned mean and diagonal covariance. For VAE, we train convolutional VAE and use sampled representation for all problems. See Section C in the supplementary material for more details.

**Metrics.** We use threshold-independent metrics: area under the receiver operating characteristic curve (AUROC) and area under the precision-recall curve (AUPR) to evaluate our method [69]. We treat OOD data as positive data. For GAD, each dataset is shuffled and then divided into groups of size \( n \). We compute AUROC and AUPR according to the portion of groups determined as OOD data.

**Baselines.** As far as we know, before this submission, there exist five methods that handle OOD data with higher likelihood in flow-based model under unsupervised setting.

1. WAIC [18]. In [20], Nalisnick et al. state that they were not able to replicate the results of WAIC. We also do not use WAIC as baseline.
2. Typicality test in latent space [18]. In Section 2.2 we have shown typicality test in latent space can be attacked by data manipulation.
3. Typicality test in model distribution (Ty-test) [20]. Ty-test is the only GAD method among the five methods. We use it as the baseline for GAD. Since Ty-test outperforms all other methods compared in [20], we do not use more baselines for GAD.
4. Input complexity compensated likelihood [70]. We use this method as the baseline for PAD.
5. Likelihood ratios [71]. In [70], Serrà et al. interpret their method as a likelihood-ratio test statistic and achieve better performance than method 5. Therefore, method 5 can be seen as an instance of method 4. Besides, the authors of method 5 did not report results on flow-based models. So we do not use method 5 as the baseline.

We run each method 5 times and show “mean±std” for each problem.

### 5.2 Experimental Results

#### 5.2.1 GAD Results

**KLODS on Unconditional Glow.** Table 2 shows the results of KLODS on Glow trained on FashionMNIST, SVHN, CIFAR-10, CelebA and tested on OOD datasets. We can see that our method outperforms the baseline. Specially, we adjust the contrast of OOD data set to make the likelihood distributions of ID and OOD data coincide. For these kinds of problems, the performance of Ty-test degenerates severely. On the contrary, our method is much more robust against data manipulation. As reported by [20], CelebA vs CIFAR-10/100 is challenging for Ty-test. Our method can achieve 100% AUROC with batch size 10. We should point out that, although CelebA vs CIFAR-10/100 is not solved by the baseline method, our experimental results on CelebA vs others may be not fair for Ty-test. It is hard to make the likelihood distributions of CelebA train and test split fit well on the official Glow model [6] (see Figure 29 in the supplementary material for details). In principle, if the train and test split of ID data have coinciding likelihood distributions, the AUROC of Ty-test should not be less than around 50%. On the contrary, our GAD method is not affected by possible underfitting or overfitting.

Besides, KLODS outperforms Ty-test when batch size is smaller (i.e., \( 2 \sim 4 \)). See Table 11 in the supplementary material for details. Without splitting representations, KLODS needs a larger batch size than KLODS but still outperforms Ty-test. The results of KLOD are omitted.

**CIFAR-10 vs CIFAR-100** is one of the most challenging problems for Ty-test. KLOD and KLODS only achieve around 70% AUROC when batch size reaches 200. We argue the main reason is the model fails to capture the distribution of CIFAR-10 as other data sets. As shown in Figure 29 in the supplementary material, the model does not succeed to generate meaningful images. Thus, \( D(q_Z, p_Z) \) is not small enough and our theoretical analysis does not fit well in this situation. Currently, we are not aware of any unconditional flow-based model that can generate high-quality CIFAR-10-like images. We argue that it is meaningless to require an OOD detection method to achieve strong results on a failed generative model. In such a case, even the model itself generates “OOD data” that differ from the training set.

**Robustness.** The results presented above have demonstrated the robustness of our method against data manipulation method M2. KLODS achieves the same performance under M1 except that a slightly larger batch size (+5) is needed for CIFAR-10-related problems. The results are omitted for brevity.

**GAD on GlowGMM.** We train GlowGMM on FashionMNIST. For each component, we use learnable mean \( \mu_i \) and diagonal covariance \( \text{diag}(\sigma_i^2) \). We treat each class as ID data and the rest classes as OOD data. As shown in Table 3, KLODS can achieve near 100% AUROC for all cases when batch size is 25. On the contrary, Ty-test is worse than random guessing in most cases.

It is clear that likelihood under each component is also not qualified for OOD detection. See Section D in the supplementary material for more details. Recent works have improved the accuracy of conditional Glow on classification problems [23], [72]. However, as long as GlowGMM does not achieve 100% classification accuracy, the question proposed in Section I remains.

**Generating OOD images using GlowGMM.** In Section 4, we have shown that unconditional Glow can generate blurred images like OOD data set with fitted Gaussian. In GlowGMM, we can generate high-quality OOD images with

4. We stop training after 2,000 epochs.
TABLE 2
GAD Results of KLODS on Glow with batch sizes 5 and 10.

| Batch size | m=5 | m=10 |
|-----------|-----|-----|
| Method    | AUROC | AUPR | AUROC | AUPR | AUROC | AUPR |
| KLODS     |       |      |       |      |       |      |
| Ty-test   |       |      |       |      |       |      |
| Constant  | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Uniform   | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| Uniform-C(0.08) | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| CelebA    | 99.7 | 99.7 | 99.7 | 99.7 | 99.7 | 99.7 |
| CelebA-C(0.08) | 99.7 | 99.7 | 99.7 | 99.7 | 99.7 | 99.7 |
| CIFAR-10  | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| CIFAR-10-C(0.12) | 97.5 | 97.5 | 97.5 | 97.5 | 97.5 | 97.5 |
| CIFAR-100  | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 |
| CIFAR-100-C(0.12) | 96.9 | 96.9 | 96.9 | 96.9 | 96.9 | 96.9 |
| ImageNet32-C(0.07) | 97.5 | 97.5 | 97.5 | 97.5 | 97.5 | 97.5 |

TABLE 3
GAD results on GlowGMM trained on FashionMNIST.

| Batch size | m=25 |
|-----------|-----|
| Method    | AUROC | AUPR | AUROC | AUPR |
| KLODS     |       |      |       |      |
| Ty-test   |       |      |       |      |
| Constant  | 100.0 | 100.0 | 100.0 | 100.0 |
| Uniform   | 100.0 | 100.0 | 100.0 | 100.0 |
| Uniform-C(0.02) | 100.0 | 100.0 | 100.0 | 100.0 |
| CelebA    | 99.2 | 99.2 | 99.2 | 99.2 |
| CelebA-C(0.3) | 84.3 | 84.3 | 84.3 | 84.3 |
| ImageNet32-C(0.3) | 72.0 | 72.0 | 72.0 | 72.0 |
| SVHN      | 97.5 | 97.5 | 97.5 | 97.5 |
| SVHN-C(2.0) | 100.0 | 100.0 | 100.0 | 100.0 |

5.2.2 Pad Results

We use the SOTA PAD method applicable to flow-based model as the baseline. In [70], the authors modify the official Glow model by using zero padding and removing ActNorm layer. In principle, the baseline method should not be affected by such modification to models. Since the authors did not release their model checkpoint, we reimplement the baseline method using the original Glow model [68]. We also use FLIF [74] as the compressor which is considered as the best compressor in [70]. However, we find that the baseline method did not reach the performance reported in [70]. In [70], the authors did not explain why they modified the official model. We are not aware of why the performance of baseline degenerates on the official Glow model.

GAD on VAE. We train convolutional VAE with 8-/16-/32-dimensional latent space on FashionMNIST, SVHN, and CIFAR-10, respectively. The latent space is too small, so we did not split representations and only use KLOD in experiments. As shown in Table [5] KLOD achieves 98.8%+ AUROC when m = 25 for almost all problems. CIFAR-10 vs CIFAR-100 is also the most difficult problem on VAE. KLOD needs a batch size 150 to achieve 98%+ AUROC. See Table [13] in the supplementary material for details. Nevertheless, KLOD still outperforms Ty-test. Again, Ty-test can be attacked by data manipulations M2. As pointed out by existing work, for vanilla VAE the reconstruction probability is not a reliable criterion for OOD detection [73].

Mixture of OOD data sets. We also use the mixture of two data sets as one OOD data set. In such problems, we can treat samples from multiple distributions as from a mixture of distributions. Table 4 shows the results of KLODS when OOD data set is a mixture of two of the three data sets: SVHN, CelebA, and CIFAR-10. We randomly choose 5,000 samples from each data set and get 10,000 samples in total. Our method outperforms the baseline significantly.

GAD on VAE. We train convolutional VAE with 8-/16-/32-dimensional latent space on FashionMNIST, SVHN, and CIFAR-10, respectively. The latent space is too small, so we did not split representations and only use KLOD in experiments. As shown in Table [5] KLOD achieves 98.8%+ AUROC when m = 25 for almost all problems. CIFAR-10 vs CIFAR-100 is also the most difficult problem on VAE. KLOD needs a batch size 150 to achieve 98%+ AUROC. See Table [13] in the supplementary material for details. Nevertheless, KLOD still outperforms Ty-test. Again, Ty-test can be attacked by data manipulations M2. As pointed out by existing work, for vanilla VAE the reconstruction probability is not a reliable criterion for OOD detection [73].

5.2.2 Pad Results

We use the SOTA PAD method applicable to flow-based model as the baseline. In [70], the authors modify the official Glow model by using zero padding and removing ActNorm layer. In principle, the baseline method should not be affected by such modification to models. Since the authors did not release their model checkpoint, we reimplement the baseline method using the original Glow model [68]. We also use FLIF [74] as the compressor which is considered as the best compressor in [70]. However, we find that the baseline method did not reach the performance reported in [70]. In [70], the authors did not explain why they modified the official model. We are not aware of why the performance of baseline degenerates on the official Glow model.

CIFAR-10 vs Others. Table 6 shows PAD results on CIFAR-10 vs others. Compared with the results reported in [70], our method outperforms the baseline method only on CIFAR-10 vs TinyImageNet. Compared with the reimple-
with lower contrast (complexity). For these several problems, Table 7 shows the results on SVHN vs others. All problems vs CIFAR-100. The likelihoods of the train and test split of CelebA do not fit well. This means that KLODS does not always achieve high AUROC in these problems. KLODS is comparable with the baseline.

SVHN vs Others. In [70], although the authors claim that the baseline method can detect OOD data with more complexity than ID data (roughly Category II problems), they did not evaluate their method on such problems thoroughly. Table 7 shows the results on SVHN vs others. All problems in the top half of Table 7 are Category II problems. KLODS can achieve 98.8%+ AUROC and outperforms the baseline. The bottom half of Table 7 shows the results of OOD data with lower contrast (complexity). For these several problems, KLODS is comparable with the baseline.

CelebA vs others. We have also conducted experiments on CelebA vs others, which are not evaluated in [70]. As shown in Table 8, KLODS outperforms the baseline. We notice that KLODS does not always achieve high AUROC in all cases. We think the reason is similar to that for CIFAR-10 vs CIFAR-100. The likelihoods of the train and test split of CelebA do not fit well. This means that $K L(p|p^i)$ is not small enough. Unlike on GAD, KLODS on PAD is affected by the possible underfitting or overfitting on CelebA.

6 Discussion

Normality of representations. The normality of ID and OOD representation facilitates our theoretical analysis and implementation in OOD detection algorithm on flow-based model. In our experiments, we find that the normality of OOD representation is a widely existing phenomenon under flow-
Based model. We observe even representation of random noise has a p-value greater than 0.05. We are investigating the underlying reason. Most importantly, our method performs even better on Category II problems although the criterion computes the lower bound of the KL divergence.

Both Flow-based model and VAE are trained to minimize KL divergence between p_Z and prior. It seems natural that ID representations should follow Gaussian-like distribution. However, we did not observe such a phenomenon in VAE. In our experiments, we find that not all ID representations in VAE have p-values greater than 0.05. All OOD representations in VAE do not follow Gaussian-like distribution. We should note that our method on VAE does not have a divergence guarantee as that in flow-based model.

In principle, we can construct latents following any distribution and decode these latents to construct an OOD data set. Such data manipulation can be seen as an attack on the normality of representations. Note that, such manipulation does not make our OOD detection method fail necessarily.

**Limitations.** This work is an attempt to detect OOD data with divergence guarantee for flow-based model. The properties of flow-based model allow us to conduct deeper analysis. Our work has the following three limitations.

The first limitation is that our method requires the model to capture the distribution of training data. Modeling data is a long-standing goal of unsupervised learning \( \beta \). There are two possible solutions to handle CIFAR-10 vs CIFAR-100. The first one is to improve the model. Up to now, we have not tried more advanced flow-based models. \( 75, 76 \). We are not aware of any unconditional flow-based model that can model CIFAR-10 satisfactorily. The second possible solution is to use a more sensitive criterion to estimate KL divergence or dependence. For example, our answer 2 to Q1 reminds us to use the sample correlation of representations for GAD. We have tried to use the standard deviation of the set of non-diagonal elements of the correlation matrix as the criterion for GAD. Experimental results show that such criterion achieves 90%+ AUROC on CIFAR-10 vs CIFAR-100 with batch size 250, although it needs a larger batch size than KLODS on other problems. We leave this direction as future work.

Besides, when applied to PAD, KLODS sets higher demands on the model. For example, KLODS is affected by the discrepancy of the likelihoods of train and test splits on CelebA vs others.

The second limitation is that PAD performance may decrease when OOD data set has very low contrast (e.g., SVHN vs CelebA-C(0.08). Nevertheless, our method is still better than the baseline.

Finally, Ty-test applies to flow-based model, VAE, and auto-regressive model. Our method applies to models which learn independent or disentangled representations \( 59, 51, 22, 77, 78, 79, 80 \). Our method is not including auto-regressive model.

**Models.** We did not conduct more experiments on flow-based models with various architectures as well as other training methods. For VAE, our method is affected by the model architecture and training method. A high-dimensional latent space may contain nearly dead neurons. This may reduce the performance of our method. We did not conduct experiments on other VAE variations, e.g., \( \beta \)-VAE \( 80 \), Factor-VAE \( 50 \), \( \beta \)-TCVAE \( 51 \), and DIP-VAE \( 52 \). These variations add more regularization strength on disentanglement and hence have more independent representations than vanilla VAE \( 53 \). We also did not conduct PAD on VAE because the VAE models used in our experiments are small. We have not enough latent variables to split into multiple groups. In the future, we will conduct experiments on larger VAE models and variations.

**7 Related Work**

**OOD Detection.** In \( 2 \), Toth et al. give a survey on GAD methods and a list of real-world GAD applications. In \( 3 \), Chalapathy et al. survey a wide range of deep learning-based GAD and PAD methods. In \( 51 \), Pang et al. also review the deep learning-based anomaly detection methods. According to the availability of supervision information, OOD detection can be classified into three categories: supervised setting, semi-supervised setting, and unsupervised setting. In this paper, we focus on unsupervised OOD detection using flow-based model, so we mainly compare with methods in the same category.

Generally, it seems straightforward to use model likelihood \( p(x) \) (if any) of a generative model to detect OOD data \( 2, 52 \). However, these methods fail when OOD data have higher or similar likelihoods. Choi et al. propose using the Watanabe-Akaike Information Criterion (WAIC) to detect OOD data \( 18 \). WAIC penalizes points that are sensitive to the particular choice of posterior model parameters. However, Nalisnick et al. \( 20 \) point out that WAIC is not stable. Choi et al. also propose using typicality test in the latent space to detect OOD data. Our results reported in section 22 demonstrate that typicality test in the latent space can be attacked. Sabeti et al. propose based on typicality \( 83 \), but their method is not suitable for DGM. Nalisnick et al. propose using typicality test on model distribution (Ty-test) for GAD \( 20 \). Ren et al. propose to use likelihood ratios for OOD detection \( 71 \). Serra et al. propose using likelihood compensated by input complexity for OOD detection \( 70 \). Before this writing, \( 20 \) and \( 70 \) are the SOTA GAD and PAD methods applicable to flow-based models under unsupervised setting, respectively. We use them as the baselines in our experiments.

OOD detection can be improved with the help of an auxiliary outlier data set. Schirrmeister et al. improve likelihood-ratio-based method by the help of a huge outlier data set (80 Million Tiny Imagenet) \( 84 \). The method in \( 84 \) is not purely unsupervised learning due to the exposure to outliers in training as like \( 58 \). Besides, the huge outlier data set includes almost all the image classes in the testing phase. We did not compare with such methods due to different problem settings.

**Theoretical Analysis.** Previous works \( 26, 85 \) analyze the training objective of flow-based model in KL divergence form. We apply the property of diffeomorphism to investigate the divergences between distributions in flow-based models for OOD detection. We also propose new theorems on the properties of KL divergence between Gaussians for further analysis. Existing research has explored the upper bound of KL divergence in different settings \( 86, 87, 88, 89 \). To the best of our knowledge, we are not aware of similar work on the properties of KL divergence between
Gaussians. Theorems 3 and 4 can be used as basic theorems in information theory and machine learning fields.

In principle, GMM can approximate a target density better than a single Gaussian. We have tried to use GMM model to estimate $D_g$ (see Subsection 4.2), but find GMM is worse. We think the reasons are twofold: a) In our problems, $D_g[p_z]$ is Gaussian-like. It is not appropriate to use GMM for ID data. b) The batch size is too small to estimate the parameters of GMM.

Local pixel dependence. In [51], Kirichenko et al. reshape the representations of flow-based models to original input shape and analyze the induction biases of flow-based model. Their work reveals the reshaped representation manifests local pixel dependence. Our theoretical analysis from the divergence perspective allows strong dependence for OOD data. We also show that the representations with raw shape also manifest local pixel dependence.

Classification of problems. We classify OOD problems into Category 1 and 2 according to the variance of data sets. This criterion is roughly similar to the complexity used in [70]. See Figure 30 in the supplementary material for details.

8 Conclusion

In this paper, we prove theorems to investigate the divergences in flow-based models. Based on these theorems, we answer the question of why cannot sample out OOD AUROC for all GAD problems and robust against data manipulation in almost all cases. For PAD, our method also outperforms the baseline.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM Comput. Surv., vol. 41, no. 3, Jul. 2009.
[2] E. Toth and S. Chawla, “Group deviation detection methods: A survey,” ACM Comput. Surv., vol. 51, no. 4, Jul. 2018.
[3] R. Chalapathy and S. Chawla, “Deep learning for anomaly detection: A survey,” 2019.
[4] K. Muandet and B. Schölkopf, “One-class support measure data configurations in flow-based models. Based on these theorems, we prove theorems to investigate the divergences in flow-based models. Based on these theorems, we answer the question of why cannot sample out OOD AUROC for all GAD problems and robust against data manipulations. On the contrary, the SOTA GAD method performs not better than random guessing for challenging problems and can be attacked by data manipulation in almost all cases. For PAD, our method also outperforms the baseline.

8 Conclusion

In this paper, we prove theorems to investigate the divergences in flow-based models. Based on these theorems, we answer the question of why cannot sample out OOD data from two perspectives. We observe the normality of ID representations and OOD representations in flow-based model for a wide range of problems. Our theoretical analysis and key observation inspire us to perform GAD by KL divergence. We decompose the KL divergence further to improve our method and support PAD as well. Experimental results demonstrate our method can achieve very strong AUROC for all GAD problems and robust against data manipulations. On the contrary, the SOTA GAD method performs not better than random guessing for challenging problems and can be attacked by data manipulation in almost all cases. For PAD, our method also outperforms the baseline.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM Comput. Surv., vol. 41, no. 3, Jul. 2009.
[2] E. Toth and S. Chawla, “Group deviation detection methods: A survey,” ACM Comput. Surv., vol. 51, no. 4, Jul. 2018.
[3] R. Chalapathy and S. Chawla, “Deep learning for anomaly detection: A survey,” 2019.
[4] K. Muandet and B. Schölkopf, “One-class support measure data configurations in flow-based models. Based on these theorems, we prove theorems to investigate the divergences in flow-based models. Based on these theorems, we answer the question of why cannot sample out OOD AUROC for all GAD problems and robust against data manipulations. On the contrary, the SOTA GAD method performs not better than random guessing for challenging problems and can be attacked by data manipulation in almost all cases. For PAD, our method also outperforms the baseline.

8 Conclusion

In this paper, we prove theorems to investigate the divergences in flow-based models. Based on these theorems, we answer the question of why cannot sample out OOD data from two perspectives. We observe the normality of ID representations and OOD representations in flow-based model for a wide range of problems. Our theoretical analysis and key observation inspire us to perform GAD by KL divergence. We decompose the KL divergence further to improve our method and support PAD as well. Experimental results demonstrate our method can achieve very strong AUROC for all GAD problems and robust against data manipulations. On the contrary, the SOTA GAD method performs not better than random guessing for challenging problems and can be attacked by data manipulation in almost all cases. For PAD, our method also outperforms the baseline.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM Comput. Surv., vol. 41, no. 3, Jul. 2009.
[2] E. Toth and S. Chawla, “Group deviation detection methods: A survey,” ACM Comput. Surv., vol. 51, no. 4, Jul. 2018.
[3] R. Chalapathy and S. Chawla, “Deep learning for anomaly detection: A survey,” 2019.
[4] K. Muandet and B. Schölkopf, “One-class support measure data configurations in flow-based models. Based on these theorems, we prove theorems to investigate the divergences in flow-based models. Based on these theorems, we answer the question of why cannot sample out OOD AUROC for all GAD problems and robust against data manipulations. On the contrary, the SOTA GAD method performs not better than random guessing for challenging problems and can be attacked by data manipulation in almost all cases. For PAD, our method also outperforms the baseline.

8 Conclusion

In this paper, we prove theorems to investigate the divergences in flow-based models. Based on these theorems, we answer the question of why cannot sample out OOD data from two perspectives. We observe the normality of ID representations and OOD representations in flow-based model for a wide range of problems. Our theoretical analysis and key observation inspire us to perform GAD by KL divergence. We decompose the KL divergence further to improve our method and support PAD as well. Experimental results demonstrate our method can achieve very strong AUROC for all GAD problems and robust against data manipulations. On the contrary, the SOTA GAD method performs not better than random guessing for challenging problems and can be attacked by data manipulation in almost all cases. For PAD, our method also outperforms the baseline.

References

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey,” ACM Comput. Surv., vol. 41, no. 3, Jul. 2009.
[2] E. Toth and S. Chawla, “Group deviation detection methods: A survey,” ACM Comput. Surv., vol. 51, no. 4, Jul. 2018.
[3] R. Chalapathy and S. Chawla, “Deep learning for anomaly detection: A survey,” 2019.
Definition 3 \((h, \phi)-\text{divergence}\) The \((h, \phi)-\text{divergence}\) between two densities \(p(x)\) and \(q(x)\) is defined by:

\[
D_{h,\phi}(p, q) = h(D_{\phi}(p, q)),
\]

where \(h\) is a differentiable increasing real function from [0,0) to \([0, \infty)\) onto [0, \infty) \([93]\).

\(h, \phi\)-divergence includes a broader range of divergences than \(\phi\)-divergence. For example, \(\text{Rényi} \) distance belongs to \((h, \phi)\)-divergence family.

### APPENDIX B

#### TABLE 9

| Divergence | \(x \log x - x + 1\) | \(- \log x + x - 1\) |
|------------------|-----------------|-----------------|
| \(\phi\)-divergence | Total Variation Distance | Squared Hellinger distance |
| Minimum Discrimination Information | Jensen-Shannon divergence |

#### TABLE 10

| \(\varepsilon\) | 0.001 | 0.005 | 0.01 | 0.05 | 0.1 | 0.5 |
|-----------------|-------|-------|------|------|-----|-----|
| \(\sup\)       | 0.001 | 0.006 | 0.011 | 0.069 | 0.016 | 1.732 |

### APPENDIX C

#### MODEL DETAILS

We use both DeepMind and OpenAI’s official implementations of Glow model. The model consists of three stages, each of which contains 32 coupling layers with width 512. After each stage, the latent variables are split into two parts, one half is treated as the final representations and another half is processed by the next stage. In our experiments, we use only the output of the last stage with shape \((4,4,48)\) as representation. We use additive coupling layers for grayscale data sets and CelebA and use affine coupling layers for SVHN and CIFAR-10. We find no difference between these two coupling layers for OOD detection. All priors are standard Gaussian except for CIFAR-10, which has learned mean and diagonal covariance. All models are trained using Adamax optimization method with a batch size of 64. The learning rate is increased from 0 up to 0.001 in the first 10 epochs and keeps invariant in remaining epochs. Flow-based models are very resource consuming. We train Glow on FashionMNIST/SVHN/CelebA32 for 130/390/2000 epochs respectively. For fairness, we use the checkpoint released by DeepMind [68] for CIFAR-10. We have also conducted experiments using the checkpoints released by OpenAI [67] for CIFAR-10 vs others. The results are similar.

For VAE, we use convolutional architecture in the encoder and decoder. The encoder consists three \(4 \times 4 \times 64\) convolution layers. On top of convolutional layers, two dense layers output the mean \(\mu(x)\) and the standard variance \(\sigma(x)\) respectively. The decoder has the mirrored architecture as encoder. All activations are LeakyReLU with \(\alpha = 0.3\). For FashionMNIST, SVHN, and CIFAR-10, we use \(8\)-, \(16\)- and \(32\)-dimensional latent space respectively. Models are trained
TABLE 11
GAD Results of KLODS on Glow with batch sizes 2 and 4.

| IDi | OODi | Batch size | m=2 | m=4 |
|-----|------|------------|-----|-----|
|     |      | Method     | AUROC | AUPR | AUROC | AUPR | AUROC | AUPR | AUROC | AUPR |
| Fash. |       | KLODS      |       |      |       |      |       |      |       |      |
|       |       |            |       |      |       |      |       |      |       |      |
|       |       | Ty-test    |       |      |       |      |       |      |       |      |
|       |       |            |       |      |       |      |       |      |       |      |
|       |       |            |       |      |       |      |       |      |       |      |
| SVHN |       |            |       |      |       |      |       |      |       |      |
|       |       |            |       |      |       |      |       |      |       |      |
|       |       |            |       |      |       |      |       |      |       |      |
|       |       |            |       |      |       |      |       |      |       |      |
|       |       |            |       |      |       |      |       |      |       |      |
|       |       |            |       |      |       |      |       |      |       |      |

APPENDIX D
MORE EXPERIMENTAL RESULTS

D.1 GAD Results on Glow

KLODS. Table 11 shows GAD results of KLODS with batch size 2 and 4.

GlowGMM. Table 12 shows the results of using p(z) for 1 vs rest classification on FashionMNIST with GlowGMM. p(z) is a bad criterion for OOD detection.

Figure 6(a) shows the generated images using noise sampled from the Gaussian components \( N_i(\mu_i, \text{diag}(\sigma_i^2)) \) as prior. The \( i \)-th column corresponds to the \( i \)-th Gaussian \( N_i \). Figure 6(b) shows the generated images using the similar operation in Section 4.1.1. For each \( i \), we compute the representations of the \((i+1)\%10\)-th class and normalize them under \( N_i(\mu_i, \text{diag}(\sigma_i^2)) \) as \( z' = (z - \mu_i) / \sigma_i \). We use the normalized representation to fit a Gaussian \( N_i(\bar{\mu}, \Sigma_i) \). Then we sample \( z'' \sim N_i(\bar{\mu}, \Sigma_i) \), and compute \( f^{-1}(z'' + \sigma_i + \mu_i) \) to generate new images. As shown in Figure 6(b), we can generate almost high quality images of the \((i+1)\%10\)-th class from the fitted Gaussian.

In Section 1, we have shown that the centroids of components are close to each other. The results shown in Figure 6(b) show that correlation of representation is more critical than the norm in GlowGMM.

D.2 GAD Results on VAE

Table 13 shows the GAD results on convolutional VAE trained on CIFAR10 vs CIFAR100/Imagenet32.

Table 14 shows the results of using reconstruction probability \( E_{z \sim q}(\log p_\theta(x | z)) \) for OOD detection in VAE.

APPENDIX E
Figures

using Adam without dropout. The learning rate is \( 5 \times 1^{-4} \) with no decay.
Fig. 6. GlowGMM with 10 components trained on FashionMNIST. (a) sampling from \( \mathcal{N}(\mu_i, \text{diag}(\sigma_i^2)) \). The \( i \)-th column corresponds to Gaussian \( \mathcal{N}_i \). (b) For the \( i \)-th Gaussian \( \mathcal{N}_i \), we fit another Gaussian \( \tilde{\mathcal{N}}_i(\tilde{\mu}_i', \tilde{\Sigma}_i') \) using the normalized representations (by parameters of \( \mathcal{N}_i \)) of inputs of the \((i + 1)\%10\)-th class. The \( i \)-th column shows images generated from \( \tilde{\mathcal{N}}_i \).
Fig. 7. Train GlowGMM on FashionMNIST. The \(i\)-th subfigure shows the histogram of log-probabilities of 10 centroids under the \(i\)-th Gaussian component. All log-probabilities are close to \(768 \times \log(1/\sqrt{2\pi}) \approx -705.74\), which is the log-probability of the center of 768-dimensional standard Gaussian. These results indicate these centroids are close to each others.

Fig. 8. Train Glow on FashionMNIST and test on MNIST and notMNIST. We scale the representations of OOD dataset to the typical set of prior Gaussian. The scaled latent vectors still corresponds to clear (a) handwritten digits or (b) letters.

Fig. 9. Glow trained on FashionMNIST. Histogram of \(\log p(x)\). We can manipulate the likelihood distribution of OOD dataset by adjusting the contrast. “\(-C(k)\)” means the dataset with adjusted contrast by a factor of \(k\).
Fig. 10. Glow trained on SVHN. Histogram of $\log p(x)$. We can manipulate the likelihood distribution of OOD dataset by adjusting the contrast. "$-C(k)$" means the dataset with adjusted contrast by a factor of $k$.

Fig. 11. Glow trained on CIFAR10. Histogram of $\log p(x)$. We can manipulate the likelihood distribution of OOD dataset by adjusting the contrast. "$-C(k)$" means the dataset with adjusted contrast by a factor of $k$. For CIFAR10 vs CelebA, the range of $\log p(x)$ of CelebA is too large. For CIFAR10 vs Uniform, $\log p(x)$ of Uniform are too small.
Fig. 12. Glow trained on CelebA. Histogram of $\log p(x)$. We can manipulate the likelihood distribution of OOD dataset by adjusting the contrast. "-C(k)" means the dataset with adjusted contrast by a factor of $k$. It is hard to make the likelihoods of train and test split fit well on the official Glow model.

Fig. 13. (a) Train Glow on CelebA and sample from the fitted Gaussian of SVHN. (b) Train on FashionMNIST and sample from the fitted Gaussian of notMNIST. From top to down, the sampled noises from Gaussian are scaled by temperature 0, 0.25, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, respectively.
Fig. 14. Glow trained on CIFAR10. Generated images according to the fitted Gaussian from representations of (a) MNIST; (b) CIFAR100; (c) SVHN; (d) Imagenet32; (e) CelebA. We replicate MNIST into three channels and pad zeros for consistency. These results demonstrate that the covariance of representations contains important information of an OOD dataset.

Fig. 15. Glow trained on CelebA32×32, sampling according to (a) standard Gaussian; (b) fitted Gaussian from MNIST representations; (c) fitted Gaussian from CIFAR10 representations.

Fig. 16. Glow trained on FashionMNIST. Sampling according to prior (up), fitted Gaussian from representations of MNIST (middle) and notMNIST (down).
Fig. 17. Glow trained on FashionMNIST. Histogram of $\log p(z)$ of (a) FashionMNIST vs MNIST, (b) FashionMNIST vs notMNIST under Glow. The green part corresponds to the $\log p(z)$ of noises sampled from the fitted Gaussian of OOD datasets.

Fig. 18. Glow trained on FashionMNIST. Heatmap of correlation of FashionMNIST representations.

Fig. 19. Glow trained on FashionMNIST. Heatmap of correlation of MNIST representations.
Fig. 20. Glow trained on FashionMNIST. Heatmap of correlation of notMNIST representations.

Fig. 21. Glow trained on SVHN. Histogram of non-diagonal elements of correlation of representations.
Fig. 22. Glow trained on CIFAR10. Histogram of non-diagonal elements of correlation of representations.

Fig. 23. Glow trained on CelebA. Histogram of non-diagonal elements of correlation of representations.

Fig. 24. VAE trained on FashionMNIST. Heatmap of correlation of (a) FashionMNIST (b) MNIST (c) notMNIST representations. (d) Histogram of non-diagonal elements of correlation of sampled representations.
Fig. 25. VAE trained on SVHN. Histogram of non-diagonal elements of correlation of sampled representations.

Fig. 26. VAE trained on CIFAR10. Histogram of non-diagonal elements of correlation of sampled representations.
Fig. 27. Train Glow on SVHN and test on Imagenet32. We randomly select the 8-th channel. The subfigure at $i$-th row and $j$-th column shows the correlation between the pixel at position $(i,j)$ and all other pixels. Adjacent pixels tend to have stronger correlation.
Fig. 28. Examples of datasets and their mutations. Under Glow trained on CIFAR10, these mutated datasets have the similar likelihood distribution with CIFAR10 test split.
Fig. 29. Generated images from Glow trained on (a)FashionMNIST; (b)CIFAR-10; (c)CelebA32.

Fig. 30. The distributions of complexity estimated by the lengths of compressed files of data sets. We use FLIF as compressor and compute lengths in bits per dimension. Datasets with decreased contrast has lower complexity.