TOWARDS UNDERSTANDING IN-DISTRIBUTION AND OUT-OF-DISTRIBUTION OF DEEP LEARNING WITH DEEP GENERATIVE MODELS

RONGYU CHEN
Bachelor of Engineering

A PAPER SUBMITTED FOR THE QUALIFYING EXAMINATION
OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE
NATIONAL UNIVERSITY OF SINGAPORE
2021

Abstract

Although deep learning techniques have been developing rapidly in recent years, we still lack an understanding of their deployment. Reasonable generalization to only in-distribution (ID) distributions of deep learning models requires understanding and recognition of ID and Out-of-Distribution (OoD). Deep generative models have made a hit in modeling the underlying distribution and generating samples from it. Compared with other OoD detection methods such as discriminative and heuristic ones, the motivation and intuition of generative-based OoD detection methods are clear and convincing in high dimensions like image space. It only needs to threshold likelihood produced by Deep Generative Models (DGMs) to detect OoD data. However, researchers recently found DGMs counter-intuitively assign higher likelihood to OoD data. They discovered domain priors and model inductive biases to account for the phenomenon and came up with calibrations to the paradigm.

In this work, we found most of the existing work adopt weak generative backbones like VAEs and Flows. Weak DGMs have difficulty modeling underlying distributions and thus generate poor-quality images. Using these backbones to do OoD detection quite deviates from the motivation. Different from them, we experiment with strong generative models and reveal the benefits brought by the change, including significant alleviation of the problem without the need for any calibrations. This leads to a clear pipeline. Additionally, with the observation that DGMs do generate only ID data, we think they are knowledgeable of what ID is. We propose another manner of mining ID/OoD knowledge of the model which does not rely on likelihood. We believe this will encourage the community to think more about OoD detection using deep generative models.

Contents

1 Introduction **1**

1.1 Paper Statement 1

1.2 Overview and Organization 2

2 Related Work **4**

2.1 Introduction 4

2.2 Preliminaries 4

2.3 Deep Learning Generalization 4

2.4 Out-Of-Distribution Detection 6

2.4.1 Problem Definition 6

2.4.2 Generative-Based Method Overview 6

2.4.3 Generative Models 7

2.4.4 Likelihood-Based Methods 15

2.4.5 Other Class Methods 19

2.5 Chapter Summary 21

3 Proposed Methods and Experiments **22**

3.1 Introduction 22

3.2 Motivations and Assumptions 22

3.3 Experiments 23

3.3.1 Datasets 23

3.3.2 Evaluation Metrics 24

3.3.3 Implementation Details 24

3.3.4 Check for Generation Capacity 25

3.3.5 Adaptation of Batch Normalization Statistics 29

3.3.6 Component Analysis 30

3.3.7 ID/OoD Semantics in Latent Space 32

3.4 Chapter Summary 34

4 Conclusion and Future Work **36**

4.1 Insights 36

4.2 Future Work 37

4.3 Conclusion 38

References 39

Chapter 1

Introduction

1.1 Paper Statement

In a classical probabilistic perspective, deep learning models are trained on finite samples of an underlying distribution. Models have guaranteed generalization performance to a similar distribution. But it does not hold for other shifted distributions. The common case is that the model usually gets signals to guide for good training performance. When it comes to deployment to an open input, we do not know whether predictions are trustworthy or not. The case may be fatal where there are some unclear "bugs" in the training data and training of the model. For instance, the dataset has sampling biases of dogs always co-occurring on the grass. In supervised learning, the model guided with only image-level labels is likely to learn a shortcut of grass to classify the dog. In the test stage, it is no wonder that the model predicts any images with grass as a dog. This example gives rise to in-distribution and out-of-distribution detection. If we can reject this kind of images as OoD and only accept dogs on the grass, it will not be seen as a severe problem. People only need to take care of what is fed to the model and are able to ignore the shortcuts the model takes.

However, although OoD detection, or anomaly detection, has been widely studied for a long time, most studies are on relatively low-dimensional table data, sequential data, etc. These methods are not so effective in high-dimensional image space. In the era of deep learning, it is surprising that DL models are able to effectively extract low-dimensional abstract features. A mainstream of existing work made use of it to distinguish ID and OoD in feature space and achieved improved results.

Generative models have achieved great success recently. Researchers use deep generative models to model target distributions, and in results, they can draw as many samples from the model as they want. This motivates researchers to think of likelihood-based generative models as a natural and perfect choice to do OoD detection. As you already

have an approximation to the underlying mystery, aren't you confident dealing with ID? It becomes straightforward to threshold likelihood as ID likelihood is expected to be higher than OoD one.

However, recent work found it counter this intuition that likelihood-based generative models tend to assign higher likelihood to OoD data. This poses a question for researchers to rethink the pipeline using generative models. One mainstream point of view is domain inherent priors and model inductive biases, e.g., convolutions with local receptive fields take local pixel correlations. Researchers propose many patches to calibrate modeled likelihood in this line. There is also a voice doubting the inferiority of generative-based OoD detection methods to others, which hinders the application of this seemingly more convincing and intuitive method.

In this work, we challenge the use of generative models in the paradigm. We would make the following essential statement,

> It is more intuitive and efficient to use a strong generative backbone for OoD detection. Generative models do learn ID/OoD knowledge and likelihood is just the simplest way to mine it.

Our work is based on further inspection of existing work and the belief of rather natural and beautiful relationships between the generation and OoD detection task. DGMs help us further understanding what is ID and OoD in DL.

1.2 Overview and Organization

In the section, we introduce the organization of the following part of the paper. The second chapter is about related work to OoD detection. We will first specify terminologies and preliminaries used in the paper. To understand the meaning of OoD detection, we get to know the probabilistic framework of deep learning generalization. Among a variety of OoD detection methods, we focus on ones with the help of generative models, for we think they are convincing and intuitive. There are various deep generative models achieving huge success in modeling a complicated high-dimensional distribution. However, researchers found a higher OoD likelihood problem in the straightforward paradigm of thresholding likelihood modeled by generative models. Two of the most agreed causes are domain priors and model inductive biases.

In the third chapter, we mainly introduce our contributions of the paper. Our motivation is based on the observation that existing work unconvincingly adopted weak generative models to study the problem. It seems more reasonable to use a strong

generative baseline in the likelihood-based paradigm. This stimulates us to verify the proposition. We follow to introduce the details of implementations and evaluation of experiments. Several indications announce the empirical soundness of the appeal. On the other hand, noticing the role of likelihood as a kind of metric, its failures do not imply the defect of generative-based OoD detection methods. It is a matter of how to better mine the knowledge learned by the model as it does generate only ID data. We propose semantic testing towards better understanding of ID and OoD using DGMs.

In the fourth and last chapter, we summarize with the insights of the work and look forward to future work.

Chapter 2

Related Work

2.1 Introduction

In this chapter, we first introduce mathematics notations and preliminaries used in the paper. In Sec. 2.3, we describe the probably approximately correct framework of deep learning generalization, which gives a sense of the meaning of studying ID and OoD in DL. Following that, we introduce the task of out-of-distribution detection and the paradigm of using generative models in the problem. Sec. 2.4.3 gives a background of classical probabilistic generative models, including VAEs, Flows, EBMs, and GANs. We introduce the problem of likelihood-based methods and related work of analysis and solutions in Sec. 2.4.4. There are also other voices except for likelihood-based methods in this field (Sec. 2.4.5).

2.2 Preliminaries

We list utilized mathematics notations in Tab. 2.1 below.

2.3 Deep Learning Generalization

There is a popular Probably Approximately Correct (PAC) framework of machine learning. Given training samples $\{\mathcal{X}, \mathcal{Y}\}$ as a batch of independent and identically distributed (i.i.d.) random variables $\mathbf{X}$ drawn from an underlying distribution, $\mathcal{X}, \mathcal{Y} \sim \mathcal{D}$, the model takes a hypothesis $h = f_\theta : \mathcal{X} \to \mathcal{Y}$ defined by parameters θ from a candidate set H_Θ by optimizing an objective $\mathcal{R}$ on the samples. Generally, a Generalization Risk is defined by an expectation over $\mathcal{D}$, $\mathcal{R} = E_{X,Y\sim\mathcal{D}}[\ell(h(\mathbf{x}), y))]$, where a loss function ℓ is usually chosen as Mean Squared Error (MSE) and cross-entropy, etc. The risk on finite samples is called the Empirical Risk, $\frac{1}{|\mathcal{X}|}\sum_{\mathcal{X}} \ell(h(\mathbf{x}), y))$, an approximation to the intractable expectation. It is guaranteed that the generalization risk is close to the

empirical risk which we have access to in practice with a very high probability under some conditions [33]. Although deep learning is well-known for its overparameterized regimes, recent work shows similar frameworks are applicable to explain its generalization [48]. From this point, we aim at showing what In-Distribution (ID) is to deep learning generalization performance. It does not speak for Out-of-Distribution (OoD). If we consider the example in the introduction, the model recognizing the grass as the dog is problematic to be directly put into practice. But it will work reasonably if we only apply it to the distributions similar to on which it is trained. Therefore, towards a more safe deployment of deep learning, it is meaningful to detect and reject OoD input to the model.

Notation	Meaning
italic lowercase letters x, y	scalars
bold lowercase letters $\mathbf{x}, \mathbf{z}$	(flatten) vectors
x_i	components
bold uppercase letters $\mathbf{A}, \mathbf{X}$	matrices and random vector variables
X_i	random variables
italic or swash uppercase letters $P, Q, \mathcal{D}$	distributions
$P_{\mathbf{X}}(\mathbf{X})$ (ignoring $\mathbf{X}$)	(joint) Probability Cumulative Functions
$p_{\mathbf{X}}(\mathbf{x})$ (ignoring $\mathbf{X}$)	(joint) Probability Density (Mass) Functions
$\Pr(E)$	probability
θ, ϕ, ψ	parameters
italic lowercase letters f, g	scalar mapping functions
bold lowercase letters $\mathbf{f}, \mathbf{g}$	vector functions
swash uppercase letters $\mathcal{X}, \mathcal{Y}, \mathcal{B}$	space, datasets, and batch samples

Table 2.1: Mathematics notations used in the paper.

2.4 Out-Of-Distribution Detection

2.4.1 Problem Definition

The problem is somewhat "ill-posed". This is because usually, it is a zero-shot setting. We do not know what OoD distribution the model will encounter in the test in advance. What we have is only ID training data $\mathcal{X}_{train}$. It requires us to learn as complete ID features as possible and use them to reject OoD data. Thus, it can be seen as an "one-class" classification task. Output $y = f(\mathbf{x}) = 1$ indicates ID, otherwise not ID, i.e., OoD.

2.4.2 Generative-Based Method Overview

Deep generative models, another kind of deep learning models different from classic discriminative models used in the classification problem, have caught researchers' eyes and been developed rapidly in recent years, since Generative Adversarial Networks (GANs) [15] were proposed in 2014. After training on the training dataset, a deep generative model is able to generate amazing high-quality "realistic" images, e.g., in [39], [1] (Fig. 2.1). Basically, generative models aim at generating images that look very much like training data, at the aspect of content and styles (including perspectives and texture [24]), etc. In other words, they are trying to model underlying data distributions of which training data can be regarded as a batch sample. People then can use them to enlarge the training dataset by mimicking sampling from the underlying training data distribution like the collection of training data, e.g., to create CIFAR-5M from original CIFAR-10 [35]. It is exactly based on the reasonable assumption that generative models well model the data distribution.

Formally, from the perspective of probability, a generative model with parameters θ represents $p(\mathbf{x};\theta)$ on the input $\mathbf{x}$, regardless of in an explicit (e.g., flow) or implicit (e.g., GAN) way. Unlike discriminative models, generative models do not necessarily require the label y of data. Data with larger probability $p(\mathbf{x})$[1] is more likely to be sampled, and the probability of training data should be (relatively) large. Thus, the training objective is naturally to maximize $p(\mathbf{x};\theta)$ for training data, so-called Maximum Likelihood Estimation (MLE) of parameters. Usually in practice, researchers maximize its log version to prevent numerical problems and approximate generalization

[1] Actually more precisely, it should be integral over $\mathbf{x}$'s δ-neighborhood $\Omega = \{\mathbf{x} + \epsilon : \|\epsilon\| < \delta\}$, i.e., $P(\Omega) = \int_\Omega p(\mathbf{x} + \epsilon)d\epsilon$, rather than raw density. We do this for sake of simpler demonstrations.

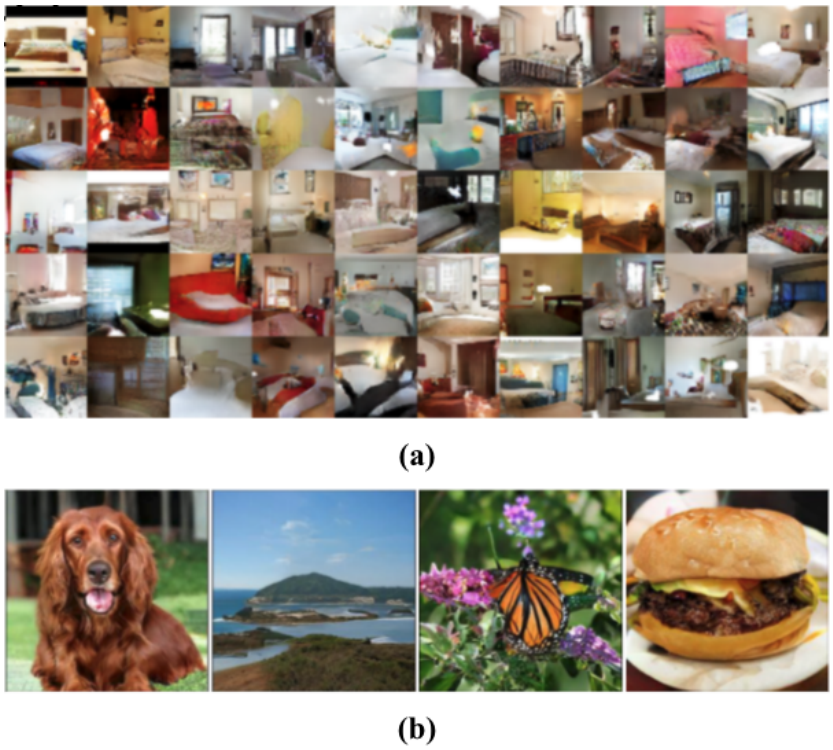


Figure 2.1: **(a)** Deep Convolutional GAN (DCGAN) generated bedroom images with resolution 64×64 [39]. **(b)** BigGAN generated images [1].

expectation over the training distribution, i.e.,

$$\max_{\theta} \frac{1}{|X|} \sum_{X} \log p(\mathbf{x};\theta) \rightarrow \max_{\theta} E_{\mathbf{X}\sim P_{gt}}[\log p(\mathbf{x};\theta)], \tag{2.1}$$

where $P_{gt}(\mathbf{X})$ is the underlying distribution of which in reality ones do not have access to the exact form but a batch sample X. Different methods and models more or less realize it in different ways which we will introduce in the following section.

After training, hopefully, one can draw samples from $p(\mathbf{x};\theta)$. Besides, the model can also be used for OoD detection because of our belief that ID data should have larger likelihood than OoD data in the training scheme.

2.4.3 Generative Models

Variational Autoencoders

Evolution from Autoencoders. Variational Autoencoders (VAEs) [25] can be regarded as an enhanced variant of autoencoders (AEs). Generally, AE is a clas-

sic method of self-supervised learning. It has the architecture of Encoder-Decoder (Fig. 2.2). The encoder encodes a high-dimensional input to a representation code $\mathbf{z}$. It is trained by the supervision of decoding $\mathbf{z}$ back to the input, e.g., Mean Squared Error (MSE) loss $mean(\|dec(\mathbf{z}) - \mathbf{x}\|_p)$, where $\mathbf{z} = enc(\mathbf{x})$, enc and dec represent the encoder and the decoder's nonlinear mapping, respectively. This is under the manifold hypothesis that data lie on a low-dimensional embedding manifold in the high-dimensional raw space. Thus, data can be compressed into codes by throwing away noise without losing key information which can be used for reconstruction. One classic method of OoD detection is reconstruction error thresholding. It is assumed that the model is incapable of extracting rich features from OoD inputs by the learned encoder as well as decoding the internal representations so it will get a higher reconstruction error on OoD data.

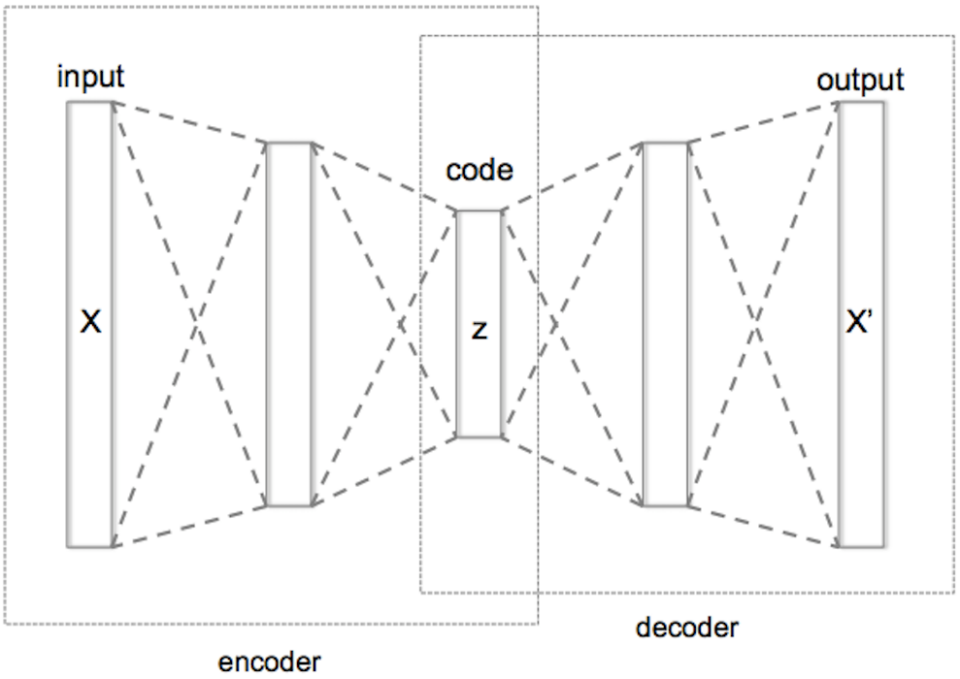


Figure 2.2: The Encoder-Decoder architecture. In general, the dimension of code $\mathbf{z}$ is smaller than that of input $\mathbf{x}$.

VAE is modified based on AE to do generative tasks. Though AE has a decoder, the decoder does not generalize well to those $\mathbf{z}$s other than training data's representations $enc(\mathbf{x})$s. VAE introduces random noise to the representation space to improve the robustness towards perturbations. Specifically, VAE constructs a Probabilistic Graphic Model (PGM) with a latent variable $\mathbf{z}$ controlling generation where the input $\mathbf{x}$ is an observed variable. Hence, $p_\theta(\mathbf{x};\theta) = \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z};\theta)p(\mathbf{z})d\mathbf{z}$, the prior $P(\mathbf{Z})$ can also be parametric. Usually, researchers model $P_\theta(\mathbf{X}|\mathbf{Z})$ with an easily computed

continuous distribution family, Gaussian with independent components, i.e., $P_\theta(\mathbf{X}|\mathbf{Z}) = \mathcal{N}(\mathbf{x}|dec_\theta(\mathbf{z}), \sigma^2\mathbf{I})$. Given dec is a complicated neural network, $P_\theta(\mathbf{X})$ can theoretically approximate arbitrarily complicated distributions. In practice, it is difficult to estimate and optimize $p_\theta(\mathbf{x})$ with classic Gradient Descent efficiently by directly sampling from $P(\mathbf{Z})$ which is usually chosen as the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ [11]. VAE introduces an approximate posterior $P_\phi(\mathbf{Z}|\mathbf{X}; \phi)$ via variation as follows,

$$\begin{aligned}
\log p_\theta(\mathbf{x}) &= \underset{\mathbf{Z}\sim P_\phi(\cdot|\mathbf{X})}{E}[\log p_\theta(\mathbf{x})] = \underset{\mathbf{Z}\sim P_\phi(\cdot|\mathbf{X})}{E}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p_\theta(\mathbf{z}|\mathbf{x})}\right] \\
&= \underset{\mathbf{Z}\sim P_\phi(\cdot|\mathbf{X})}{E}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p_\phi(\mathbf{z}|\mathbf{x})} + \log \frac{p_\phi(\mathbf{z}|\mathbf{x})}{p_\theta(\mathbf{z}|\mathbf{x})}\right] \\
&= \underset{\mathbf{Z}\sim P_\phi(\cdot|\mathbf{X})}{E}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(P_\phi(\cdot|\mathbf{X})\|P) + D_{KL}(P_\phi(\cdot|\mathbf{X})\|P_\theta(\cdot|\mathbf{X})) \\
&\geq \underset{\mathbf{Z}\sim P_\phi(\cdot|\mathbf{X})}{E}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(P_\phi(\cdot|\mathbf{X})\|P) = ELBO,
\end{aligned} \tag{2.2}$$

where the true posterior has a Probability Density Function (PDF) as $p_\theta(\mathbf{z}|\mathbf{x}) = \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{\int_\mathbf{z} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})d\mathbf{z}}$, and $D_{KL}(P||Q)$ is Kullback–Leibler (KL) divergence between the distribution P and Q defined by $\int_x p(x)\log\frac{p(x)}{q(x)}dx \geq 0$. $P_\phi(\mathbf{Z}|\mathbf{X})$ is usually chosen as Gaussian with a diagonal covariance matrix $\mathcal{N}(\mathbf{z}|\mu_\phi(\mathbf{x}), diag(\sigma^2_\phi(\mathbf{x})))$, where μ and σ are both neural networks, since KL divergence between two Gaussians can be efficiently computed in a closed form instead of estimation. Analogous to AE, $P_\phi(\mathbf{Z}|\mathbf{X})$ is the encoder while $P_\theta(\mathbf{X}|\mathbf{Z})$ is the decoder in VAE. Eq. 2.2 defines the main part of VAE's objective called Evidence Lower Bound which now can be optimized sufficiently to converge based on sampling a single (or a few) $\mathbf{z}$(s) from $P_\phi(\mathbf{Z}|\mathbf{X})$ every step [11].

By substituting PDFs with specific Gaussian distribution expressions, the terms inside ELBO's expectation can be written as,

$$\begin{aligned}
\mathcal{L} = -\frac{1}{2\sigma^2}\|\mathbf{x} - dec(\mathbf{z})\|^2 - D_{KL}(P_\phi(\cdot|\mathbf{X})\|P) + C, \\
\text{where } \mathbf{z} = \mu(\mathbf{x}) + \sigma(\mathbf{x})\epsilon,\ \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}),
\end{aligned} \tag{2.3}$$

where $C = \log \frac{1}{\sqrt{(2\pi\sigma^2)^{dim(\mathbf{z})}}}$ is a constant. It consists of two terms: an AE reconstruction loss and KL divergence. The KL divergence can be regarded as a regularization on the posterior which emerges naturally under the formulation. If the constraint on the KL divergence is too loose, the model will degenerate to AE where only point-level encoding and decoding are trained. Besides, from the perspective of the Information Theory, a too small KL divergence hinders $\mathbf{z}$ to contain sufficient information of $\mathbf{x}$ for reconstruction [21, 3].

A More Accurate Approximation. Though the above provides an effective variational training objective for VAE to generate images resembling training data, it is a lower bound with a gap of at least non-trivial $D_{KL}(P_\phi(\cdot|\mathbf{X})\|P_\theta(\cdot|\mathbf{X}))$ (Eq. 2.2)[2]. To obtain a tighter bound and accurate estimation of log-likelihood $\log p_\theta(\mathbf{x})$, Importance Weighted Autoencoder (IWAE) is deduced in another way – Importance Sampling with auxiliary $P_\phi(\mathbf{Z}|\mathbf{X})$ [2],

$$\begin{aligned}\log p_\theta(\mathbf{x}) &= \log \int_{\mathbf{z}} \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p_\phi(\mathbf{z}|\mathbf{x})} p_\phi(\mathbf{z}|\mathbf{x})d\mathbf{z} = \log \underset{\mathbf{z}\sim P_\phi(\cdot|\mathbf{X})}{E}\left[\frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p_\phi(\mathbf{z}|\mathbf{x})}\right] \\ &= \log \underset{\mathbf{z}_k\sim P_\phi(\cdot|\mathbf{X})}{E}\left[\frac{1}{K}\sum_k \frac{p_\theta(\mathbf{x}|\mathbf{z}_k)p(\mathbf{z}_k)}{p_\phi(\mathbf{z}_k|\mathbf{x})}\right] \\ &\geq \underset{\mathbf{z}_k\sim P_\phi(\cdot|\mathbf{X})}{E}\left[\log \frac{1}{K}\sum_k \frac{p_\theta(\mathbf{x}|\mathbf{z}_k)p(\mathbf{z}_k)}{p_\phi(\mathbf{z}_k|\mathbf{x})}\right]. \end{aligned} \tag{2.4}$$

There is no gap of KL divergence between the approximate posterior and true posterior any longer. It is exactly vanilla VAE ELBO when $K = 1$. Generally, the more samples drawn from the approximate posterior for calculation are, the better the bound approximates true log-likelihood [2]. But one thing worthy noticing as well is a tighter lower bound can hurt training by reducing the signal-to-noise ratio of gradient estimation [40].

Problems. Though the model is simple and effective and has solid theoretical foundations as well as good interpretation, the qualitative and quantitative performance of VAE is not competitive to that of GAN. Images generated by VAE tend to be blurred, which indicates generated images are out of the underlying data manifold (Fig. 2.3). One cause is the use of dimension-equal losses, e.g., MSE, making the model produce average predictions. Another cause is Gaussian posterior approximation is limited as a specific distribution family, thus it can deviate much from the intractable and complicated true posterior. Some works [22, 36] also claim VAE covers more regions than true modes of the data distribution are, since VAE does not constrain likelihood of OoD data to be small. In the following section, we will see the connection between VAE and other methods and the improvement of VAE by them.

[2]However, at least in the 1D case, VAE is proved to have zero approximation error as $\sigma \to 0$ [11].

Figure 2.3: The out-of-manifold problem of VAE [21]. Images are generated after VAE trains on CelebA.

Flow-Based Models

Autoregressive Flows. Another powerful class of generative models is Autoregressive Flows (AFs). AFs, as the name suggests, predict variables from themselves, such as typically predicting the current variable based on the history state of variables, i.e., Markov assumption. PixelCNN [50] is one of representatives of AFs. In PixelCNNs, it is assumed that each component x_i is conditional on the preceding components $x_{<i}$ except the first one, i.e., usually modelled by a continuous Gaussian $P_{\theta i}(X_i|\mathbf{X}_{<i}) = P_{\theta i}(X_i|X_1, X_2, \cdots, X_{i-1}) = \mathcal{N}(x_i|\mu_{\theta i}(\mathbf{x}_{<i}), \sigma^2_{\theta i}(\mathbf{x}_{<i}))$ for $i > 1$ or a discrete Categorical distribution $Cat(x_i|K = 256, \lambda_{\theta i}(\mathbf{x}_{<i}))$. PDF of the joint distribution is,

$$p_\theta(\mathbf{x}) = p_\theta(x_1) \prod_i p_\theta(x_i|\mathbf{x}_{<i}) \tag{2.5}$$

which can be exactly calculated unlike a variational lower bound in VAEs. P_θ is called flow here. Extensive experiments have shown very strong approximation power of this kind of models. However, though training and inference of likelihood can be processed in parallel, it takes much time to generate images in sequence, i.e., x_1 first followed by x_2 followed by x_3, etc.

Figure 2.4: The quality of generated images. **(a)** PixelCNNs on CIFAR-10 [50]. **(b)** Glows on CelebA-HQ. **(c)** Glows on CIFAR-10 [26].

Normalizing Flows. (Finite) Normalizing Flows (NFs) which are closely related to AFs catch researchers' eyes in Glow [26] for it can generate very high-quality images (Fig. 2.4). There is also a latent variable $\mathbf{z}$ in PGM of NFs. Specifically, NFs transform the latent variable $\mathbf{z}$ of a simple prior distribution to obtain the observed variable $\mathbf{x}$ of a complicated distribution by a series of invertible mappings, namely, $\mathbf{x} = \mathbf{g}_\theta(\mathbf{z}),\ \mathbf{z} = \mathbf{g}_\theta^{-1}(\mathbf{x}),\ \mathbf{g}_\theta = \mathbf{f}_{\theta_K} \circ \mathbf{f}_{\theta_{K-1}} \circ \cdots \circ \mathbf{f}_{\theta_1}$. According to change of variables, we have,

$$\begin{aligned} \log p_\theta(\mathbf{x}) &= \log p_\theta(\mathbf{z}) \left| \det \left(\frac{d\mathbf{z}}{d\mathbf{x}} \right) \right| \\ &= \log p_\theta(\mathbf{g}^{-1}(\mathbf{x})) + \sum_k \log \left| \det \left(\frac{d\mathbf{h}_{k-1}}{d\mathbf{h}_k} \right) \right|, \end{aligned} \tag{2.6}$$

$$\text{where } \mathbf{z}\ (\mathbf{h}_0) \xleftrightarrow{\mathbf{f}_1} \mathbf{h}_1 \xleftrightarrow{\mathbf{f}_2} \mathbf{h}_2 \longleftrightarrow \cdots \xleftrightarrow{\mathbf{f}_K} \mathbf{x}\ (\mathbf{h}_K).$$

Actually, an analogous encoder $P_\theta(\mathbf{X}|\mathbf{Z})$ and decoder (true posterior) $P_\theta(\mathbf{Z}|\mathbf{X})$ are now modelled by Dirac distribution $\delta(\mathbf{x} - \mathbf{g}(\mathbf{z}))$ and $\delta(\mathbf{z} - \mathbf{g}^{-1}(\mathbf{x}))$ instead of Gaussians in VAEs[3]. Besides, notice, when $K = 1$ and $x_i = f_{1i}(\mathbf{z}) = \sigma_i(\mathbf{f}_{1<i}(\mathbf{z}))z_i + \mu_i(\mathbf{f}_{1<i}(\mathbf{z}))$, $z_i = {f_1}^{-1}{}_i(\mathbf{x}) = (x_i - \mu_i(\mathbf{x}_{<i}))/\sigma_i(\mathbf{x}_{<i})$, it is exactly aforementioned AFs [38] (Eq. 2.5). When $x_i = f_{1i}(\mathbf{z}) = \sigma_i(\mathbf{z}_{<i})z_i + \mu_i(\mathbf{z}_{<i})$, $z_i = {f_1}^{-1}{}_i(\mathbf{x}) = (x_i - \mu_i({\mathbf{f}_1^{-1}}_{<i}(\mathbf{x})))/\sigma_i({\mathbf{f}_1^{-1}}_{<i}(\mathbf{x}))$, it is Inversed Autoregressive Flows (IAFs) [27] which are instead fast in generation

[3]Actually, Dirac distribution is the limit of Gaussians as the variance $\sigma \to 0$.

but slow in training (Fig. 2.5(a-c)). A simpler design of flow to accelerate speed is only conditioning a half once each flow step and stacking to enhance representation power, e.g., Affine Coupling Layers (ACL), $\mathbf{h}_{iB} = \mathbf{s}_i(\mathbf{h}_{i-1A}) \otimes \mathbf{h}_{i-1B} + \mathbf{t}_i(\mathbf{h}_{i-1A})$, $\mathbf{h}_{iA} = \mathbf{h}_{i-1A}$, where A and B are an even separation of dimensions for more extensive interaction [9, 10, 26] (Fig. 2.5(d)). These all ensure the Jacobian matrix is a lower triangular matrix so the additional determinant term is easy to compute.

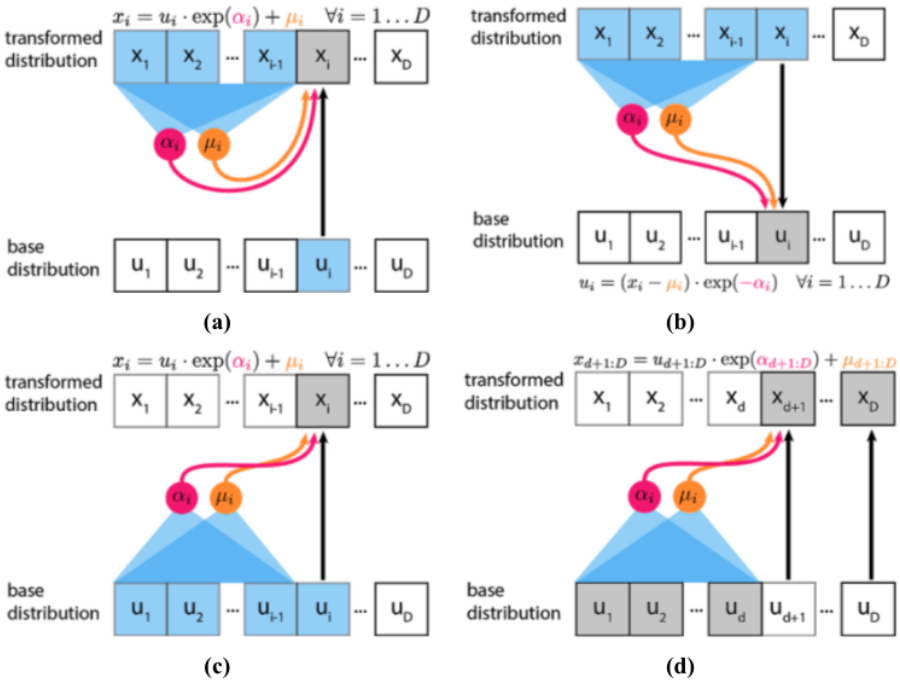


Figure 2.5: Flows different in the generative process $\mathbf{z} \to \mathbf{x}$ and inference process $\mathbf{x} \to \mathbf{z}$ (notion $\mathbf{u}$ is also used to denote the latent variable in flows). **(a-b)** The generative process and inference process of the Masked Autoregressive Flow (MAF). **(c)** IAF. **(d)** RealNVP.

Improving VAEs with Flows. Flows attract researches by excellent representation power and exact calculation of likelihood but discourage them by training cost. There is waste in the design of constant dimensions which is required to achieve invertibility according to the low-dimensional manifold assumption. And due to the limited representation power of single coupling layer, researches usually need to stack many layers [26]. These make training inefficient. On contrast, variational inference of VAEs makes training relatively much faster but suffers inferior generation quality. Therefore, some works [27, 46] propose to combine them, e.g., to enhance the approximate posterior of VAEs by NFs. Specifically, for improved variational inference, new $p_\omega(\mathbf{z}|\mathbf{x}) = \int_{\mathbf{z}'} \delta(\mathbf{z} - \mathcal{F}(\mathbf{z}'|\mathbf{x})) p_\phi(\mathbf{z}'|\mathbf{x}) d\mathbf{z}'$, where NF $\mathcal{F}$ is only w.r.t. $\mathbf{z}'$, and the objective is still VAEs' ELBO (Eq. 2.2). In fact, if the decoder $dec : \mathbf{z} \mapsto \mathbf{x}$ is also designed as

invertible, then ELBO will become exact likelihood of flows, since from a Gaussian decoder $\mathbf{x} = dec(\mathbf{z}) + \sigma\epsilon,\ \epsilon \sim \mathcal{N}(\mathbf{0}, I)$, we can naturally obtain a flow encoder $\mathbf{z} = \mathcal{F}(\epsilon|\mathbf{x}) = dec^{-1}(-\sigma\epsilon + \mathbf{x})$ [46]. More general complicated mappings give Flow-VAEs stronger representation power reducing the number of stacked flow layers.

Prospect. As we will see later, VAEs and Flows[4] are the most popular backbones studied in likelihood-based OoD detection. They both have reasonable generation and are easily trained to model likelihood.

Energy-Based Models

Energy-Based Models (EBMs) are of a very classic kind of generative models and its idea can be seen behind many other designs. ELBO and exact (log) likelihood can be considered as two types of energy. Formally, EBMs are defined as,

$$p_\theta(\mathbf{x}) = \frac{e^{-U_\theta(\mathbf{x})}}{Z_\theta},\ Z_\theta = \int_{\mathbf{x}} e^{-U_\theta(\mathbf{x})} d\mathbf{x}, \tag{2.7}$$

where U_θ is called energy functions. By classic MLE, the gradient of the (negative) objective is,

$$\nabla_\theta(-E_{\mathbf{x}\sim P_{gt}}[\log p_\theta(\mathbf{x})]) = \underset{\mathbf{x}\sim P_{gt}}{E}[\nabla_\theta U(\mathbf{x})] - \underset{\mathbf{x}\sim P_\theta}{E}[\nabla_\theta U(\mathbf{x})] \to 0. \tag{2.8}$$

Surprisingly, we find the two separate expectations over the underlying distribution and modelled distribution respectively, are actually not only pushing down ID data's energy, but also pulling up OoD data's energy, to approximate on the distribution level. This is what VAEs and flows miss, partly leading to the problem such as the OoD high likelihood problem (Sec. 2.4.4). In view of the partition function Z_θ is intractable, different methods were proposed to estimate $E_{\mathbf{x}\sim P_\theta}[\nabla_\theta U(\mathbf{x})]$ approximately with reduced variance, e.g., Markov Chain Monte Carlo (MCMC) sampling, Noise-Contrastive Estimation (NCE) [17], and adversarial learning [30, 15] introduced as follows.

Generative Adversarial Networks

GANs [15] are usually formulated as a two-player min-max adversary. A forger (generator) is trying to forge fake data while a discriminator is trying to discriminate these generated fake data from true real data. They are progressing together. Formally,

[4] Non-autoregressive one is out-of-scope.

a typical GAN defines a generator P_θ and a discriminator P_ϕ with the objective,

$$\min_\theta \max_\phi \underset{\mathbf{x}\sim P_{gt}}{E}[\log p_\phi(\mathbf{x})] + \underset{\mathbf{x}\sim P_\theta}{E}[\log(1-p_\phi(\mathbf{x}))], \tag{2.9}$$

where $p_\theta(\mathbf{x}) = \int_{\mathbf{z}} \delta(\mathbf{x}-G(\mathbf{z}))p(\mathbf{z})d\mathbf{z}$, G is a deterministic generative mapping. It is noticed that the likelihood is hard to compute for GANs unlike others. There are indeed some other formulations and understanding of GANs, e.g., f-Divergence [37] and variation [22] perspective. Kumar et al. [30] connected GANs with EBMs. Specifically, we can use easily-sampled GAN's generator (now denoted by $P_{\theta'}$) to approximate intractable P_θ in Eq. 2.8 under a constraint of a KL Divergence with the real distribution $D_{KL}(P_{\theta'}\|P_\theta)$. Thus, it is equivalent to Eq. 2.9 to some extent,

$$\begin{cases} \min_\theta E_{\mathbf{X}\sim P_{gt}}[U_\theta(\mathbf{x})] - E_{\mathbf{X}\sim P_{\theta'}}[U_\theta(\mathbf{x})] \Leftrightarrow \max_\theta E_{\mathbf{X}\sim P_{gt}}[\log p_\theta(\mathbf{x})] - E_{\mathbf{X}\sim P_{\theta'}}[\log p_\theta(\mathbf{x})] \\ \min_{\theta'} D_{KL}(P_{\theta'}\|P_\theta) \Leftrightarrow \min_{\theta'} -H(P_{\theta'}) + E_{\mathbf{X}\sim P_{\theta'}}[\log p_\theta(\mathbf{x})], \end{cases} \tag{2.10}$$

where $H(P_{\theta'})$ is entropy $-\int_{\mathbf{x}} p_{\theta'}(\mathbf{x})\log p_{\theta'}(\mathbf{x})d\mathbf{x}$ usually considered to avoid the mode collapse and missing problem in GANs [15, 42]. Adversarial training of GANs brings strong approximation power to the underlying data distribution [24] as well as training difficulty [15, 42]. Besides, in comparisons with VAEs and flows, GANs generally do not have an traditional encoder mapping input to the latent space though discriminators can behave analogously to encoders [12, 13].

2.4.4 Likelihood-Based Methods

Raw Likelihood Assumption

A Higher OoD Likelihood Problem. Since a PDF must sum up to 1, it is regarded that limited resource will be distributed to training data in priority. Based on the assumption that ID data have higher likelihood, one of the most straightforward OoD detection methods is setting a threshold to distinguish ID and OoD (Fig. 2.6(a)). Flows are naturally picked for modelling exact likelihood. However, Nalisnick et al. [36] and Choi et al. [5] early find flows can assign higher likelihood to OoD data (Fig. 2.6(b)). Nalisnick et al. further observe the phenomenon is asymmetric or uni-directional. For instance, it holds when models are trained on CIFAR-10 and tested on SVHN but not the reverse. With the insight from a rough analysis of a difference population expectation $E_P[\log p_\theta(\mathbf{x})] - E_Q[\log p_\theta(\mathbf{x})]$, they empirically conclude the distribution difference is due to different statistic variances of datasets, regardless of model parameter choices (Fig. 2.7(a)). Therefore, they propose graying

ID images to shrink the variance and increase likelihood to alleviate the problem. Serrà et al. [44] also similarly verify the correlation between individual likelihood and image complexity (image variance is one factor of it). Images of simpler datasets tend to have higher likelihood, e.g., X-MNIST low-resolution gray images (Fig. 2.7(b)). They then propose to consider the factor by estimating image complexity scores $C(\mathbf{x})$ with image compressors and compensating likelihood, i.e., $IC(\mathbf{x}) = \log p_\theta(\mathbf{x}) + C(\mathbf{x})$.

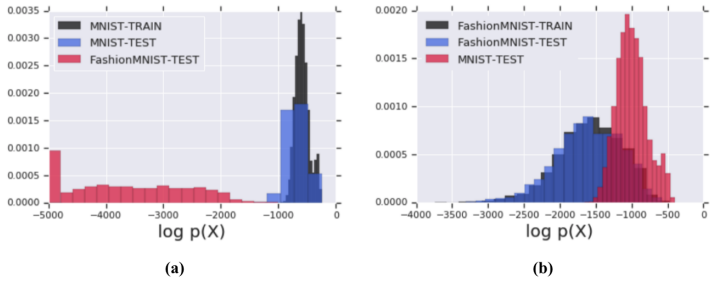


Figure 2.6: **(a)** Likelihood-based methods and **(b)** their OoD high likelihood problem [36]. A model trained on **(a)** MNIST and **(b)** Fashion-MNIST respectively.

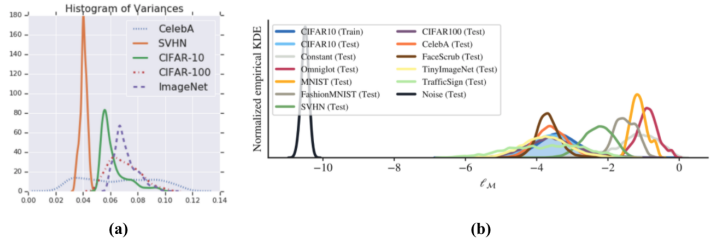


Figure 2.7: Experimental evidences for the (a) dataset variance and (b) image complexity hypothesis for the OoD high likelihood problem [36, 44]. **(a)** Variance: SVHN $<$ CIFAR-10 $<$ ImageNet. **(b)** Likelihood negatively correlated to image complexity: noise $<$ Tiny ImageNet $<$ CIFAR-10 $<$ SVHN $<$ X-MNIST $\approx$ constant.

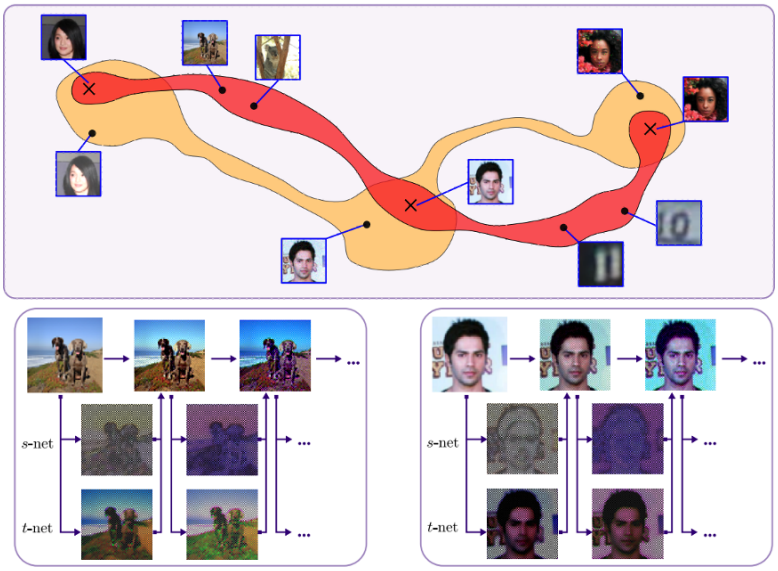
Likelihood Ratios. Another observation of input image complexity is images with simple background are regarded as simple as well, e.g., plain back background in MNIST gray-scale images. Simple background of large area can dominate generative modelling and make itself a salient ID feature. Hence, Ren et al. [41] learn a background model P_{θ_B} by training on perturbed data in practice, based on the assumption that semantic foreground will be corrupted while background will not be affected much and can still be learned well. With the help of the background model, they propose a likelihood ratio to remove the effects of excessive background modelling, i.e., $LLR_{BG}(\mathbf{x}) = \log \frac{p_\theta(\mathbf{x})}{p_{\theta_B}(\mathbf{x})} = \log p_\theta(\mathbf{x}) - \log p_{\theta_B}(\mathbf{x})$. Actually, from the likelihood ratio perspective, $C(\mathbf{x})$ in IC can be also regarded as $p_C(\mathbf{x}) \propto 2^{-C(\mathbf{x})}$.

Model Inductive Biases. Noticing flows use equal-dimension invertible transformations which actually preserve much information of images in the raw pixel space, Kirichenko et al. [28] found coupling layers induce the biases of leveraging local pixel correlations, e.g., local smoothness (Fig. 2.8). They think the cues are shared across natural image datasets, and hence it partially explains the observed high OoD likelihood problems. They give insights on the shift term $\mathbf{t}(\mathbf{x}_{ID})$ of likelihood in coupling layers. Flows can accurately predict the shift term on OoD data just like ID data models train on. Apart from designing a new kind of coupling layers to remedy the problem, they circumvent direct modeling on the pixel space but on an intermediate extracted feature embedding space. In this way, inductive biases of models are about semantics, making it easier to tell apart ID and OoD data as the human does. Schirrmeister et al. report these model biases on low-level and high-level features in different level layers as well [43]. They similarly find replacing classical convolutions with fully connected layers help calibrate model inductive biases and hence alleviate the problem. An alternative is to induce domain prior knowledge by adopting an aforementioned likelihood ratio between an ID model and general distribution when general OoD data are available, e.g., 80 M Tiny Images. The technique is also termed by Outlier Exposure [19, 32] requiring an auxiliary or prior OoD distribution. Things left are more specific to the ID distribution after model inductive biases are controlled. Model inductive biases are not exposed in ID regions since the model is optimized there so that they are relatively consistent with underlying regularities, but they become visible and dominate predictions when inputs deviate from those regions.

Perspective Summary. Modeled Likelihood is a synthesis result of data domains and model inductive biases where calibrations usually are considered.

Bottleneck Compression Assumption

Compressed autoencoders are even more popular in OoD detection. There is limited intermediate representation ability given, so the architecture design forces the model to learn essential information of data and only allow them to pass an analogous bottleneck. It is expected that the model extracts shared features among ID data, e.g., eyes and hair, which are likely not applicable to other domain distributions, hence leading to good reconstructions on ID data and bad reconstructions on OoD data. VAE is an enhanced variant of AEs, also based on the reconstruction assumption. Even though it is, Nalisnick et al. [36, 5] also early reported the same higher OoD likelihood problem in VAEs, more specifically, in terms of approximate likelihood ELBO instead of exact calculation. Choi et al. [5] further point out consistent observations even if using a



(b) ImageNet input, in-distribution (c) CelebA input, OOD

Figure 2.8: An illustration of model inductive biases, e.g., Flows leverage local pixel correlations [28]. **(a)** One possibility may be that Flows treat natural images as a manifold while humans recognize human faces with tolerant perturbations given the CelebA dataset as ID. **(b-c)** The visualization of s and t-net on ID and OoD distributions when the Flow model is trained on ImageNet. t-Net predicts the structure well on both distributions.

simple ensemble. Nevertheless, the assumptions and designs of VAEs make them seem a better choice than alternatives. Empirically, variational training gives more flexibility of approximation, compared with the case that each data point is required to optimize in MLE of exact likelihood like Flows [29]. Moreover, the unconstrained objective of Flows can lead to extreme parameters which may have unexpected generalization, e.g., increasingly growing scales $\mathbf{s}$ in the volume term $|\det(\mathbf{J}_\mathbf{z})|$ (Eq. 2.6) [36, 28]. Che et al. [4] indicated the help of the disentanglement regularization $D_{KL}(P_\phi(\cdot|\mathbf{X})\|P_\mathbf{Z})$ in the latent space which inherently involves an adversarial mechanism between the left part of the reconstruction objective $E_{P_\phi}[\log p_\theta(\mathbf{x}|\mathbf{z})]$ as mentioned before (Para. VAEs). Recently, Daniel et al. [6, 23] also made use of the relatively stronger encoder $P_\phi(\mathbf{Z}|\mathbf{X})$ of VAEs by further discriminating the KL divergence between real ID and fake generated samples, i.e., used the encoder as a discriminator as well (Fig. 2.9). They found their models, a GAN enhanced VAE, not only improve image generation quality like GANs but also solve the higher OoD likelihood problem without the need

of any post-hoc calibrations[5]. As seen in the previous subsection, VAEs can also be remedied by EBMs to increase model capability. Xiao et al. [52] used VAEs as a base distribution jointly modeled with an energy distribution, $p_{\theta,\psi}(\mathbf{x},\mathbf{z}) \propto p_\theta(\mathbf{x},\mathbf{z})e^{-U_\psi(\mathbf{x})}$, and showed EBMs help VAEs remove spurious modes in OoD regions and improve OoD detection performance. Compared to Flow backbones, VAEs seem more handleable to be enhanced from the source to produce a better PDF, despite the lack of exact analysis.

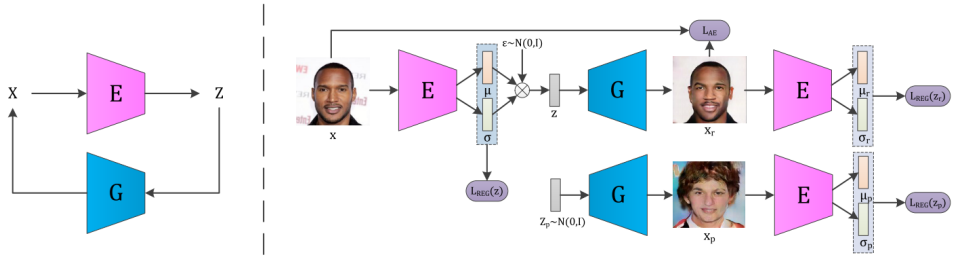


Figure 2.9: An overview of IntroVAE's framework [23].

2.4.5 Other Class Methods

Typicality in High Dimensions. Although likelihood-based OoD detection is quite intuitive, it does exist problems, especially in the high dimensional space. In a low dimensional space like 2D and 3D, a higher likelihood well explains the probability of being sampled. As the dimension increases, all likelihood will become very small, e.g., the density round of a d-dim uniform distribution $\mathcal{U}(a,b)^d$ decreases exponentially as $\int_{[-\delta,\delta]^d} \frac{1}{(b-a)^d} = \left(\frac{2\delta}{b-a}\right)^d$, $\frac{2\delta}{b-a} \ll 1$. It is not surprising that OoD data can have higher likelihood, especially when their supports occupy only very small regions in the space, e.g., simpler OoD MNIST compared with ID Fashion-MNIST [28, 55]. According to the Weak Law of Large Numbers, samples drawn from the model distribution actually fall into a typical set determined by the entropy. For instance, for a simple standard normal distribution $\mathcal{N}(\mathbf{0},\mathbf{I})$, samples likely concentrate in an annulus of a bandwidth $O(d^{\frac{1}{4}})$ at a radius $\sqrt{d}$. The center origin $\mathbf{0}$ with exponentially higher and the highest density is never sampled. Based on the observation that probability mass is more essential than probability density itself in the setting of OoD detection, Nalisnick et al. [36, 5] used a statistic goodness-of-fit testing, with the zero hypothesis $H_0 : |s(\mathcal{B}) - H(P)| < \epsilon$, to detect OoD groups $\mathcal{B}$. Unfortunately, there is no theoretical guarantee and experimental verification of applying to individual OoD detection as

[5]Here, the likelihood estimation of GAN-VAEs is kept the same as VAEs.

estimation bias trades off with variance [34]. Additionally, the definition of typicality should also be cautious, e.g., any substitutions of finite members or small parts does not affect the whole mass and thus the set can be made to include OoD data [55]. Although how to leverage typicality for individual OoD detection is still an emerged open question, we will provide our own perspectives towards likelihood and typicality in the following chapter (Sec. 3.3.7).

Others. We will not further category in detail but just introduce some of other existing branch work briefly to give an overview to the field.

Distance-Based Distance is another popular score metric used for OoD detection. It naturally comes with unsupervised representation clustering. We tend to trust predictions more on regions close to training data on the manifold in the representation space. This complements our manifold learning like reconstruction-based methods. Researchers usually learn latent metric space where distance can be easily computed or post-model representations with simply assumed Gaussians $\mathcal{N}(\mu, \Sigma)$, which is equivalent to an application of Mahalanobis distance [8, 51].

Discriminative-Based OoD detection is actually an one-class classification. If OoD data is exposed, one will straightforwardly train a binary classifier. However, a more common way is performing on discriminative feature representations learned by powerful DNNs on other tasks, e.g., image classification. Some existing work shows a direct combination with Mahalanobis distance metric achieves state-of-the-art performance [14]. It is well known that discriminative models are much more easily trained than generative models, our subject, while supervisions are requirements. It is equally notorious for infinite shortcuts and biases to take under the unconstrained learning paradigm.

Bayesian Uncertainty Bayesian methods are good at uncertainty modeling, which is a process of updating the posterior from the prior after observing data. It may seem intuitive that higher uncertainty is corresponding to OoD [20]. On the other hand, people also expect the model to give higher uncertainty when input data are deviate from and not covered by training data [18]. The thought of introducing priors can generally benefit other class methods [7]. Appropriate priors are able to enhance both prediction accuracy as well as trustworthiness, while bad ones can deteriorate the performance of the main task.

2.5 Chapter Summary

In this chapter, we introduce the meaning of the OoD detection task from deep learning generalization. We describe the motivation of using generative models to do OoD detection and derive a likelihood-based paradigm. Generative models are good at modeling high-dimensional distributions. We introduce classical probabilistic generative models for readers to mind a basic background. However, recent studies found a higher OoD likelihood problem of the paradigm. By investigating root domain priors and model inductive biases, researchers proposed calibrations to likelihood to make the paradigm work again. For a rich command of the field, we also introduce other prevalent branches of OoD detection.

Chapter 3

Proposed Methods and Experiments

3.1 Introduction

In this chapter, we pose our proposed methods and the validation of experiments. We first make our motivations and assumptions clear in Sec. 3.2. Then, we introduce the datasets, evaluation metrics as well as implementation details of experiments in Sec. 3.3. In Sec. 3.3.4, we verify the idea of using a strong generative backbone for being more intuitive and efficient. Sec. 3.3.5 gives a BN technique that is easy to plug and use to alleviate the higher OoD likelihood problem. It can be seen as an enhancement of the backbone model. In Sec. 3.3.6, detailed component analysis of likelihood is provided to complement assumed reconstruction intuitions. At the end of the chapter, we propose another point of view to mine ID/OoD semantics knowledge encoded in latent space.

3.2 Motivations and Assumptions

The paradigm of likelihood-based OoD detection is inspired by a thought. We have witnessed the power and success of deep generative models like GANs [15] and Glow [26]. Given they are able to generate images very analogous to training data, it is rational to think they well model the ID distribution. One naturally would like to make use of the knowledge of generative models to do OoD detection since they know what they generate are only ID, e.g, a CelebA trained model will not generate a handwriting digit forever. If the generative model is able to produce a PDF which training and sampling are following (Sec. 2.4.3, so-called likelihood-based generative models), thresholding the likelihood as an ID/OoD metric becomes a natural choice. Therefore, the paradigm forms up.

However, existing work found a higher OoD likelihood problem (Sec. 2.4.4), i.e., deep generative models are found to assign higher likelihood on OoD data for which they do

not train. The first glance at the finding is somewhat counter-intuitive and surprising. Researches analyzed several causes of the phenomenon and proposed corresponding solutions, covering data domains and model inductive biases. Calibrated likelihood of existing work can also be regarded as another OoD score metric.

In this paper, we explore the interesting phenomenon from different perspectives on our own. We discover existing work problematically studied on weak generative models. These generative models, e.g., vanilla VAEs and small Glows, originally fail to model complicated real-word data distributions like LSUN and CIFAR-10 (Sec. 3.3.4). This is against the primary motivation and assumption of the likelihood-based OoD detection paradigm, which is generative models well model ID distributions. "To do a good job, an artisan needs the best tools". Although many fixes found in existing work help us understand more about the OoD detection problem, we think it may not be necessary to consider these to achieve reasonable OoD detection performance. Our experiments show a better generative model backbone spontaneously alleviates the higher OoD likelihood problem to a large extent without any calibrations. This also more convincingly fits the motivation and paradigm. Furthermore, towards better understanding ID and OoD properties, we mine the knowledge about ID and OoD learned by expressive generative models in latent space (Sec. 3.3.7). These are all supported by our core belief that powerful generative models do learn ID concepts.

3.3 Experiments

3.3.1 Datasets

We introduce datasets used in the experiments as follows.

CelebA CelebFaces Attributes Dataset (CelebA) is a large-scale face attributes dataset with more than 200K celebrity images. The images in this dataset have large pose variations and background clutter.

SVHN SVHN is a real-world image dataset similar to MNIST. It is obtained from house numbers in Google Street View images. There are more than 70K training images of size 32×32.

CIFAR-10 The CIFAR-10 dataset consists of 60K 32×32 colour natural images evenly in 10 classes, including airplanes, automobiles, and birds, etc.

LSUN LSUN contains images of 10 scene categories and 20 object categories, e.g., bedrooms and churches.

Fashion-MNIST Fashion-MNIST consists of a training set of 60K samples and a test set of 10K samples. Each sample is a 28×28 grayscale clothing image making it a hard version of MNIST.

3.3.2 Evaluation Metrics

There are several common metrics for measuring the generation and OoD detection task.

Bit-Per-Dimension (BPD)/Negative Likelihood (NLL) Likelihood-based generative models have a natural built-in metric for model capacity and generation quality, i.e., likelihood. In the information theory, log-likelihood $\log p(\mathbf{x})$ also has the meaning of the length of code required to describe the data, a constant scaling of so-called Bit-Per-Dimension (BPD). The higher, the better, which means richer information in the model distribution. In our experiments, if the exact likelihood is intractable, we approximate it using a tight ELBO called IWAE (Sec. 2.4.3). We pick the number of samples for estimation $K = 5, 50$ and find results are consistent as continuously increasing K.

Fréchet Inception Distance (FID) and Inception Score (IS) FID and IS are two metrics widely used for quantifying the quality of images generated. Intuitively, it measures the reality of generated images by some feature distances of a pre-trained discriminative deep learning model, usually chosen as Inception V3. Some statistics such as the mean and variance are calculated on real ID training data and generated images, respectively, for comparison. More powerful generative models have lower scores.

AUCROC The full name is Area Under the Curve of Receiver Operating Characteristic, a metric for OoD detection. It both considers two types of errors, False Positive and False Negative ($FNR = 1 - TPR$). The value is independent of the selection of thresholds, so it reveals the quality of predictions itself. In our case, it is likelihood. The higher, the better. It varies from 0 to 1 and random choice achieves 0.5.

3.3.3 Implementation Details

Data Pre-Processing. As noted in [47, 36, 28], the underlying image distribution is actually quantized to be discrete. When DGMs are modeling a continuous distribution, the calculation of entropy should be cautious. The convention is to simply apply small

amount of noise to the training distribution to bound entropy with differential entropy, typically by data augmentation. Although it is also reported to help stabilize training of GANs, we find the results are consistent no matter whether we do that. We do not adopt other data augmentations except for this. To prevent overfitting, we use the validation dataset to inspect the model and apply early stopping. In most experiments, if without specification, we resize and crop the data into size 32×32.

Neural Network Architectures and Training. In our early experiments as well as literature, we find one-to-one invertible Flows seem not as flexible for enhancement, compared to another class of backbones, VAEs. As discussed in Sec. 2.4.4 of the last chapter, they have different mechanisms. The compressed bottleneck design forces the model to capture more essential features in training data, while Flows requiring equal-dimension mappings are likely to take general inductive biases. Hence, we currently focus more on VAEs of bottleneck compression designs and leave the study on Flows as future work. We adopt the classical and light DCGAN architecture [39] for both the VAE baseline and the enhanced one. The only difference is that enhanced GAN-VAE uses adversarial training to learn discriminative posteriors $P_\phi(\mathbf{Z}|\mathbf{X})$ and sharp image generation (Sec. 2.4.4). We use the technique proposed in [6] to stabilize training and converge smoothly.

3.3.4 Check for Generation Capacity

VAE Blur and Flow Chaos. We mentioned this in the previous section (Sec. 2.3). VAEs tend to generate blurred images, which is notorious for VAE study. In both existing work and our experiments, we find it mainly results from i) inherent trade-offs between reconstruction and KL divergence regularization and ii) compressed latent space $dim(\mathbf{z}) < dim(\mathbf{x})$ (Fig. 3.1). For i), the objective of a β-VAE is $\frac{1}{2\sigma^2} reconst + \beta KLD$ (Eq. 2.3). It is sort of equivalent to increasing the weighting of KL divergence β which is preferred for representation disentanglement and tolerant variance σ^2 of the reconstruction decoder distribution $P_\theta(\mathbf{X}|\mathbf{Z}) = \mathcal{N}(\mathbf{x}|dec(\mathbf{z}), \sigma^2\mathbf{I})$. For ii), while the bottleneck is forcing the extraction of useful information, it also limits the representation power of the model. As we can see, recent state-of-the-art VAEs tend to adopt a large latent space for the preservation of details [49]. In contrast, Flows are observed to generate demonizing and distorted images, although they achieved fancy high-fidelity generation on some datasets like human face CelebA (Fig. 2.4). We start from a Flow-VAE hybrid model, f-VAE [46] (Sec. 2.4.3) which uses expressive Flows to enhance VAEs to improve generation quality. As the relative weighting of KL divergence increases and the latent

space dimension decreases, generated images become blurred. It seems the unweighted pixel-wise L-p (e.g., L1 and L2) reconstruction metric may contribute to blur in a non-trivial way since the original f-VAE also optimizes it (Eq. 2.6).

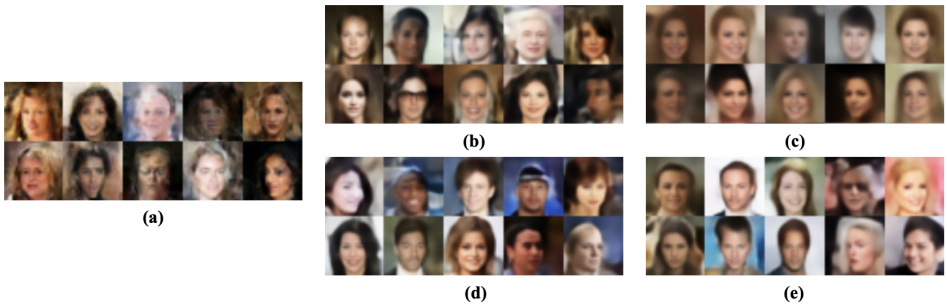


Figure 3.1: Two causes of VAE blur. **(a)** f-VAE [46] baseline trained, $dim(\mathbf{z}) = dim(\mathbf{x})$, temperature $T = 1.0$. **(b-e)** Unlearnable constant $\sigma^2 = 0.1, T = 1.0$. **(c)** Additional $\beta = 10$. **(d)** $dim(\mathbf{z}) = 128$. **(e)** $dim(\mathbf{z}) = 64$.

Correction of Existing VAE-Based Work. Here, we show unsatisfactory generation quality of some existing pioneer work uncovering the higher OoD likelihood problem in Fig. 3.2. What is worse, Xiao et al. [53] blamed the problem to VAEs for originally reconstructing OoD data well, e.g., VAEs trained on Fashion-MNIST are able to reconstruct OoD MNIST digits well. Given the poor generation of their VAE, we suspect this is originated from the degeneration to AEs (sharing same architectures). We validate, although AEs are designed based on the reconstruction assumption (Sec. 2.4.4), because of model inductive biases like leveraging local pixel correlations, they may be able to do good reconstruction on images they are not trained on and hence recognize these images as ID based on their assumptions [54] (Fig. 3.3). On the contrary, variation in latent space or generative training helps model on the problem by further regularizing representation (distribution)s $\mathbf{Z}$ to bind with training data to capture data modes, i.e., optimizing $p_\theta(\mathbf{x}|\mathbf{z})$ for all $\mathbf{z} = \mu(\mathbf{x}) + \sigma(\mathbf{x})\epsilon,\ \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, given a relative deterministic decoder (σ_θ^2 is small). Otherwise, it will not have good generation. In other words, we are kind of trading off good ID reconstruction with bad OoD reconstruction. Both of them are desirable characteristics for OoD detection. In this regard, we find by changing the decoder from a categorical distribution to Gaussian distribution, corresponding to a cross-entropy loss and sum squared error, VAEs come with perceptually better generation quality and do not suffer the higher OoD likelihood problem in the relatively simple Fashion-MNIST/MNIST pair (Fig. 3.4).

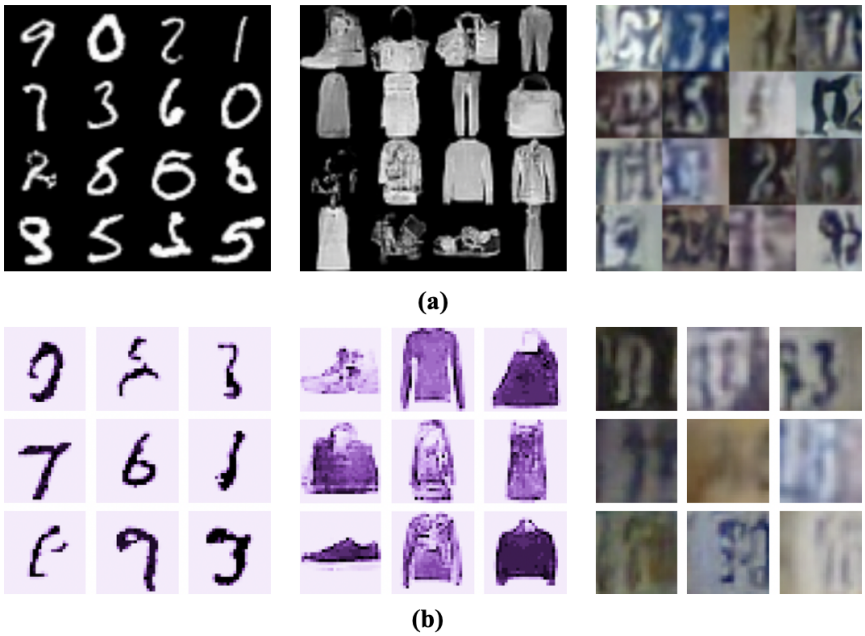


Figure 3.2: Poor generation quality of generative models which pioneer and latest existing work studied on. **(a)** Nalisnick et al. [36]. **(b)** Kirichenko et al. [28] on MNIST, Fashion-MNIST, and SVHN, respectively.

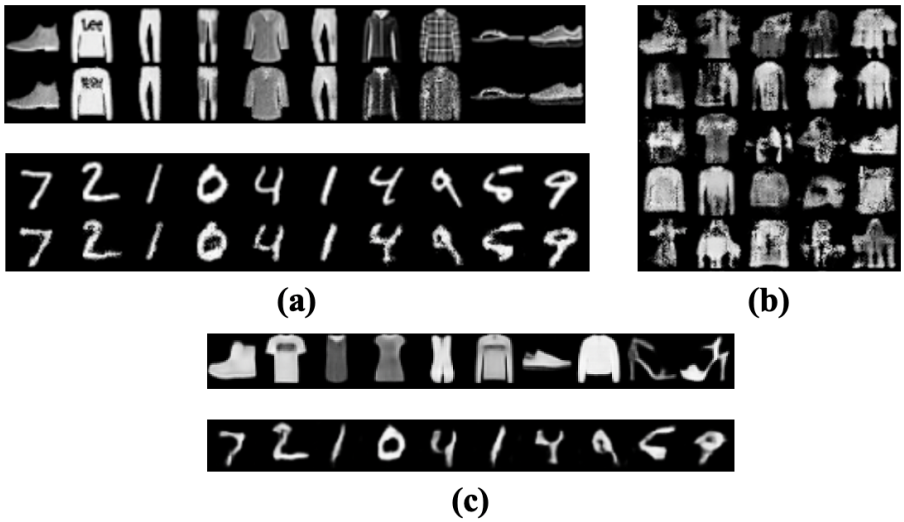


Figure 3.3: Good reconstruction on OoD data of Xiao et al.'s VAE [53] and our AE. **(a)** VAE trained on Fashion-MNIST (first two rows) is able to reconstruct OoD MNIST data well (the third and fourth row). **(b)** Random samples generated by their VAE. **(c)** We show AE, degenerate VAE incapable of image generation, also both reconstructs ID and OoD data well.

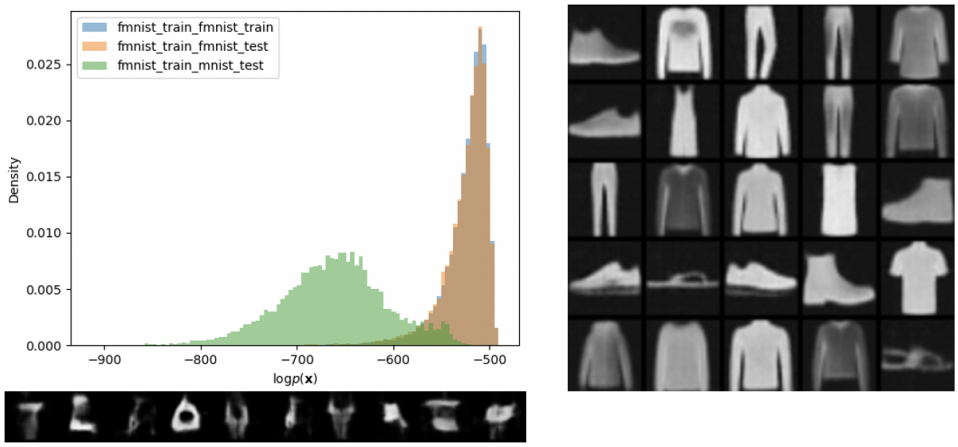


Figure 3.4: After applying a Gaussian decoder to improve generation quality, VAEs almost do not assign higher likelihood to OoD MNIST data. **(Left panel)** The histograms of log-likelihood on ID Fashion-MNIST and OoD MNIST data, respectively, ID $>$ OoD. Reconstruction is also shown below. It can be seen that VAEs are trying to reconstruct ID features such as clothing unlike just using local pixel correlations. **(Right panel)** Recognizable random samples drawn from the model.

Use of Enhanced VAEs. Because of model incapacity, vanilla VAEs still have similar problems on more complicated distributions, e.g., CelebA and SVHN. By looking into reconstruction, we find OoD reconstruction shares much common with OoD data in large-scale features like coarse shapes and colors, although it quite outperforms the AE baseline (Fig. 3.5 (a-c,e)). Hence, plus SVHN is much simpler than ID CelebA, due to an effect of reconstruction on likelihood similar to the previous examples, the likelihood is relatively higher on OoD than ID. We then apply a more powerful model, a GAN-VAE [6] (with the same architecture but different training, 4.16 FID on CIFAR-10) which generates much sharper and more diverse images. The results are shown in Fig. 3.6. Comparing OoD reconstruction on SVHN digits, GAN-VAEs make their efforts to reconstruct rather complete ID human faces despite sharing coarse color styles with OoD data, while vanilla VAEs are reconstructing color blobs. Due to the significant appearance difference between CelebA and SVHN, GAN-VAEs achieve larger OoD reconstruction and hence lower OoD likelihood. It is unbelievable for one to regard vanilla VAEs model ID distributions for so blurred images (Fig. 3.6 (c)). Generally, a strong generative model is qualified to learn a good encoder and representation manifold. Except for specific adversarial attacks, these will make the model map data close to ID regions and reconstruct ID features. Therefore, we highlight the significance of studying on stronger generative model baselines for convincing and promising results. There is no post-processing required. Here, we

mainly illustrate the effect of reconstruction on the final likelihood. We will conduct a detailed analysis of factors affecting likelihood in the later section 3.3.6.

Figure 3.5: Reconstruction on a bit complicated ID and OoD data. **(a-b)** CelebA test and SVHN data samples and corresponding reconstruction of VAE trained on CelebA. **(c)** LSUN OoD reconstruction of VAE trained on CelebA. **(d)** CelebA OoD reconstruction when VAE is trained on SVHN this time (ID and OoD are exchanged). **(e)** SVHN OoD reconstruction (CelebA ID) of AE and the reverse CelebA OoD reconstruction.

3.3.5 Adaptation of Batch Normalization Statistics

BN Cautions in GAN-VAE Practice. Actually, unlike discriminative models and common GANs, we find there is a trick to ensure GAN-VAEs work fine. Whether it is VAE or GAN, the input to the decoder (or generator in GANs) is assumed to be a consistent distribution during the training and the inference. For instance, we always feed a random normally distributed variable to GANs. However, things are different in this kind of hybrid models and not revealed in existing work in the context. In practice, GAN-VAEs are found to achieve better performance if reconstruction is also used together with random samples as part of fake samples to train. This occurs alternatively during each iteration so Batch Normalizations (BNs) in the decoder are adjusting to the average distribution between the reconstruction (also discriminator) $P_\phi(\mathbf{Z}|\mathbf{X})$ and the normal distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The problem exists due to the reconstruction is not close enough to the prior, and random sampling will face a distributional shift and produce unsatisfactory generation, e.g., dark visualization. The solution to the problem is simple. We just need to adjust the BN statistics to do a step of preliminary domain adaptation [31].

Train Mode of Encoder's BNs. With the lessons from the last section, we further think that adjusting BN statistics helps reconstruct ID distributions' features. Since

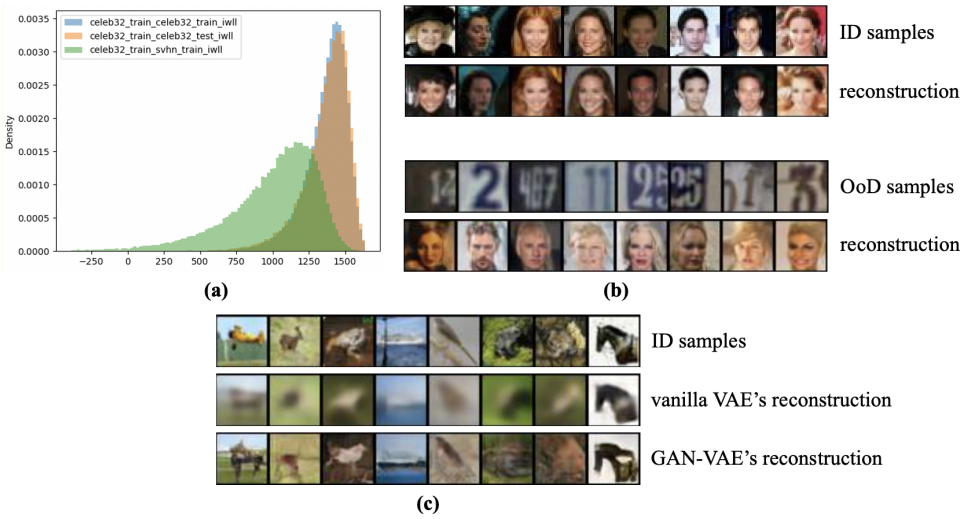
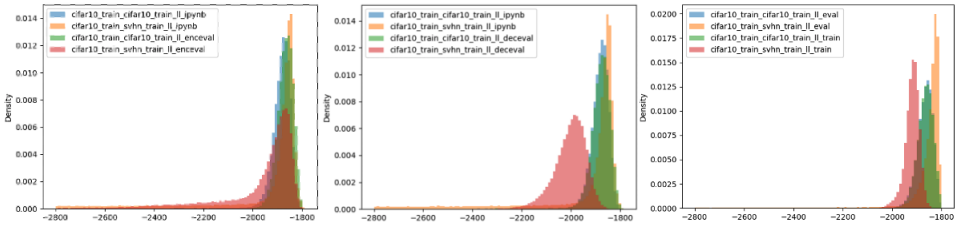


Figure 3.6: GAN-VAE experiments on complicated OoD detection. **(a)** Log-Likelihood: ID > OoD. **(b)** (Semantic) Reconstruction on ID and OoD data. It is trying to reconstruct ID human faces on SVHN digits. **(c)** Reconstruction of vanilla VAEs and GAN-VAEs on difficult CIFAR-10. GAN-VAEs far outperform blur VAEs.

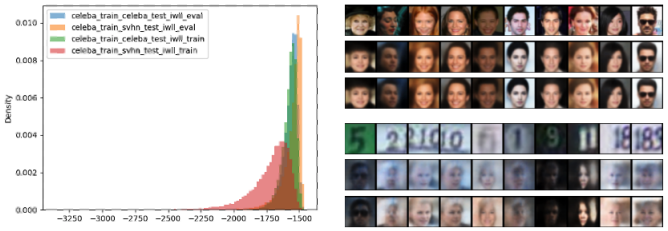
setting BNs as the evaluation mode is quite a standard, we question the opposite, whether a change of it makes differences, to make a fair comparison with the improvement brought by a strong GAN-VAE backbone. In the experiments, we find this action indeed alleviates the higher OoD likelihood problem that the model suffered before (Fig. 3.7). Specifically, although what affects GAN-VAE's sampling mostly is BN statistics adjustment of the decoder, we find the key to improving OoD likelihood is to adjust that of the encoder. This is because the encoder determines latent feature representations to decoder (we will explore it further in Sec. 3.3.7). We verify this also on the experiment of a weak generative backbone (Fig. 3.7 (b)). After a rough domain adaptation, the model is more likely to reconstruct ID features as reconstructed human faces are clearer on the OoD SVHN digit dataset and keeps performing similarly on ID data. It can also be seen as an enhancement of model capacity and come with little cost, which is accessible to plug on top of any other methods. Notice, it may not be directly extended to the case of flow-based methods because of different mechanisms (Sec. 2.4.4).

3.3.6 Component Analysis

We already use reconstruction errors as rough information guiding the design. Next, we conduct a more detailed analysis. Precisely, the ELBO objective (leaving β) in



(a) (GAN-VAE) Dec train, Enc train, Both train, respectively



(b) (Vanilla VAE) Log-Likelihood, samples and reconstructions of eval and train mode, respectively

Figure 3.7: BN effects to likelihood. **(a)** GAN-VAE experiments of adjusting the encoder and decoder separately and together, on CIFAR-10 vs. SVHN. **(b)** Vanilla VAE experiments on CelebA vs. SVHN. We visualize some samples and corresponding reconstruction. The simple domain Adaptation does not affect ID data much but reconstructs more ID features on OoD data.

VAEs can be factorized into three components.

$$\begin{aligned} ELBO &= \underset{P_\phi(\cdot|\mathbf{X})}{E}\left[\log \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{p_\phi(\mathbf{z}|\mathbf{x})}\right] = \underset{P_\phi(\cdot|\mathbf{X})}{E}[\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}(P_\phi(\cdot|\mathbf{X})\|P_\mathbf{Z}) \\ &= \underset{P_\phi(\cdot|\mathbf{X})}{E}[\log p_\theta(\mathbf{x}|\mathbf{z})] + \underset{P_\phi(\cdot|\mathbf{X})}{E}[\log p(\mathbf{z})] - \underset{P_\phi(\cdot|\mathbf{X})}{E}[\log p_\phi(\mathbf{z}|\mathbf{x})]. \end{aligned} \tag{3.1}$$

Using a classical setting $P_\theta(\mathbf{X}|\mathbf{Z}) = \mathcal{N}(\mathbf{x}|dec(\mathbf{z}), \sigma^2\mathbf{I})$, $P(\mathbf{Z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, $P_\phi(\mathbf{Z}|\mathbf{X}) = \mathcal{N}(\mathbf{z}|\mu(\mathbf{x}), Diag(\sigma^2(\mathbf{x})))$, the three terms can be grounded to,

$$\underset{P_\phi(\cdot|\mathbf{X})}{E}[\log p_\theta(\mathbf{x}|\mathbf{z})] \propto -\frac{1}{\sigma^2}\|\mathbf{x} - dec(\mathbf{z})\|^2, \tag{3.2}$$

$$\underset{P_\phi(\cdot|\mathbf{X})}{E}[\log p(\mathbf{z})] \propto -\|\mathbf{z}\|^2 = -\|\mu(\mathbf{x})\|^2 + O(\epsilon),\ \mathbf{z} \sim P_\phi(\cdot|\mathbf{X}), \tag{3.3}$$

$$-\underset{P_\phi(\cdot|\mathbf{X})}{E}[\log p_\phi(\mathbf{z}|\mathbf{x})] \propto \sum_d \log \sigma^2(\mathbf{x}), \tag{3.4}$$

where we reparameterize $\mathbf{Z} = \mu(\mathbf{x}) + \sigma(\mathbf{x})\epsilon$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and ignore the stochastic impact of ϵ. From that, we can see the smaller the reconstruction is, the closer the code is to the origin, the larger what the approximate posterior expands is, and the larger the compound likelihood is. $\sum_d \log \sigma^2(\mathbf{x})$ describes the number of bits required to compress a std of a approximate posterior $\mathcal{N}(\mathbf{z}|\mu(\mathbf{x}), Diag(\sigma^2(\mathbf{x})))$. The latter two

objectives are saying about a smaller KL divergence. Here, we show the component analysis of two previous experiments (Fig. 3.8). In many experiments, we observe reconstruction really gives insights to the likelihood of generative models which are based on a compression assumption. The difference of radii $\|\mathbf{z}\|^2$ is not salient. Besides, on some datasets, the model can also map OoD data into narrow regions with a small (Lebesgue) measure. We leave the exploration of underlying differences behind the behaviors as future work.

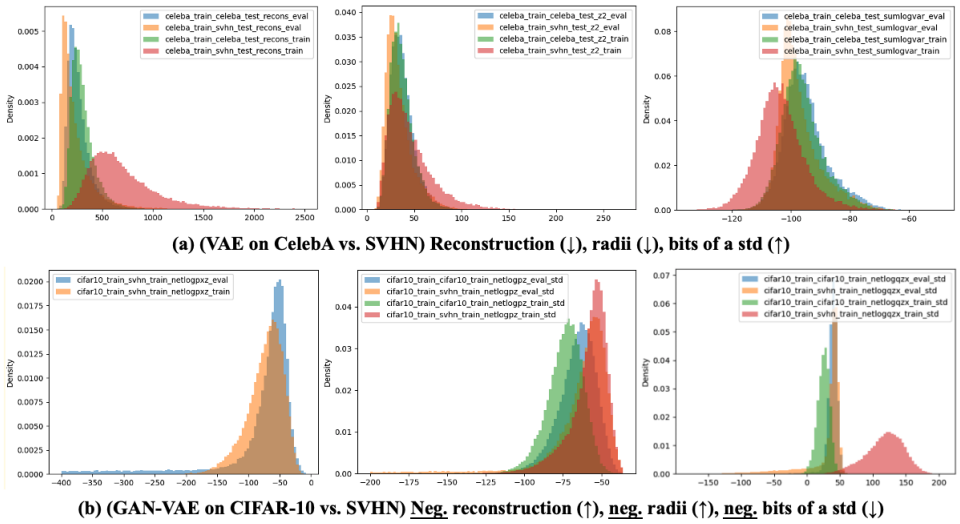


Figure 3.8: Component analysis of two experiments. **(a)** In the BN statistics adjustment of VAEs, reconstruction is the essential one. **(b)** In the GAN-VAE case, the span of approximate posterior also makes big impacts. (The axes are not normalized)

3.3.7 ID/OoD Semantics in Latent Space

Given VAEs suffer the higher OoD likelihood problem, however, the key observation is that OoD reconstruction really looks significantly different from random samples drawn from the model. This is our motivation of using generative models to detect an individual OoD sample, i.e., generative models are trained to only generate ID samples. Based on the component analysis we do in the last section, we propose two hypotheses explaining the higher OoD counter-intuition.

Sharp Likelihood in OoD Regions We think individual or point-wise likelihood in high dimensions is meaningless (see also Sec. 2.4.5). What is meaningful is only population statistics over a region. One case is, we may sample points close to some OoD codes $\mathbf{z}$s to obtain an OoD sample. But we do not really see that. One possibility is, if we think the decoder as a feature descriptor for

codes that is able to distinguish from others, e.g., the human perceptual system, the characteristics change rapidly from OoD to ID when the code moves a bit away from OoD codes. This is depicted by the smoothness of the decoder such as Lipschitz around the region. Another perspective is we are not willing to consider $\log p(\mathbf{x})$ itself but its score, i.e., the gradient w.r.t. $\mathbf{x}$, $\nabla_{\mathbf{x}} \log p(\mathbf{x})$ [16].

Small OoD "Clusters" Characteristics are also reflected on clustering. Here, we mean a broad and general concept of clustering, including linear separable sets. It is impossible to come across a different type of samples when traversing in the latent space. This may mean that (ID) OoD samples cluster and are separated from the other.

In this section, we are studying the second small OoD clusters hypothesis. Inspired by the semantic hyperplane idea in Shen et al. [45], we first decide to test whether there is ID/OoD semantics encoded in latent space in a similar way. Specifically, we try to use a simple hyperplane $y = \mathbf{w}^T\mathbf{z} + b$ to separate ID and OoD latent codes. As in Fig. 3.9 and Tab. 3.1, the linear SVM classifier gets 98.60% and 92.00% accuracy to predict ID and OoD on CelebA vs. SVHN, respectively. According to the principle of SVMs, it means there is an optimal "axis" where ID and OoD latent variables are separated, although two groups seem to mix with each other from the component analysis (Sec. 3.3.6). Fig. 3.9 (c) t-SNE visualization gives a good concept of semantic linear separability. Notice, linear separability announces less in high dimensions if we use a model like one-to-one invertible Flows. For a basic comparison, we test direct separation on random transforms (projections) of raw pixel space, and the performance is not consistent and generalized. This indicates our encoder is a semantic transformation. The experiment results are also pretty in line with our raw observations that ID/OoD semantics includes the differences between ID and OoD samples on image contrast, frequency resolution, and diversity, etc.

To further verify the existence of ID/OoD semantics in latent space, we perform linear interpolation along the direction $\mathbf{w}$ perpendicular to the decision boundary $\mathbf{w}^T\mathbf{z}+b = 0$ in latent space (Fig. 3.10). From the result visualization, we do see the process of generated images showing clearer ID human faces changing from blur. Even in the GAN-VAE experiment, features in generation are consistently enhanced, meaning there is still space for improvement. We do not access to any ID training data in the procedure. The semantic vector $\mathbf{w}$ can be seen as an aggregated representation of the knowledge learned by the model. This encourages us to make efforts mining the knowledge of generative models. Simple linear SVMs can be substituted with other

models fitting different settings, e.g., one-class SVMs for zero-shot OoD detection. Additionally, we visualize some hard samples both of ID CelebA and OoD SVHN in Fig. 3.11. They are indeed confusing. So far, we do not find more interesting findings regarding the correction between distances to the boundary and log-likelihood.

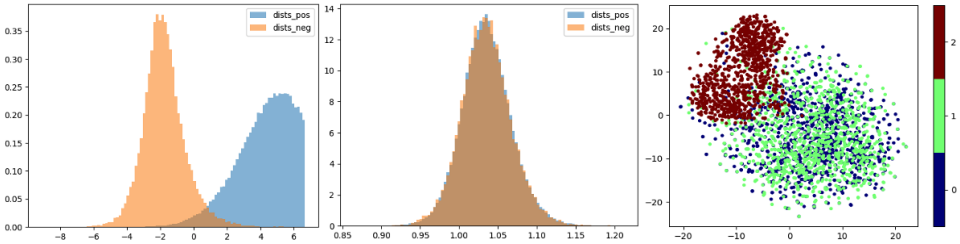


Figure 3.9: Histograms of weighted (signed) distances $d = y = \mathbf{w}^T + b$ and latent code visualization. **(a)** Highly separable in CelebA (ID) vs. SVHN (OoD). **(b)** Large overlap in CelebA train (ID) vs. CelebA test (ID). **(c)** Qualitative t-SNE visualization of $\mathbf{z}$s. 0: CelebA train (ID). 1: CelebA test (ID). 2: SVHN (OoD). The results are similar on other dataset pairs.

Model	Pos.	Neg.	Pos. Acc.	Neg. Acc.
VAE	CelebA	CelebA	1.000	0.000
VAE	CelebA	SVHN	0.986	0.920
VAE (BN train)	CelebA	SVHN	0.920	0.980
GAN-VAE	CelebA	LSUN	0.930	0.960
GAN-VAE	CIFAR-10	SVHN	0.934	0.947

Table 3.1: Classification accuracy of a linear SVM trained to separate different learned latent codes. We set the ID dataset as positive. All accuracy is above 0.900.

3.4 Chapter Summary

We find existing works used to use weak generative models to do OoD detection. This is neither intuitive nor wise. We study the benefits of using a strong generative backbone instead. Strong generative models have more power to map the data close to ID data regions. Thus, given OoD data have significant differences with ID data, it will receive lower likelihood. We also find a preliminary domain adaptation of BN statistics which can be regarded as an enhancement, boosts OoD detection performance. We provide a precise component analysis to verify the improvement further. Apart from

likelihood signals, noticing DGMs do generate only ID data, we decide to mine the knowledge learned by them. We find ID/OoD semantics is really encoded in latent space, which partially supports the small OoD cluster hypothesis.

Figure 3.10: Linear interpolation along the ID/OoD semantics direction $\mathbf{w}$ from the negative class to positive class (always set to ID CelebA train). The step is of length $2|d|$, which means the center column is on the boundary. **(a)** OoD SVHN. **(b)** ID CelebA test. One can really catch the change in so-called ID/OoD semantics. **(c)** OoD LSUN. **(d)** (GAN-VAE) OoD SVHN.

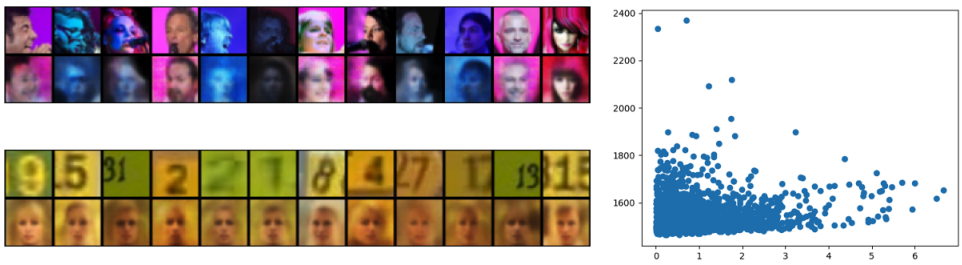


Figure 3.11: **(a)** Visualization of some hard samples near the linear SVM decision boundary. (First row) Hard CelebA and (second row) hard SVHN samples closest to the boundary. **(b)** The plot of distance d to negative log-likelihood $-\log p(\mathbf{x})$. It shapes like a triangle laying on the vertical axis.

Chapter 4

Conclusion and Future Work

4.1 Insights

The insights of the work mainly fall into two folds. The first one is we appeal to using a strong generative backbone to study OoD detection for more intuition and efficiency. The motivation of using deep generative models is that we want to leverage them which are super good at modeling distributions to tell ID and OoD. The premise is DGMs model ID distribution. Under the setting of likelihood-based generative models, the paradigm is as simple as thresholding a trained generative model. Weak generative models have difficulty approximating ID distributions and manifolds. This severe defect requires researchers to come up with many in-training and post-processing patches to calibrate higher likelihood on OoD data. Although some are truths complementing common domain prior and model inductive biases, the whole things make the paradigm complicated and unnatural. Therefore, this motivates us to challenge conventional work to investigate the effects of strong generative models. We believe DGMs based on bottleneck compression assumptions benefit from a stronger ability to map OoD data close to ID regions. It will make ID features reconstructed on OoD data and leave codes in narrow regions. Thus, it naturally alleviates the higher OoD likelihood problem and allows good OoD detection performance.

The other is based on a key observation that DGMs can only generate ID images. This is parallel with likelihood metric-based detection. The knowledge encoded in generative models telling about ID features is over there. The fact is just we are currently not equipped with appropriate methods to mine it. The calibrations of likelihood are not sufficient. Based on the preliminary factor analysis, we propose two hypothetical accounts for the counter-intuition. Thanks to well disentangled and meaningful latent space, we discover obvious ID/OoD semantics learned by the model. It also justifies that by randomly sampling from the prior, some DGMs do not generate OoD images forever.

4.2 Future Work

We plan to further investigate the problem along the directions as follows.

Powerful Generative Models The field of generative models is more active, and there are many emerging DGMs coming into being. As mentioned in existing related work, most of them surround old-fashion ones, e.g., VAEs and Flows. These rather limit the growth of the area, generative-based OoD detection, as generative backbones are a crucial part of the pipeline. We look forward to more interesting findings when applying broader latest generative models such as SBMs and EBMs.

Distance-Based Post-Processing For almost all deep learning methods, including Bayesian ones, there are exceptions in the deployment. This is "hopeless" since input space is huge in high dimensions. We claim it is necessary for us to apply some heuristic post-processing with human knowledge. Distance-based methods are good and natural candidates. In many cases, we may at best trust those predictions of NNs in training data regions for what is learned by the model can be generalized similarly. These methods can complement other methods for more trustworthiness.

Theoretical Explanations and Understanding VAEs and Flows are popular for their good explainability. In general, explainability is traded off with expressive ability. For instance, the introduction of GANs into VAEs actually makes implicit distributions. Besides, the analysis of equilibrium reveals it converges to an entropy-regularized version of the underlying distribution rather than the exact distribution itself. To apply stronger DGMs, a better understanding and theoretical explanations definitely help doing the task.

More Difficult Scenarios Current OoD detection settings are preliminary. Since OoD detection of images in high dimensions is really a tough and opening problem, researchers usually consider simple settings, e.g., different two datasets or different classes of one dataset. These are so-called far and near OoD detection. We hope for more studies on some more difficult real-world settings as the field is developing. For instance, it can be long-tailed imbalanced distributions. What is more, one promising application is to help diagnose whether test distributions have similar biases to the training distribution. Generative-based models which learn more robust features from "reconstruction" may be a better choice over their counterparts.

4.3 Conclusion

In this paper, we are tackling OoD detection in computer vision. The high dimensionality makes it a difficult challenge where many traditional machine learning abnormal detection methods cannot work. Attracted by the great hit made by generative models capable of modeling high-dimensional distributions effectively, researchers started exploring OoD detection using DGMs. For those DGMs modeling a PDF, a practical paradigm is thresholding likelihood prediction as a signal of ID and OoD. Although DGMs have many advantages over other OoD detection methods, it is known that a higher OoD likelihood problem exists.

While most of the existing work focus on calibrations or substitutions of likelihood, we find they ignored the significance of adopting a strong generative backbone. By switching to a stronger one, we find some benefits brought, and the problem is naturally alleviated to some extent without the need of calibrations. We claim it more fits the motivations and intuitions.

We also think of the other case that likelihood does not provide sufficient informative signals, but DGMs can really generate only ID data. Two hypotheses are then proposed to account for the observation. For the clustering hypothesis, we discover ID/OoD semantics encoded in latent space. We think it mines some knowledge of generative models and will help the development of new methods in this direction in the future.

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

[2] Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *ICLR (Poster)*, 2016.

[3] Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in β-vae. *arXiv preprint arXiv:1804.03599*, 2018.

[4] Tong Che, Xiaofeng Liu, Site Li, Yubin Ge, Ruixiang Zhang, Caiming Xiong, and Yoshua Bengio. Deep verifier networks: Verification of deep discriminative models with deep generative models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(8):7002–7010, May 2021.

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

[6] Tal Daniel and Aviv Tamar. Soft-introvae: Analyzing and improving the introspective variational autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4391–4400, 2021.

[7] Erik Daxberger and José Miguel Hernández-Lobato. Bayesian variational autoencoders for unsupervised out-of-distribution detection. *arXiv preprint arXiv:1912.05651*, 2019.

[8] Taylor Denouden, Rick Salay, Krzysztof Czarnecki, Vahdat Abdelzad, Buu Phan, and Sachin Vernekar. Improving reconstruction autoencoder out-of-distribution detection with mahalanobis distance. *arXiv preprint arXiv:1812.02765*, 2018.

[9] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: non-linear independent components estimation. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015.

[10] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[11] Carl Doersch. Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*, 2016.

[12] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[13] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Alex Lamb, Martín Arjovsky, Olivier Mastropietro, and Aaron C. Courville. Adversarially learned inference. *CoRR*, abs/1606.00704, 2016.

[14] Stanislav Fort, Jie Ren, and Balaji Lakshminarayanan. Exploring the limits of out-of-distribution detection. *arXiv preprint arXiv:2106.03004*, 2021.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[16] Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020.

[17] Michael U Gutmann and Aapo Hyvärinen. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(2), 2012.

[18] Danijar Hafner, Dustin Tran, Timothy Lillicrap, Alex Irpan, and James Davidson. Noise contrastive priors for functional uncertainty. In *Uncertainty in Artificial Intelligence*, pages 905–914. PMLR, 2020.

[19] Dan Hendrycks, Mantas Mazeika, and Thomas Dietterich. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019.

[20] Christian Henning, Francesco D'Angelo, and Benjamin F Grewe. Are bayesian neural networks intrinsically good at out-of-distribution detection? *arXiv preprint arXiv:2107.12248*, 2021.

[21] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. 2016.

[22] Zhiting Hu, Zichao Yang, Ruslan Salakhutdinov, and Eric P. Xing. On unifying deep generative models. In *International Conference on Learning Representations*, 2018.

[23] Huaibo Huang, zhihang li, Ran He, Zhenan Sun, and Tieniu Tan. Introvae: Introspective variational autoencoders for photographic image synthesis. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[24] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019.

[25] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun, editors, *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[26] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, 2018.

[27] Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *Advances in neural information processing systems*, 29:4743–4751, 2016.

[28] Polina Kirichenko, Pavel Izmailov, and Andrew G Wilson. Why normalizing flows fail to detect out-of-distribution data. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20578–20589, 2020.

[29] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*, 2021.

[30] Rithesh Kumar, Anirudh Goyal, Aaron C. Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. *CoRR*, abs/1901.08508, 2019.

[31] Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.

[32] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21464–21475. Curran Associates, Inc., 2020.

[33] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[34] Warren Morningstar, Cusuh Ham, Andrew Gallagher, Balaji Lakshminarayanan, Alex Alemi, and Joshua Dillon. Density of states estimation for out of distribution detection. In *International Conference on Artificial Intelligence and Statistics*, pages 3232–3240. PMLR, 2021.

[35] Preetum Nakkiran, Behnam Neyshabur, and Hanie Sedghi. The deep bootstrap framework: Good online learners are good offline generalizers. In *International Conference on Learning Representations*, 2021.

[36] Eric Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Gorur, and Balaji Lakshminarayanan. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019.

[37] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 271–279, 2016.

[38] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30, 2017.

[39] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning*

Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings, 2016.

[40] Tom Rainforth, Adam Kosiorek, Tuan Anh Le, Chris Maddison, Maximilian Igl, Frank Wood, and Yee Whye Teh. Tighter variational bounds are not necessarily better. In *International Conference on Machine Learning*, pages 4277–4285. PMLR, 2018.

[41] Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, 2019.

[42] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29:2234–2242, 2016.

[43] Robin Schirrmeister, Yuxuan Zhou, Tonio Ball, and Dan Zhang. Understanding anomaly detection with deep invertible networks through hierarchies of distributions and features. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21038–21049. Curran Associates, Inc., 2020.

[44] Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F. Núñez, and Jordi Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. In *International Conference on Learning Representations*, 2020.

[45] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[46] Jianlin Su and Guang Wu. f-vaes: Improve vaes with conditional flows. *arXiv preprint arXiv:1809.05861*, 2018.

[47] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. In Yoshua Bengio and Yann LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.

[48] Zhuozhuo Tu, Fengxiang He, and Dacheng Tao. Understanding generalization in recurrent neural networks. In *International Conference on Learning Representations*, 2019.

[49] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19667–19679. Curran Associates, Inc., 2020.

[50] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756. PMLR, 2016.

[51] Sachin Vernekar, Ashish Gaurav, Vahdat Abdelzad, Taylor Denouden, Rick Salay, and Krzysztof Czarnecki. Out-of-distribution detection in classifiers via generation. *arXiv preprint arXiv:1910.04241*, 2019.

[52] Zhisheng Xiao, Karsten Kreis, Jan Kautz, and Arash Vahdat. {VAEBM}: A symbiosis between variational autoencoders and energy-based models. In *International Conference on Learning Representations*, 2021.

[53] Zhisheng Xiao, Qing Yan, and Yali Amit. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20685–20696, 2020.

[54] Sangwoong Yoon, Yung-Kyun Noh, and Frank Park. Autoencoding under normalization constraints. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12087–12097. PMLR, 18–24 Jul 2021.

[55] Lily Zhang, Mark Goldstein, and Rajesh Ranganath. Understanding failures in out-of-distribution detection with deep generative models. In *International Conference on Machine Learning*, pages 12427–12436. PMLR, 2021.