
Enemy is Inside: Alleviating VAE’s Overestimation in Unsupervised OOD Detection

Anonymous Author(s)

Affiliation

Address

email

Abstract

Deep generative models (DGMs) aim at characterizing the distribution of the training set by maximizing the marginal likelihood of inputs in an unsupervised manner, making them a promising option for unsupervised out-of-distribution (OOD) detection. However, recent works have reported that DGMs often assign higher likelihoods to OOD data than in-distribution (ID) data, *i.e.*, **overestimation**, leading to their failures in OOD detection. Although several pioneer works have tried to analyze this phenomenon, and some VAE-based methods have also attempted to alleviate this issue by modifying their score functions for OOD detection, the root cause of the *overestimation* in VAE has never been revealed to our best knowledge. To fill this gap, this paper will provide a thorough theoretical analysis on the *overestimation* issue of VAE, and reveal that this phenomenon arises from two Inside-Enemy aspects: 1) the improper design of prior distribution; 2) the gap of dataset entropies between ID and OOD datasets. Based on these findings, we propose a novel score function to **Alleviate VAE’s Overestimation In** unsupervised OOD Detection, named “**AVOID**”, which contains two novel techniques, specifically post-hoc prior and dataset entropy calibration. Experimental results verify our analysis, demonstrating that the proposed method is effective in alleviating *overestimation* and improving unsupervised OOD detection performance.

1 Introduction

The detection of out-of-distribution (OOD) data, *i.e.*, identifying data that differ from the in-distribution (ID) training set, is crucial for ensuring the reliability and safety of real-world applications [1, 2, 3, 4]. While the most commonly used OOD detection methods rely on supervised classifiers [5, 6, 7, 8, 9, 10, 11], which require labeled data, the focus of this paper is on designing an unsupervised OOD detector. **Unsupervised OOD detection** refers to the task of designing a detector, based solely on the unlabeled training data, that can determine whether an input is ID or OOD [12, 13, 14, 15, 16, 17, 18]. This unsupervised approach is more practical for real-world scenarios where the data lack labels.

Deep generative models (DGMs) are a highly attractive option for unsupervised OOD detection. DGMs, mainly including the auto-regressive model [19, 20], flow model [21, 22], diffusion model [23], generative adversarial network [24], and variational autoencoder (VAE) [25], are designed to model the distribution of the training set by explicitly or implicitly maximizing the likelihood estimation of $p(\mathbf{x})$ for its input \mathbf{x} without category label supervision or additional OOD auxiliary data. They have achieved great successes in a wide range of applications, such as image and text generation. Since generative models are promising at modeling the distribution of the training set, they could be seen as an ideal unsupervised OOD detector, where the likelihood of the unseen OOD data output by the model should be lower than that of the in-distribution data.

Unfortunately, developing a flawless unsupervised OOD detector using DGMs is not as easy as it seems to be. Recent experiments have revealed a counterfactual phenomenon that directly applying the likelihood of generative models as an OOD detector can result in **overestimation**, *i.e.*, **DGMs assign higher likelihoods to OOD data than ID data** [12, 13, 17, 18]. For instance, a generative model trained on the FashionMNIST dataset could assign higher likelihoods to data from the MNIST dataset (OOD) than data from the FashionMNIST dataset (ID), as shown in Figure 6(a). Since OOD detection can be viewed as a verification of whether a generative model has learned to model the distribution of the training set accurately, the counterfactual phenomenon of *overestimation* not only poses challenges to unsupervised OOD detection but also raises doubts about the generative model’s fundamental ability in modeling the data distribution. Therefore, it highlights the need for developing more effective methods for unsupervised OOD detection and, more importantly, a more thorough understanding of the reasons behind the *overestimation* in deep generative models.

To develop more effective methods for unsupervised OOD detection, some approaches have modified the likelihood to new score functions based on empirical assumptions, such as low- and high-level features’ consistency [17, 18] and ensemble approaches [26]. While these methods, particularly the VAE-based methods [18], have achieved state-of-the-art (SOTA) performance in unsupervised OOD detection, none of them provides a clear explanation for the *overestimation* issue. To gain insight into the *overestimation* issue in generative models, pioneering works have shown that the *overestimation* issue could arise from the intrinsic model curvature brought by the invertible architecture in flow models [27]. However, in contrast to the exact marginal likelihood estimation used in flow and auto-regressive models, VAE utilizes a lower bound of the likelihood, making it difficult to analyze. Overall, the reasons behind the *overestimation* issue of VAE are still not fully understood.

In this paper, we try to address the research gap by providing a theoretical analysis of VAE’s *overestimation* in unsupervised OOD detection. Our contributions can be summarized as follows:

1. Through theoretical analyses, we are the first to identify two factors that cause the *overestimation* issue of VAE: 1) the improper design of prior distribution; 2) the intrinsic gap of dataset entropies between ID and OOD datasets;
2. Focused on these two discovered factors, we propose a new score function, named “**AVOID**”, to alleviate the *overestimation* issue from two aspects: i) post-hoc prior for the improper design of prior distribution; ii) dataset entropy calibration for the gap of dataset entropies;
3. Extensive experiments demonstrate that our method can effectively improve the performance of VAE-based methods on unsupervised OOD detection, with theoretical guarantee.

2 Preliminaries

2.1 Unsupervised Out-of-distribution Detection

In this part, we will first give a problem statement of OOD detection and then we will introduce the detailed setup for applying unsupervised OOD detection.

Problem statement. While deploying a machine learning system, it is possible to encounter inputs from unknown distributions that are semantically and/or statistically different from the training data, and such inputs are referred to as OOD data. Processing OOD data could potentially introduce critical errors that compromise the safety of the system [1]. Thus, the OOD detection task is to identify these OOD data, which could be seen as a binary classification task: determining whether an input \mathbf{x} is more likely ID or OOD. It could be formalized as a level-set estimation:

$$\mathbf{x} = \begin{cases} \text{ID}, & \text{if } \mathcal{S}(\mathbf{x}) > \lambda, \\ \text{OOD}, & \text{if } \mathcal{S}(\mathbf{x}) \leq \lambda, \end{cases} \quad (1)$$

where $\mathcal{S}(\mathbf{x})$ denotes the score function, *i.e.*, **OOD detector**, and the threshold λ is commonly chosen to make a high fraction (*e.g.*, 95%) of ID data is correctly classified [9]. In conclusion, OOD detection aims at designing the $\mathcal{S}(\mathbf{x})$ that could assign higher scores to ID data samples than OOD ones.

Setup. Denoting the input space with \mathcal{X} , an *unlabeled* training dataset $\mathcal{D}_{\text{train}} = \{\mathbf{x}_i\}_{i=1}^N$ containing of N data points can be obtained by sampling *i.i.d.* from a data distribution $\mathcal{P}_{\mathcal{X}}$. Typically, we treat the $\mathcal{P}_{\mathcal{X}}$ as p_{id} , which represents the in-distribution (ID) [17, 27]. With this *unlabeled* training set, unsupervised OOD detection is to design a score function $\mathcal{S}(\mathbf{x})$ that can determine whether an input is ID or OOD. This is different from supervised OOD detection, which typically leverages a classifier that is trained on labeled data [4, 7, 9]. We provide a detailed discussion in Appendix A.

2.2 VAE-based Unsupervised OOD Detection

DGMs could be an ideal choice for unsupervised OOD detection because the estimated marginal likelihood $p_\theta(\mathbf{x})$ can be naturally used as the score function $\mathcal{S}(\mathbf{x})$. Among DGMs, VAE can offer great flexibility and strong representation ability [28], leading to a series of unsupervised OOD detection methods based on VAE that have achieved SOTA performance [17, 18]. Specifically, VAE estimates the marginal likelihood by training with the variational evidence lower bound (ELBO), *i.e.*,

$$\text{ELBO}(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})] - D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})), \quad (2)$$

where the posterior $q_\phi(\mathbf{z}|\mathbf{x})$ is modeled by an encoder, the reconstruction likelihood $p_\theta(\mathbf{x}|\mathbf{z})$ is modeled by a decoder, and the prior $p(\mathbf{z})$ is set as a Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$. After well training the VAE, $\text{ELBO}(\mathbf{x})$ is an estimation of the $p(\mathbf{x})$, which could be directly seen as the score function $\mathcal{S}(\mathbf{x})$ to do OOD detection. But the VAE would suffer from the *overestimation* issue, which will be introduced in the next section. More details and **Related Work** can be seen in Appendix B.

3 Analysis of VAE’s *overestimation* in Unsupervised OOD Detection

We will first conduct an analysis to identify the factors contributing to VAE’s *overestimation*, *i.e.*, the improper design of prior distribution and the gap between ID and OOD datasets’ entropies. Subsequently, we will give a deeper analysis of the first factor to have a better understanding.

3.1 Identifying Factors of VAE’s *Overestimation* Issue

Following the common analysis procedure [27], an ideal score function $\mathcal{S}(\mathbf{x})$ that could achieve good OOD detection performance is expected to have the following property for any OOD dataset:

$$\mathcal{G} = \mathbb{E}_{\mathbf{x} \sim p_{\text{id}}(\mathbf{x})} [\mathcal{S}(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p_{\text{ood}}(\mathbf{x})} [\mathcal{S}(\mathbf{x})] > 0, \quad (3)$$

where $p_{\text{id}}(\mathbf{x})$ and $p_{\text{ood}}(\mathbf{x})$ denote the true distribution of the ID and OOD dataset, respectively. A larger gap between these two expectation terms can usually lead to better OOD detection performance.

Using the $\text{ELBO}(\mathbf{x})$ as the score function $\mathcal{S}(\mathbf{x})$, we could give a formal definition of the repeatedly reported VAE’s *overestimation* issue in the context of unsupervised OOD detection [12, 13, 17, 18].

Definition 1 (VAE’s *overestimation* in unsupervised OOD Detection). Assume we have a VAE trained on a training set and we use the $\text{ELBO}(\mathbf{x})$ as the score function to distinguish data points sampled *i.i.d.* from the in-distribution testing set (p_{id}) and an OOD dataset (p_{ood}). When

$$\mathcal{G} = \mathbb{E}_{\mathbf{x} \sim p_{\text{id}}(\mathbf{x})} [\text{ELBO}(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p_{\text{ood}}(\mathbf{x})} [\text{ELBO}(\mathbf{x})] \leq 0, \quad (4)$$

it is called VAE’s *overestimation* in unsupervised OOD detection.

With a clear definition of *overestimation*, we could now investigate the underlying factors causing the *overestimation* in VAE. After well training a VAE, we could reformulate the expectation term of $\text{ELBO}(\mathbf{x})$ from the perspective of information theory [29] as:

$$\begin{aligned} \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [\text{ELBO}(\mathbf{x})] &= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z})] - \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] \\ &= -\mathcal{H}_p(\mathbf{x}) - D_{\text{KL}}(q(\mathbf{z})||p(\mathbf{z})), \end{aligned} \quad (5)$$

because we have

$$\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z})] = \mathcal{I}_q(\mathbf{x}, \mathbf{z}) + \mathbb{E}_{p(\mathbf{x})} \log p(\mathbf{x}) = \mathcal{I}_q(\mathbf{x}, \mathbf{z}) - \mathcal{H}_p(\mathbf{x}), \quad (6)$$

$$\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] = \mathcal{I}_q(\mathbf{x}, \mathbf{z}) + D_{\text{KL}}(q(\mathbf{z})||p(\mathbf{z})), \quad (7)$$

where the $\mathcal{I}_q(\mathbf{x}, \mathbf{z})$ is mutual information between \mathbf{x} and \mathbf{z} and the $q(\mathbf{z})$ is the aggregated posterior distribution of the latent variables \mathbf{z} , which is defined by $q(\mathbf{z}) = \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} q_\phi(\mathbf{z}|\mathbf{x})$. We leave the detailed definition and derivation in Appendix C.1. Thus, the gap \mathcal{G} in Eq. (4) could be rewritten as

$$\mathcal{G} = [-\mathcal{H}_{p_{\text{id}}}(\mathbf{x}) + \mathcal{H}_{p_{\text{ood}}}(\mathbf{x})] + [-D_{\text{KL}}(q_{\text{id}}(\mathbf{z})||p(\mathbf{z})) + D_{\text{KL}}(q_{\text{ood}}(\mathbf{z})||p(\mathbf{z}))], \quad (8)$$

where the dataset entropy $\mathcal{H}_{p_{\text{id}}}(\mathbf{x})/\mathcal{H}_{p_{\text{ood}}}(\mathbf{x})$ is a constant that only depends on the true distribution of ID/OOD dataset; the prior $p(\mathbf{z})$ is typically set as a standard (multivariate) Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$ to enable reparameterization for efficient gradient descent optimization [25].

Through analyzing the most widely used criterion, specifically the expectation of ELBO reformulated in Eq. (8), for VAE-based unsupervised OOD detection, we find that there will be two potential factors that lead to the *overestimation* issue of VAE, *i.e.*, $\mathcal{G} \leq 0$:

127 **Factor I: The improper design of prior distribution $p(z)$.** Several studies have argued that the
 128 aggregated posterior distribution of latent variables $q(z)$ cannot always equal $\mathcal{N}(\mathbf{0}, \mathbf{I})$, particularly
 129 when the dataset exhibits intrinsic multimodality [28, 30, 31, 32]. In fact, when $q(z)$ is extremely
 130 close to $p(z)$, it is more likely to become trapped in a bad local optimum known as posterior collapse
 131 [33, 34, 35], *i.e.*, $q_\phi(z|x) \approx p(z)$, resulting in $q(z) = \int_x q_\phi(z|x)p(x) \approx \int_x p(z)p(x) = p(z)$. In
 132 this situation, the posterior $q_\phi(z|x)$ becomes uninformative about the inputs. Thus, the value of
 133 $D_{\text{KL}}(q_{\text{id}}(z)||p(z))$ could be overestimated, potentially contributing to $\mathcal{G} \leq 0$.

134 **Factor II: The gap between $\mathcal{H}_{p_{\text{id}}}(x)$ and $\mathcal{H}_{p_{\text{ood}}}(x)$.** Considering the dataset’s statistics, such as the
 135 variance of pixel values, different datasets exhibit various levels of entropy. It is reasonable that a
 136 dataset containing images with richer low-level features and more diverse content is expected to have
 137 a higher entropy. As an example, the FashionMNIST dataset should possess higher entropy compared
 138 to the MNIST dataset. Therefore, when the entropy of the ID dataset is higher than that of an OOD
 139 dataset, the value of $-\mathcal{H}_{p_{\text{id}}}(x) + \mathcal{H}_{p_{\text{ood}}}(x)$ is less than 0, potentially leading to *overestimation*.

140 3.2 More Analysis on Factor I

141 In this part, we will focus on addressing the following question: *when is the common design of the*
 142 *prior distribution proper, and when is it not?*

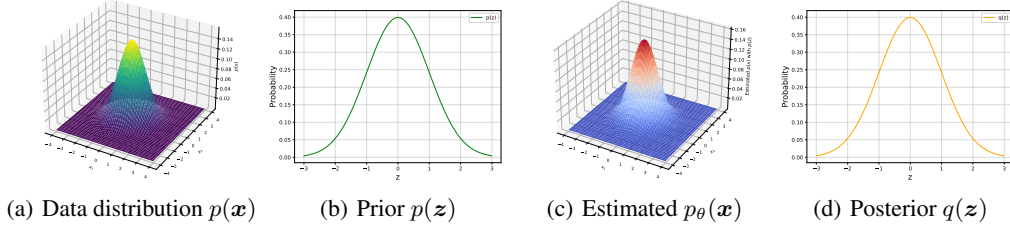


Figure 1: Visualization of modeling a single-modal data distribution with a linear VAE.

143 **When the design of prior is proper?** Assuming that we have a dataset consisting of N data points
 144 $\{x_i\}_{i=1}^N$, each of which is sampled from a given d -dimensional data distribution $p(x) = \mathcal{N}(x|\mathbf{0}, \Sigma_x)$
 145 as shown in Figure 1(a). Then we construct a linear VAE to estimate $p(x)$, formulated as:

$$\begin{aligned} p(z) &= \mathcal{N}(z|\mathbf{0}, \mathbf{I}) \\ q_\phi(z|x) &= \mathcal{N}(z|\mathbf{A}x + \mathbf{B}, \mathbf{C}) \\ p_\theta(x|z) &= \mathcal{N}(x|\mathbf{E}z + \mathbf{F}, \sigma^2\mathbf{I}), \end{aligned} \quad (9)$$

146 where $\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{E}, \mathbf{F}$, and σ are all learnable parameters and their optimal values can be obtained by
 147 the derivation in Appendix C.3. As the estimated distribution $p_\theta(x)$ depicted in Figure 1(c), we can
 148 find that the linear VAE with the optimal parameter values can accurately estimate the $p(x)$ through
 149 maximizing ELBO, *i.e.*, the *overestimation* issue is not present. In this case, Figures 1(b) and 1(d)
 150 indicate that the design of the prior distribution is proper, where the posterior $q(z)$ equals prior $p(z)$.

151 **When the design of prior is NOT proper?** Consider a more complex data distribution, *e.g.*, a mixture
 152 of Gaussians, $p(x) = \sum_{k=1}^K \pi_k \mathcal{N}(x|\mu_k, \Sigma_k)$, $K = 2$ as shown in Figure 2(a), where $\pi_k = 1/K$
 153 and $\sum_{k=1}^K \mu_k = \mathbf{0}$. We construct a dataset consisting of $K \times N$ data points, obtained by sampling
 154 N data samples $\{x_i^{(k)}\}_{i=1, k=1}^{N, K}$ from each component Gaussian $\mathcal{N}(x|\mu_k, \Sigma_k)$. The formulation of
 155 $p(z)$, $q_\phi(z|x)$, and $p_\theta(x|z)$ is consistent with those in Eq. (9). More details are in Appendix C.2.

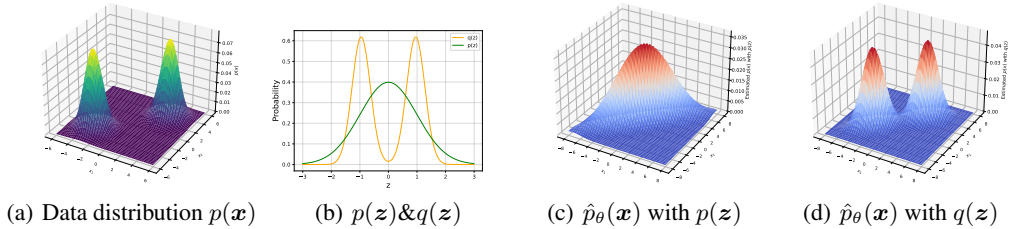


Figure 2: Visualization of modeling a multi-modal data distribution with a linear VAE.

156 In what follows, we will provide a basic derivation outline for the linear VAE under the multi-modal
 157 case. We can first obtain the marginal likelihood $\hat{p}_\theta(x; \mathbf{E}, \mathbf{F}, \sigma) = \int p_\theta(x|z)p(z) = \mathcal{N}(x|\mathbf{F}, \mathbf{E}\mathbf{E}^\top +$

158 $\sigma^2 \mathbf{I}$) with the strictly tighter importance sampling on ELBO [36], *i.e.*, learning the optimal generative
 159 process. Then, the joint log-likelihood of the observed dataset $\{\mathbf{x}_i^{(k)}\}_{i=1, k=1}^{N, K}$ can be formulated as:

$$\mathcal{L} = \sum_{k=1}^K \sum_{i=1}^N \log \hat{p}_\theta(\mathbf{x}_i^{(k)}) = -\frac{KNd}{2} \log(2\pi) - \frac{KN}{2} \log \det(\mathbf{M}) - \frac{KN}{2} \text{tr}[\mathbf{M}^{-1} \mathbf{S}], \quad (10)$$

160 where $\mathbf{M} = \mathbf{E}\mathbf{E}^\top + \sigma^2 \mathbf{I}$ and $\mathbf{S} = \frac{1}{KN} \sum_{k=1}^K \sum_{i=1}^N (\mathbf{x}_i^{(k)} - \mathbf{F})(\mathbf{x}_i^{(k)} - \mathbf{F})^\top$. After that, we could
 161 explore the stationary points of parameters through the ELBO, which can be analytically written as:

$$\begin{aligned} \text{ELBO}(\mathbf{x}) &= \overbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} [\log p_\theta(\mathbf{x}|\mathbf{z})]}^{L_1} - \overbrace{D_{\text{KL}}[q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]}^{L_2}, \\ L_1 &= \frac{1}{2\sigma^2} [-\text{tr}(\mathbf{E}\mathbf{C}\mathbf{E}^\top) - (\mathbf{E}\mathbf{A}\mathbf{x} + \mathbf{E}\mathbf{B})^\top (\mathbf{E}\mathbf{A}\mathbf{x} + \mathbf{E}\mathbf{B}) + 2\mathbf{x}^\top (\mathbf{E}\mathbf{A}\mathbf{x} + \mathbf{E}\mathbf{B}) - \mathbf{x}^\top \mathbf{x}] - \frac{d}{2} \log(2\pi\sigma^2), \\ L_2 &= \frac{1}{2} [-\log \det(\mathbf{C}) + (\mathbf{A}\mathbf{x} + \mathbf{B})^\top (\mathbf{A}\mathbf{x} + \mathbf{B}) + \text{tr}(\mathbf{C}) - 1]. \end{aligned} \quad (11)$$

162 The detailed derivation of parameter solutions in Eq. (10) and (11) can be found in Appendix C.4.

163 In conclusion of this case, Figure 2(b) illustrates that $q(\mathbf{z})$ is a multi-modal distribution instead of
 164 $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$, *i.e.*, the design of the prior is not proper, which leads to *overestimation* as seen in
 165 Figure 2(c). However, as analyzed in Factor I, we found that the *overestimation* issue is mitigated
 166 when replacing $p(\mathbf{z})$ in the KL term of the ELBO with $q(\mathbf{z})$, which is shown in Figure 2(d).

167 **More empirical studies on the improper design of prior.** To extend to a more practical and
 168 representative case, we used a 3-layer MLP to model $q_\phi(\mathbf{z}|\mathbf{x})$ and $p_\theta(\mathbf{x}|\mathbf{z})$ with $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ on
 169 the same dataset of the above multi-modal case. Implementation details are provided in Appendix
 170 C.5. After training, we observed that $q(\mathbf{z})$ still differs from $p(\mathbf{z})$, as shown in Figure 3(a). The ELBO
 171 still suffers from *overestimation*, especially in the region near $(0, 0)$, as shown in Figure 3(b).

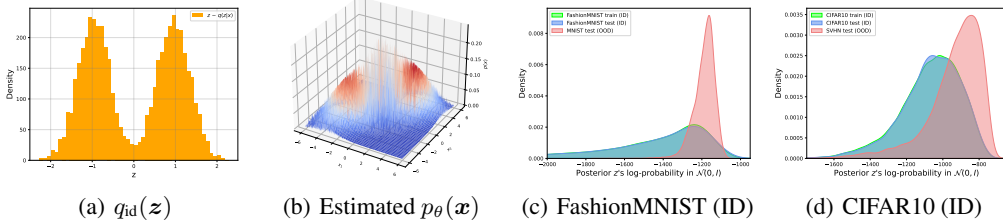


Figure 3: (a) and (b): visualization of $q_{\text{id}}(\mathbf{z})$ and estimated $p(\mathbf{x})$ by ELBO on the multi-modal data distribution with a non-linear deep VAE; (c) and (d): the density plot of the log-probability of posterior \mathbf{z} , *i.e.*, $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$, in prior $\mathcal{N}(\mathbf{0}, \mathbf{I})$ on two dataset pairs.

172 Finally, we extend the analysis directly to high-dimensional image data. Since VAE trained on image
 173 data needs to be equipped with a higher dimensional latent variable space, it is hard to visualize
 174 directly. But please note that, if $q_{\text{id}}(\mathbf{z})$ is closer to $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$, $\mathbf{z}_{\text{id}} \sim q_{\text{id}}(\mathbf{z})$ should occupy
 175 the center of latent space $\mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mathbf{z}_{\text{ood}} \sim q_{\text{ood}}(\mathbf{z})$ should be pushed far from the center, leading to
 176 $p(\mathbf{z}_{\text{id}})$ to be larger than $p(\mathbf{z}_{\text{ood}})$. However, surprisingly, we found this expected phenomenon
 177 does not exist, as shown in Figure 3(c) and 3(d), where the experiments are on two dataset pairs,
 178 Fashion-MNIST(ID)/MNIST(OOD) and CIFAR10(ID)/SVHN(OOD). This still suggests that the
 179 prior $p(\mathbf{z})$ is improper, even $q_{\text{ood}}(\mathbf{z})$ for OOD data may be closer to $p(\mathbf{z})$ than $q_{\text{id}}(\mathbf{z})$.

180 **Brief summary.** Through analyzing *overestimation* scenarios from simple to complex, the answer
 181 to the question at the beginning of this part could be: *the prior distribution $p(\mathbf{z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ is an*
 182 *improper choice for VAE when modeling a complex data distribution $p(\mathbf{x})$, leading to an overestimated*
 183 *$D_{\text{KL}}(q_{\text{id}}(\mathbf{z})||p(\mathbf{z}))$ and further raising the *overestimation* issue in unsupervised OOD detection.*

184 4 Alleviating VAE's *overestimation* in Unsupervised OOD Detection

185 In this section, we develop the “**AVOID**” method to alleviate the influence of two aforementioned
 186 factors in Section 3, including **i)** post-hoc prior and **ii)** dataset entropy calibration, both of which are
 187 implemented in a simple way to inspire related work and can be further investigated for improvement.

188 4.1 Post-hoc Prior Method for Factor I

To provide a more insightful view to investigate the relationship between $q_{\text{id}}(\mathbf{z})$, $q_{\text{ood}}(\mathbf{z})$, and $p(\mathbf{z})$, we use t-SNE [37] to visualize them in Figure 4. The visualization reveals that $p(\mathbf{z})$ cannot distinguish between the latent variables sampled from $q_{\text{id}}(\mathbf{z})$ and $q_{\text{ood}}(\mathbf{z})$, while $q_{\text{id}}(\mathbf{z})$ is clearly distinguishable from $q_{\text{ood}}(\mathbf{z})$. Therefore, to alleviate *overestimation*, we can explicitly modify the prior distribution $p(\mathbf{z})$ in Eq. (8) to force it to be closer to $q_{\text{id}}(\mathbf{z})$ and far from $q_{\text{ood}}(\mathbf{z})$, *i.e.*, decreasing $D_{\text{KL}}(q_{\text{id}}(\mathbf{z})||p(\mathbf{z}))$ and increasing $D_{\text{KL}}(q_{\text{ood}}(\mathbf{z})||p(\mathbf{z}))$.

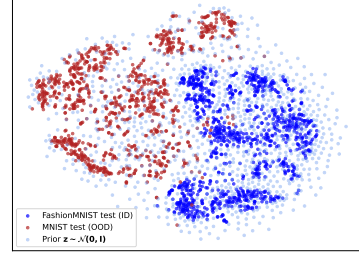


Figure 4: The t-SNE visualization of the latent representations on FashionMNIST(ID)/MNIST(OOD) dataset pair.

A straightforward modifying approach is to replace $p(\mathbf{z})$ in ELBO with an additional distribution $\hat{q}_{\text{id}}(\mathbf{z})$ that can fit $q_{\text{id}}(\mathbf{z})$ well after training, where the target value of $q_{\text{id}}(\mathbf{z})$ can be acquired by marginalizing $q_{\phi}(\mathbf{z}|\mathbf{x})$ over the training set, *i.e.*, $q_{\text{id}}(\mathbf{z}) = \mathbb{E}_{\mathbf{x} \sim p_{\text{id}}(\mathbf{x})}[q_{\phi}(\mathbf{z}|\mathbf{x})]$. Previous study on distribution matching [30] has developed an LSTM-based method to efficiently fit $q_{\text{id}}(\mathbf{z})$ in the latent space, *i.e.*,

$$\hat{q}_{\text{id}}(\mathbf{z}) = \prod_{t=1}^T q(\mathbf{z}_t | \mathbf{z}_{<t}), \text{ where } q(\mathbf{z}_t | \mathbf{z}_{<t}) = \mathcal{N}(\mu_i, \sigma_i^2). \quad (12)$$

Thus, we could propose a “post-hoc prior” (PHP) method for Factor I, formulated as

$$\text{PHP}(\mathbf{x}) := \mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} \log p_{\theta}(\mathbf{x}|\mathbf{z}) - D_{\text{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x})||\hat{q}_{\text{id}}(\mathbf{z})), \quad (13)$$

which could lead to better OOD detection performance since it could enlarge the gap \mathcal{G} , *i.e.*,

$$\mathcal{G}_{\text{PHP}} = [-\mathcal{H}_{p_{\text{id}}}(\mathbf{x}) + \mathcal{H}_{p_{\text{ood}}}(\mathbf{x})] + [-D_{\text{KL}}(q_{\text{id}}(\mathbf{z})||\hat{q}_{\text{id}}(\mathbf{z})) + D_{\text{KL}}(q_{\text{ood}}(\mathbf{z})||\hat{q}_{\text{id}}(\mathbf{z}))] > \mathcal{G}. \quad (14)$$

Please note that PHP can be directly integrated into a trained VAE in a “plug-and-play” manner.

4.2 Dataset Entropy Calibration Method for Factor II

While the entropy of a dataset is a constant that remains unaffected by different model settings, it is still an essential factor that leads to *overestimation*. To address this, a straightforward approach is to design a calibration method that ensures the value added to the ELBO of ID data will be larger than that of OOD data. Specifically, we denote the calibration term as $\mathcal{C}(\mathbf{x})$, and its expected property could be formulated as

$$\mathbb{E}_{\mathbf{x} \sim p_{\text{id}}(\mathbf{x})}[\mathcal{C}(\mathbf{x})] > \mathbb{E}_{\mathbf{x} \sim p_{\text{ood}}(\mathbf{x})}[\mathcal{C}(\mathbf{x})]. \quad (15)$$

After adding the calibration $\mathcal{C}(\mathbf{x})$ to the ELBO(\mathbf{x}), we could obtain the “dataset entropy calibration” (DEC) method for Factor II, formulated as

$$\text{DEC}(\mathbf{x}) := \mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} \log p_{\theta}(\mathbf{x}|\mathbf{z}) - D_{\text{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x})||p(\mathbf{z})) + \mathcal{C}(\mathbf{x}). \quad (16)$$

With the property in Eq. (15), we could find that the new gap \mathcal{G}_{DEC} becomes larger than the original gap \mathcal{G} based solely on ELBO, as $\mathcal{G}_{\text{DEC}} = \mathcal{G} + \mathbb{E}_{\mathbf{x} \sim p_{\text{id}}(\mathbf{x})}[\mathcal{C}(\mathbf{x})] - \mathbb{E}_{\mathbf{x} \sim p_{\text{ood}}(\mathbf{x})}[\mathcal{C}(\mathbf{x})] > \mathcal{G}$, which should alleviate the *overestimation* and lead to better unsupervised OOD detection performance.

How to design the calibration $\mathcal{C}(\mathbf{x})$? For the choice of the function $\mathcal{C}(\mathbf{x})$, inspired by the previous work [13], we could use image compression methods like Singular Value Decomposition (SVD) [38] to roughly measure the complexity of an image, where the images from the same dataset should have similar complexity. An intuitive insight into this could be shown in Figure 5, where the ID dataset’s statistical feature, *i.e.*, the curve, is distinguishable to other datasets. Based on this empirical study, we could first propose a **non-scaled** calibration function, denoted as $\mathcal{C}_{\text{non}}(\mathbf{x})$. First, we could set the number of singular values as n_{id} , which can achieve the reconstruction error $\|\mathbf{x}_{\text{recon}} - \mathbf{x}\| = \epsilon$ in the ID training set; then for a test input \mathbf{x}_i , we use SVD to calculate the smallest n_i that could also achieve a smaller reconstruction error ϵ , then $\mathcal{C}_{\text{non}}(\mathbf{x})$ could be formulated as:

$$\mathcal{C}_{\text{non}}(\mathbf{x}) = \begin{cases} (n_i/n_{\text{id}}), & \text{if } n_i < n_{\text{id}}, \\ (n_{\text{id}} - (n_i - n_{\text{id}}))/n_{\text{id}}, & \text{if } n_i \geq n_{\text{id}}, \end{cases} \quad (17)$$

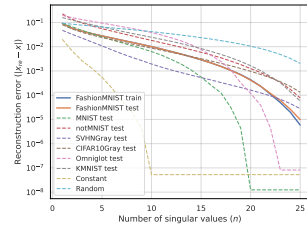


Figure 5: Visualization of the relationship between the number of singular values and the reconstruction error.

which can give the ID dataset a higher expectation $\mathbb{E}_{\mathbf{x} \sim p_{\text{id}}(\mathbf{x})}[\mathcal{C}_{\text{non}}(\mathbf{x})]$ than that of other statistically different OOD datasets. More details to obtain $\mathcal{C}_{\text{non}}(\mathbf{x})$ can be found in Appendix D.

4.3 Putting Them Together to Get “AVOID”

By combining the post-hoc prior (PHP) method and the dataset entropy calibration (DEC) method, we could develop a new score function, denoted as $\mathcal{S}_{\text{AVOID}}(\mathbf{x})$:

$$\mathcal{S}_{\text{AVOID}}(\mathbf{x}) := \mathbb{E}_{q_{\phi}(\mathbf{z}|\mathbf{x})} [\log p_{\theta}(\mathbf{x}|\mathbf{z})] - D_{\text{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x})||\hat{q}_{\text{id}}(\mathbf{z})) + \mathcal{C}(\mathbf{x}). \quad (18)$$

To balance the importance of PHP and DEC terms in Eq. (18), we consider to set an appropriate scale for $\mathcal{C}(\mathbf{x})$. For the scale of $\mathcal{C}(\mathbf{x})$, if it is too small, its effectiveness in alleviating *overestimation* could be limited. Otherwise, it may hurt the effectiveness of the PHP method since DEC will dominate the value of “AVOID”. Additionally, for statistically similar datasets, *i.e.*, $\mathcal{H}_{p_{\text{id}}}(\mathbf{x}) \approx \mathcal{H}_{p_{\text{ood}}}(\mathbf{x})$, the property in Eq. (15) cannot be guaranteed and we may only have $\mathbb{E}_{\mathbf{x} \sim p_{\text{id}}(\mathbf{x})}[\mathcal{C}_{\text{non}}(\mathbf{x})] \approx \mathbb{E}_{\mathbf{x} \sim p_{\text{ood}}(\mathbf{x})}[\mathcal{C}_{\text{non}}(\mathbf{x})]$, in which case we could only rely on the PHP method. Thus, an appropriate scale of $\mathbb{E}_{\mathbf{x} \sim p_{\text{id}}(\mathbf{x})}[\mathcal{C}(\mathbf{x})]$, named “ $\mathcal{C}_{\text{scale}}$ ”, could be derived by $\mathcal{C}_{\text{scale}} = \mathbb{E}_{\mathbf{x} \sim p_{\text{id}}(\mathbf{x})}[\text{PHP}(\mathbf{x})] \approx \mathcal{H}_{p_{\text{id}}}(\mathbf{x})$, which leads to

$$\mathbb{E}_{\mathbf{x} \sim p_{\text{id}}(\mathbf{x})}[\text{DEC}(\mathbf{x})] = -\mathcal{H}_{p_{\text{id}}}(\mathbf{x}) - D_{\text{KL}}(q_{\text{id}}(\mathbf{z})||p(\mathbf{z})) + \mathcal{C}_{\text{scale}} \approx -D_{\text{KL}}(q_{\text{id}}(\mathbf{z})||p(\mathbf{z})). \quad (19)$$

Thus, when $\mathcal{H}_{p_{\text{id}}}(\mathbf{x}) \approx \mathcal{H}_{p_{\text{ood}}}(\mathbf{x})$ and $\mathbb{E}_{\mathbf{x} \sim p_{\text{id}}(\mathbf{x})}[\mathcal{C}(\mathbf{x})] \approx \mathbb{E}_{\mathbf{x} \sim p_{\text{ood}}(\mathbf{x})}[\mathcal{C}(\mathbf{x})]$, the PHP part of “AVOID” could still be helpful to alleviate *overestimation*.

Motivated by the above analysis, we could implement the **scaled** calibration function, formulated as

$$\mathcal{C}(\mathbf{x}) = \mathcal{C}_{\text{non}}(\mathbf{x}) \times \mathcal{C}_{\text{scale}} = \begin{cases} (n_i/n_{\text{id}}) \times \mathcal{C}_{\text{scale}}, & \text{if } n_i < n_{\text{id}}, \\ [(n_{\text{id}} - (n_i - n_{\text{id}}))/n_{\text{id}}] \times \mathcal{C}_{\text{scale}}, & \text{if } n_i \geq n_{\text{id}}. \end{cases} \quad (20)$$

5 Experiments

5.1 Experimental Setup

Datasets. In accordance with existing literature [17, 18, 39], we evaluate our method against previous works using two standard dataset pairs: FashionMNIST [40] (ID) / MNIST [41] (OOD) and CIFAR10 [42] (ID) / SVHN [43] (OOD). The suffixes “ID” and “OOD” represent in-distribution and out-of-distribution datasets, respectively. To more comprehensively assess the generalization capabilities of these methods, we incorporate additional OOD datasets, the details of which are available in Appendix E.1. Notably, datasets featuring the suffix “-G” (e.g., “CIFAR10-G”) have been converted to grayscale, resulting in a single-channel format.

Evaluation and Metrics. We adhere to the previous evaluation procedure [17, 18], where all methods are trained using the training split of the in-distribution dataset, and their OOD detection performance is assessed on both the testing split of the in-distribution dataset and the OOD dataset. In line with previous works [1, 5, 44], we employ evaluation metrics including the area under the receiver operating characteristic curve (AUROC \uparrow), the area under the precision-recall curve (AUPRC \uparrow), and the false positive rate at 80% true positive rate (FPR80 \downarrow). The arrows indicate the direction of improvement for each metric.

Baselines. Our experiments primarily encompass two comparison aspects: **i)** evaluating our novel score function “AVOID” against previous unsupervised OOD detection methods to determine whether it can achieve competitive performance; and **ii)** comparing “AVOID” with VAE’s ELBO to assess whether our method can mitigate *overestimation* and yield improved performance. For comparisons in **i)**, we can categorize the baselines into three groups, as outlined in [18]: “**Supervised**” includes supervised OOD detection methods that utilize in-distribution data labels [1, 5, 9, 45, 46, 47, 48, 49]; “**Auxiliary**” refers to methods that employ auxiliary knowledge gathered from OOD data [13, 39, 44]; and “**Unsupervised**” encompasses methods without reliance on labels or OOD-specific assumptions [14, 17, 18, 26]. For comparisons in **ii)**, we compare our method with a standard VAE [25], which also serves as the foundation of our method. Further details regarding these baselines and their respective categories can be found in Appendix E.2.

Implementation Details. The VAE’s latent variable \mathbf{z} ’s dimension is set as 200 for all experiments with the encoder and decoder parameterized by a 3-layer convolutional neural network, respectively.

Table 1: The comparisons of our method and other OOD detection methods. The best results achieved by the methods of the category “Not ensembles” of “Unsupervised” have been bold.

FashionMNIST(ID)/MNIST(OOD)					CIFAR10(ID)/SVHN(OOD)				
Supervised		Auxiliary		Unsupervised	Supervised		Auxiliary		Unsupervised
Method	AUROC \uparrow	Method	AUROC \uparrow	Method	AUROC \uparrow	Method	AUROC \uparrow	Method	AUROC \uparrow
CP [1]	73.4	LR(PC) [39]	99.4	<i>-Ensembles</i>	MD [46]	99.7	LR(PC) [39]	93.0	<i>-Ensembles</i>
CP(Ent) [1]	74.6	LR(BC) [39]	45.5	WAIC(SVAE) [26]	76.6	LMD [47]	27.9	LR(VAE) [39]	26.5
ODIN [45]	75.2	CP(OOD) [39]	87.7	WAIC(5PC) [26]	22.1	EN [6]	98.9	OE [44]	98.4
VIB [5]	94.1	CP(Cal) [39]	90.4	<i>-Not Ensembles</i>	iDE [52]	95.7	IC(Glow) [13]	95.0	<i>-Not Ensembles</i>
MD(CNN) [46]	94.2	IC(Glow) [13]	99.8	LRe [14]	98.8	LN[9]	98.4	IC(PC++) [13]	92.9
MD(DN) [46]	98.6	IC(PC++) [13]	96.7	HVK [17]	98.4	ODIN [45]	82.9	IC(HVAE) [13]	83.3
DE [1]	85.7			\mathcal{LLR}^{ada} [18]	98.0	GN [49]	76.7		
				AVOID(ours)	99.2			\mathcal{LLR}^{ada} [18]	94.2
								AVOID(ours)	94.5

Table 2: The comparisons of our method with post-hoc prior (denoted as “PHP”) or dataset entropy calibration (denoted as “DEC”) individually and other unsupervised OOD detection methods. “PHP+DEC” is equal to our method “AVOID”. Bold numbers are superior results.

FashionMNIST(ID)/MNIST(OOD)				CIFAR10(ID)/SVHN(OOD)			
Method	AUROC \uparrow	AUPRC \uparrow	FPR80 \downarrow	Method	AUROC \uparrow	AUPRC \uparrow	FPR80 \downarrow
ELBO [25]	23.5	35.6	98.5	ELBO [25]	24.9	36.7	94.6
WAIC(5PC) [26]	22.1	40.1	91.1	WAIC(5PC) [26]	62.8	61.6	65.7
HVK [17]	98.4	98.4	1.3	HVK [17]	89.1	87.5	17.2
\mathcal{LLR}^{ada} [18]	97.0	97.6	0.9	\mathcal{LLR}^{ada} [18]	92.6	91.8	11.1
<i>-Ours:</i>				<i>-Ours:</i>			
PHP	89.7	90.3	13.3	PHP	39.6	42.6	85.7
DEC	34.1	40.7	92.5	DEC	87.8	89.9	17.8
PHP+DEC	99.2	99.4	0.00	PHP+DEC	94.5	95.3	4.24

The reconstruction likelihood distribution is modeled by a discretized mixture of logistics [20]. For optimization, we adopt the same Adam optimizer [50] with a learning rate of 1e-3. We train all models in comparison by setting the batch size as 128 and the max epoch as 1000. All experiments are performed on a PC with an NVIDIA A100 GPU and our code is implemented with PyTorch [51]. More implementation details can be found in Appendix E.3.

5.2 Comparison with Unsupervised OOD Detection Baselines

First, we compare our method with other SOTA baselines in Table 1. The results demonstrate that our method achieves competitive performance compared to “Supervised” and “Auxiliary” methods and outperforms “Unsupervised” OOD detection methods. Next, we provide a more detailed comparison with some unsupervised methods, particularly the ELBO of VAE, as shown in Table 2. These results indicate that our method effectively mitigates *overestimation* and enhances OOD detection performance when using VAE as the backbone. Lastly, to assess our method’s generalization capabilities, we test it on a broader range of datasets, as displayed in Table 3. Experimental results strongly verify our analysis of the VAE’s *overestimation* issue and demonstrate that our method consistently mitigates *overestimation*, regardless of the type of OOD datasets.

5.3 Ablation Study on Verifying the Post-hoc Prior Method

To evaluate the effectiveness of the Post-hoc Prior (PHP), we compare it with other unsupervised methods in Table 2. Moreover, we test the PHP method on additional datasets and present the results in Table 4 of Appendix F. The experimental results demonstrate that the PHP method can alleviate the *overestimation*. To provide a better understanding, we also visualize the density plot of ELBO and PHP for the “FashionMNIST(ID)/MNIST(OOD)” dataset pair in Figures 6(a) and 6(b), respectively.

The Log-likelihood Ratio (\mathcal{LLR}) methods [17, 18] are the current SOTA unsupervised OOD detection methods that also focus on latent variables. These methods are based on an empirical assumption that the bottom layer latent variables of a hierarchical VAE could learn low-level features and top layers learn semantic features. However, we discovered that while ELBO could already perform well in detecting some OOD data, the \mathcal{LLR} method [18] could negatively impact OOD detection performance to some extent, as demonstrated in Figure 6(c), where the model is trained on MNIST and detects FashionMNIST as OOD. On the other hand, our method can still maintain comparable performance since the PHP method can explicitly alleviate *overestimation*, which is one of the strengths of our method compared to the SOTA methods.

5.4 Ablation Study on Verifying the Dataset Entropy Calibration Method

We evaluate the performance of dataset entropy calibration, referred to as “DEC”, in Table 2 and Table 5 of Appendix G. Although the DEC method is simple, our results show that it effectively alleviates *overestimation*. To better understand DEC, we visualize the calculated $\mathcal{C}(x)$ of CIFAR10

Table 3: The comparisons of our method “AVOID” and baseline “ELBO” on more datasets. Bold numbers are superior performance.

ID	FashionMNIST			ID	CIFAR10		
OOD	AUROC \uparrow	AUPRC \uparrow	FPR80 \downarrow	OOD	AUROC \uparrow	AUPRC \uparrow	FPR80 \downarrow
	ELBO / AVOID (ours)				ELBO / AVOID (ours)		
KMNIST	60.03 / 78.71	54.60 / 68.91	61.6 / 48.4	CIFAR100	52.91 / 55.36	51.15 / 72.13	77.42 / 73.93
Omniglot	99.86 / 100.0	99.89 / 100.0	0.00 / 0.00	CelebA	57.27 / 71.23	54.51 / 72.13	69.03 / 54.45
notMNIST	94.12 / 97.72	94.09 / 97.70	8.29 / 2.20	Places365	57.24 / 68.37	56.96 / 69.05	73.13 / 62.64
CIFAR10-G	98.01 / 99.01	98.24 / 99.04	1.20 / 0.40	LFWPeople	64.15 / 67.72	59.71 / 68.81	59.44 / 54.45
CIFAR100-G	98.49 / 98.59	97.49 / 97.87	1.00 / 1.00	SUN	53.14 / 63.09	54.48 / 63.32	79.52 / 68.63
SVHN-G	95.61 / 96.20	96.20 / 97.41	3.00 / 0.40	STL10	49.37 / 64.51	47.79 / 65.50	78.02 / 67.23
CelebA-G	97.33 / 97.87	94.71 / 95.82	3.00 / 0.40	Flowers102	67.68 / 76.83	64.68 / 78.01	57.94 / 46.65
SUN-G	99.16 / 99.32	99.39 / 99.47	0.00 / 0.00	GTSRB	39.50 / 53.06	41.73 / 49.84	86.61 / 73.63
Places365-G	98.92 / 98.89	98.05 / 98.61	0.80 / 0.80	DTD	37.86 / 81.82	40.93 / 62.42	82.22 / 64.24
Const	94.94 / 95.20	97.27 / 97.32	1.80 / 1.70	Const	0.001 / 80.12	30.71 / 89.42	100.0 / 22.38
Random	99.80 / 100.0	99.90 / 100.0	0.00 / 0.00	Random	71.81 / 99.31	82.89 / 99.59	85.71 / 0.000

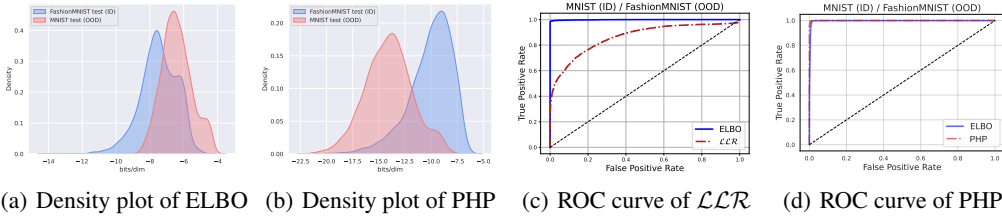


Figure 6: Density plots and ROC curves. **(a):** directly using $\text{ELBO}(\mathbf{x})$, an estimation of the $p(\mathbf{x})$, of a VAE trained on FashionMNIST leads to *overestimation* in detecting MNIST as OOD data; **(b):** using PHP method could alleviate the *overestimation*; **(c):** SOTA method \mathcal{LLR} hurts the performance when ELBO could already work well; **(d):** PHP method would not hurt the performance.

(ID) in Figure 7(a) and other OOD datasets in Figure 7(b) when $n_{id} = 20$. Our results show that the $\mathcal{C}(\mathbf{x})$ of CIFAR10 (ID) achieves generally higher values than that of other datasets, which is the underlying reason for its effectiveness in alleviating *overestimation*. Additionally, we investigate the impact of different n_{id} on OOD detection performance in Figure 7(c), where our results show that the performance is consistently better than ELBO.

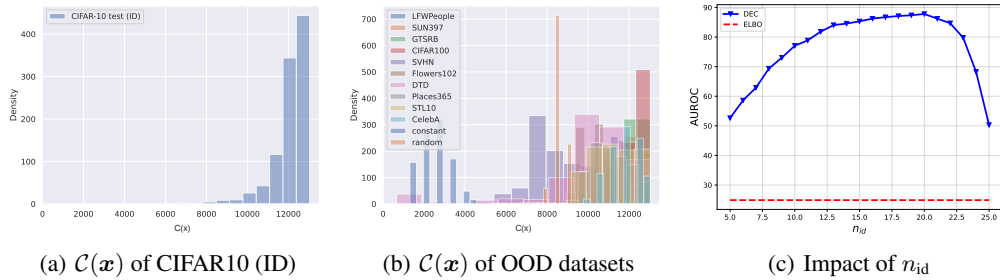


Figure 7: **(a)** and **(b)** are respectively the visualizations of the calculated entropy calibration $\mathcal{C}(\mathbf{x})$ of CIFAR10 (ID) and other OOD datasets, where the $\mathcal{C}(\mathbf{x})$ of CIFAR10 (ID) could achieve generally higher values. **(c)** is the OOD detection performance of dataset entropy calibration with different n_{id} settings, which consistently outperforms ELBO.

6 Conclusion

In conclusion, we have identified the underlying factors that lead to VAE’s *overestimation* in unsupervised OOD detection: the improper design of the prior and the gap of the dataset entropies between the ID and OOD datasets. With this analysis, we have developed a novel score function called “AVOID”, which is effective in alleviating *overestimation* and improving unsupervised OOD detection. This work may lead a research stream for improving unsupervised OOD detection by developing more efficient and sophisticated methods aimed at optimizing these revealed factors.

References

- [1] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. In *ICLR*, 2017.
- [2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.
- [3] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015.
- [4] Hongxin Wei, Lue Tao, Renchunzi Xie, Lei Feng, and Bo An. Open-sampling: Exploring out-of-distribution data for re-balancing long-tailed datasets. In *ICML*, 2022.
- [5] Alexander A. Alemi, Ian Fischer, and Joshua V. Dillon. Uncertainty in the variational information bottleneck. *CoRR*, abs/1807.00906, 2018.
- [6] Weitang Liu, Xiaoyun Wang, John Owens, and Yixuan Li. Energy-based out-of-distribution detection. In *NeurIPS*, 2020.
- [7] Hongxin Wei, Lue Tao, Renchunzi Xie, and Bo An. Open-set label noise can improve robustness against inherent label noise. In *NeurIPS*, 2022.
- [8] Zhuo Huang, Xiaobo Xia, Li Shen, Bo Han, Mingming Gong, Chen Gong, and Tongliang Liu. Harnessing out-of-distribution examples via augmenting content and style. *arXiv preprint arXiv:2207.03162*, 2022.
- [9] Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. In *ICML*, 2022.
- [10] Shuyang Yu, Junyuan Hong, Haotao Wang, Zhangyang Wang, and Jiayu Zhou. Turning the curse of heterogeneity in federated learning into a blessing for out-of-distribution detection. In *ICLR*, 2023.
- [11] Ido Galil, Mohammed Dabbah, and Ran El-Yaniv. A framework for benchmarking class-out-of-distribution detection and its application to imagenet. In *ICLR*, 2023.
- [12] Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *NeurIPS*, 2019.
- [13] 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 *ICLR*, 2020.
- [14] Zhisheng Xiao, Qing Yan, and Yali Amit. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. In *NeurIPS*, 2020.
- [15] Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. BIVA: A very deep hierarchy of latent variables for generative modeling. In *NeurIPS*, 2019.
- [16] Griffin Floto, Stefan Kremer, and Mihai Nica. The tilted variational autoencoder: Improving out-of-distribution detection. In *ICLR*, 2023.
- [17] Jakob D Drachmann Havtorn, Jes Frellsen, Søren Hauberg, and Lars Maaløe. Hierarchical vae’s know what they don’t know. In *ICML*, 2021.
- [18] Yewen Li, Chaojie Wang, Xiaobo Xia, Tongliang Liu, and Bo An. Out-of-distribution detection with an adaptive likelihood ratio on informative hierarchical vae. In *NeurIPS*, 2022.
- [19] Aäron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Koray Kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional image generation with pixelcnn decoders. In *NeurIPS*, 2016.
- [20] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P. Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. In *ICLR*, 2017.

- [21] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *ICLR*, 2017.
- [22] Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *NeurIPS*, 2018.
- [23] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020.
- [24] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.
- [25] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.
- [26] Hyunsun Choi, Eric Jang, and Alexander A Alemi. Waic, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018.
- [27] Eric T. Nalisnick, Akihiro Matsukawa, Yee Whye Teh, Dilan Görür, and Balaji Lakshminarayanan. Do deep generative models know what they don’t know? In *ICLR*, 2019.
- [28] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. In *ICLR*, 2022.
- [29] Thomas M Cover. *Elements of Information Theory*. John Wiley & Sons, 1999.
- [30] Mihaela Rosca, Balaji Lakshminarayanan, and Shakir Mohamed. Distribution matching in variational inference. *CoRR*, abs/1802.06847, 2018.
- [31] Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, 2015.
- [32] William Feller. On the theory of stochastic processes, with particular reference to applications. In *Selected Papers I*, pages 769–798. Springer, 2015.
- [33] Yixin Wang, David M. Blei, and John P. Cunningham. Posterior collapse and latent variable non-identifiability. In *NeurIPS*, 2021.
- [34] Adji B. Dieng, Yoon Kim, Alexander M. Rush, and David M. Blei. Avoiding latent variable collapse with generative skip models. In *AISTATS*, 2019.
- [35] Yewen Li, Chaojie Wang, Zhibin Duan, Dongsheng Wang, Bo Chen, Bo An, and Mingyuan Zhou. Alleviating “posterior collapse” in deep topic models via policy gradient. In *NeurIPS*, 2022.
- [36] Yuri Burda, Roger B. Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *ICLR*, 2016.
- [37] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11), 2008.
- [38] Gilbert W Stewart. On the early history of the singular value decomposition. *SIAM Review*, 35(4):551–566, 1993.
- [39] Jie Ren, Peter J. Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark A. DePristo, Joshua V. Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. In *NeurIPS*, 2019.
- [40] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.
- [41] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [42] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Master’s thesis, University of Tront*, 2009.

- [43] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.
- [44] Dan Hendrycks, Mantas Mazeika, and Thomas G. Dietterich. Deep anomaly detection with outlier exposure. In *ICLR*, 2019.
- [45] Shiyu Liang, Yixuan Li, and R Srikant. Enhancing the reliability of out-of-distribution image detection in neural networks. In *ICLR*, 2018.
- [46] Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018.
- [47] Saikiran Bulusu, Bhavya Kailkhura, Bo Li, Pramod K Varshney, and Dawn Song. Anomalous example detection in deep learning: A survey. *IEEE Access*, 8:132330–132347, 2020.
- [48] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *NeurIPS*, 2017.
- [49] Rui Huang, Andrew Geng, and Yixuan Li. On the importance of gradients for detecting distributional shifts in the wild. In *NeurIPS*, 2021.
- [50] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.
- [51] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Z. Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.
- [52] Ramneet Kaur, Susmit Jha, Anirban Roy, Sangdon Park, Edgar Dobriban, Oleg Sokolsky, and Insup Lee. idecode: In-distribution equivariance for conformal out-of-distribution detection. In *AAAI*, 2022.
- [53] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [54] Adji B. Dieng, Yoon Kim, Alexander M. Rush, and David M. Blei. Avoiding latent variable collapse with generative skip models. In *International Conference on Artificial Intelligence and Statistics*, pages 2397–2405, 2019.
- [55] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian J. Goodfellow. Adversarial autoencoders. *CoRR*, abs/1511.05644, 2015.
- [56] Lars M. Mescheder, Sebastian Nowozin, and Andreas Geiger. Adversarial variational bayes: Unifying variational autoencoders and generative adversarial networks. In *ICML*, 2017.
- [57] James Lucas, George Tucker, Roger B Grosse, and Mohammad Norouzi. Don’t blame the elbo! a linear vae perspective on posterior collapse. *Advances in Neural Information Processing Systems*, 32, 2019.
- [58] Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.
- [59] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, 2013.
- [60] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *CoRR*, abs/1812.01718, 2018.
- [61] Yaroslav Bulatov. notMNIST dataset. <http://yaroslavvb.blogspot.com/2011/09/notmnist-dataset.html>.
- [62] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

- 458 [63] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the
459 wild. In *International Conference on Computer Vision, ICCV 2015*.
- 460 [64] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A
461 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and
462 Machine Intelligence*, 2017.
- 463 [65] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large
464 number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*,
465 2008.
- 466 [66] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the
467 wild: A database for studying face recognition in unconstrained environments. Technical Report
468 07-49, University of Massachusetts, Amherst, October 2007.
- 469 [67] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. SUN
470 database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.
- 471 [68] Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in
472 unsupervised feature learning. In *AISTATS*, 2011.
- 473 [69] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel.
474 Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark.
475 In *International Joint Conference on Neural Networks*, number 1288, 2013.
- 476 [70] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the
477 wild. In *CVPR*, 2014.
- 478 [71] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep
479 convolutional neural networks. In *NeurIPS*, 2012.
- 480 [72] Xingjian Shi, Zhouong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun
481 Woo. Convolutional LSTM network: A machine learning approach for precipitation nowcasting.
482 In *NeurIPS*, 2015.

483 Appendix

484 A More Background on OOD Detection

485 To provide a clear distinction and avoid confusion between supervised and unsupervised OOD
 486 detection, we delineate the key differences here, primarily focusing on their respective setups.

487 **Setup of *unsupervised* OOD detection.** Denoting the input space with \mathcal{X} , an *unlabeled* training
 488 dataset $\mathcal{D}_{\text{train}} = \{x_i\}_{i=1}^N$ containing of N data points can be obtained by sampling *i.i.d.* from a data
 489 distribution $\mathcal{P}_{\mathcal{X}}$. Typically, we treat the $\mathcal{P}_{\mathcal{X}}$ as p_{id} , which represents the in-distribution (ID) [17, 27].
 490 With this *unlabeled* training set, unsupervised OOD detection is to design a score function $\mathcal{S}(x)$ that
 491 can determine whether an input is ID or OOD.

492 **Setup of *supervised* OOD detection.** Compared with the setup of unsupervised OOD detection,
 493 supervised one needs to additionally introduce a label space $\mathcal{Y} = \{1, \dots, k\}$ with k classes, and the
 494 training set becomes $\mathcal{D}_{\text{train}} = \{(x_i, y_i)\}_{i=1}^N$. Then, it typically needs to train a classifier $f : \mathcal{X} \rightarrow \mathbb{R}^k$,
 495 and OOD detection can be achieved based on the property of the classifier [4, 7, 9].

We illustrate the distinction between supervised and unsupervised OOD detection in Figure 8.

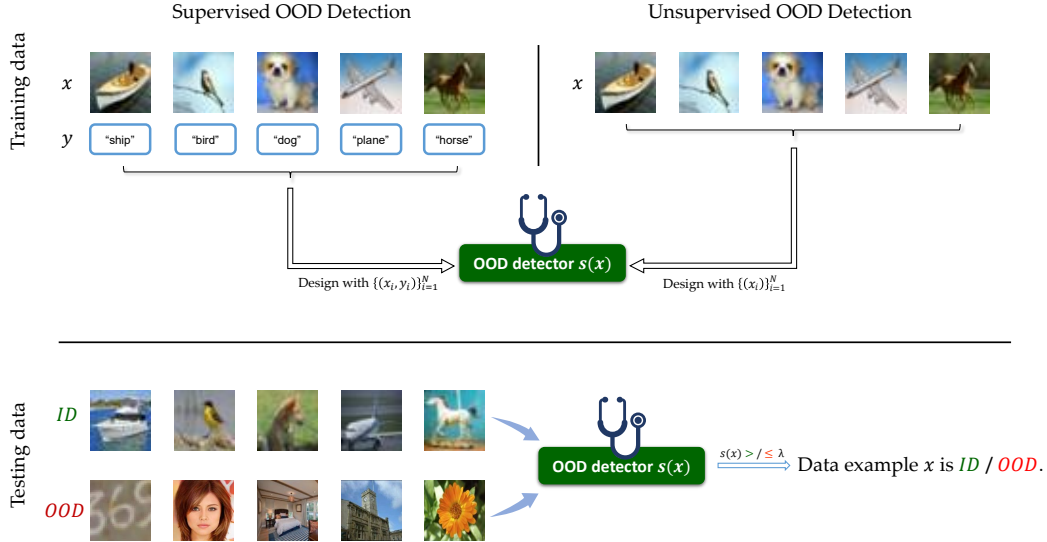


Figure 8: An illustration showcasing the difference between supervised and unsupervised OOD detection.

496

497 B Related Work

498 B.1 Deep Generative Models

499 Deep Generative Models (DGMs) have been developed with the aim of modeling the true data
 500 distribution $p(x)$, leveraging deep neural networks to learn a generative process [53]. These models
 501 span several types, mainly including the autoregressive model [19, 20], flow model [21, 22], generative
 502 adversarial network [24], diffusion model [23], and variational autoencoder (VAE) [25]. Below,
 503 we briefly introduce each of these models: The autoregressive model operates under the premise
 504 that a data sample x is a sequential series, implying that the value of a pixel in an image is only
 505 dependent on the pixels preceding it. The flow model comes with an inherent requirement for the
 506 invertibility of the projection between x and z , which imposes constraints on the implementation
 507 of its backbone. The generative adversarial network adopts an additional discriminator to implicitly
 508 learn the data distribution. Despite its power, it faces challenges such as unstable training and mode

collapse [28]. The diffusion model, trained using a score-based method, has the drawback of being slow in sampling due to its multiple stochastic layers. Among these models, VAE stands out for its flexibility in implementation, comprehensive mode coverage, and fast sampling [28]. However, its training objective, an evidence lower bound of the data distribution, presents difficulties for analysis.

B.2 VAE-based Unsupervised OOD Detection

Given the advantages of flexibility, comprehensive mode coverage, and fast sampling capabilities, variational autoencoder (VAE)-based methods have emerged as a promising choice for unsupervised out-of-distribution (OOD) detection. Based on the necessity to modify the training of VAE, these methods can be categorized into two groups. *i)* The first group includes methods that modify the training of VAE. Hierarchical VAE expands the VAE’s layers to augment its representational capacity [15], yet the improvements in performance are marginal, and the issue of *overestimation* persists. The adaptive log-likelihood ratio method, \mathcal{LLR}^{ada} , is also grounded in the hierarchical VAE and introduces a generative skip connection to propagate information to higher layers of latent variables [18]. It utilizes the differences between each layer of latent variables for OOD detection, achieving state-of-the-art performance despite certain shortcomings as discussed in section 5.3. The tilted variational autoencoder enforces the latent variable to exist within the sphere of a tilted Gaussian [16], thereby disrupting the efficient, widely adopted reparameterization based on the Gaussian. It should be noted that modifying the training of VAE may be less practical as the proposed method cannot be directly applied to other VAEs. This implies that applying the OOD detection method to a new advanced VAE necessitates meticulous training using the new modification method. *ii)* The second group of methods attempts to utilize the properties of a trained VAE for OOD detection without modifying it. The likelihood-ratio method simulates the background using noise and employs the difference between the original and simulated background images for OOD detection [12]. The likelihood-regret method finetunes the trained VAE with the test sample to observe changes in likelihood [14]. The log-likelihood ratio method leverages the assumption that latent variables of lower layers capture low-level features of inputs while those of higher layers grasp semantic features [17]. The difference between these latent variables can then be used for OOD detection. WAIC utilizes empirical ensemble methods for OOD detection [26]. However, it should be stressed that none of these methods have strived to provide an exhaustive theoretical analysis of the VAE’s *overestimation* issue.

C Derivation of the Analysis

C.1 Derivation for Eq. (5)

We first give the definition of the mutual information $\mathcal{I}_q(\mathbf{x}, \mathbf{z})$ as follows:

$$\begin{aligned}\mathcal{I}_q(\mathbf{x}, \mathbf{z}) &= \int_{\mathbf{x}} \int_{\mathbf{z}} q(\mathbf{x}, \mathbf{z}) \log \frac{q(\mathbf{x}, \mathbf{z})}{p(\mathbf{x})q(\mathbf{z})} \\ &= \int_{\mathbf{x}} \int_{\mathbf{z}} q_{\phi}(\mathbf{z}|\mathbf{x})p(\mathbf{x}) \log \frac{q_{\phi}(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})} \\ &= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}), \mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} [\log \frac{q_{\phi}(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})}],\end{aligned}\tag{21}$$

where the data distribution $p(\mathbf{x})$ should actually be replaced by $q(\mathbf{x})$, *i.e.*, the data distribution given the observed data points in the whole training set, when the size of the training set is big enough, *i.e.*, $q(\mathbf{x})$ is close to $p(\mathbf{x})$; and the $q(\mathbf{z})$ is called the aggregated posterior distribution [54, 55, 56], expressed as:

$$q(\mathbf{z}) = \int_{\mathbf{x}} q_{\phi}(\mathbf{z}|\mathbf{x})p(\mathbf{x}).\tag{22}$$

Recall that Eq. (5) comprises two components, denoted as:

$$\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [\text{ELBO}(\mathbf{x})] = \overbrace{\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [\mathbb{E}_{\mathbf{z} \sim q_{\phi}(\mathbf{z}|\mathbf{x})} \log p_{\theta}(\mathbf{x}|\mathbf{z})]}^{L_1} - \overbrace{\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [D_{\text{KL}}(q_{\phi}(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))]}^{L_2}.\tag{23}$$

547 Let's begin with the second component:

$$\begin{aligned}
L_2 &= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})}] \\
&= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log [\frac{q_\phi(\mathbf{z}|\mathbf{x})}{p(\mathbf{z})} \cdot \frac{q(\mathbf{z})}{q(\mathbf{z})}]] \\
&= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log [\frac{q_\phi(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})} \cdot \frac{q(\mathbf{z})}{p(\mathbf{z})}]] \\
&= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})}] + \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log [\frac{q(\mathbf{z})}{p(\mathbf{z})}]] \\
&= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}), \mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})}] + \mathbb{E}_{\mathbf{z} \sim q(\mathbf{z})} [\log \frac{q(\mathbf{z})}{p(\mathbf{z})}] \\
&= \mathcal{I}_q(\mathbf{x}, \mathbf{z}) + D_{\text{KL}}(q(\mathbf{z})||p(\mathbf{z})).
\end{aligned} \tag{24}$$

548 Before embarking on a similar derivation for the first component, it's crucial to comprehend the
549 notation “ q ” and “ p ” within the context of VAE. Here, “ q ” signifies an approximated distribution
550 given observed data, typically parameterized by a neural network, while “ p ” represents the actual
551 distribution. For example, $q_\phi(\mathbf{z}|\mathbf{x})$ denotes the approximated posterior distribution, and its corre-
552 sponding true posterior is $p(\mathbf{z}|\mathbf{x})$. The gap between $q_\phi(\mathbf{z}|\mathbf{x})$ and $p(\mathbf{z}|\mathbf{x})$ contributes to the concept
553 of a "lower bound", as depicted by

$$\log p(\mathbf{x}) = \text{ELBO}(\mathbf{x}) + D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x})). \tag{25}$$

554 However, it may appear that $p_\theta(\mathbf{x}|\mathbf{z})$, approximated by a decoder whose parameter is θ , should be
555 represented as $q(\mathbf{x}|\mathbf{z})$. This particular interpretation arises due to the fact that the global optimum
556 of the decoder's parameters in the ELBO coincides with the global maximum of the marginal
557 likelihood of the observed data [57]. Specifically, this means that the generative process $p_\theta(\mathbf{x}) =$
558 $\int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$ achieves optimality once the VAE has been trained to reach the ELBO's optimum.
559 Thus, after the VAE is well trained and the data distribution $q(\mathbf{x})$ of the observed data points in the
560 training set could well represent the true data distribution $p(\mathbf{x})$, implying that $p_\theta(\mathbf{x}|\mathbf{z})$'s parameters
561 reach the maximum likelihood estimation given the observed training data, we can state the following:

$$p_\theta(\mathbf{x}|\mathbf{z}) = q(\mathbf{x}|\mathbf{z}) = \frac{q(\mathbf{x}, \mathbf{z})}{q(\mathbf{z})} = \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})} q(\mathbf{x}) = \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})} p(\mathbf{x}). \tag{26}$$

562 Inserting this into the first component of Eq. (5), we obtain the following result:

$$\begin{aligned}
L_1 &= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z})] \\
&= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [\mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log [\frac{q_\phi(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})} p(\mathbf{x})]] \\
&= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})} + \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} \log p(\mathbf{x}) \\
&= \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x}), \mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})} [\log \frac{q_\phi(\mathbf{z}|\mathbf{x})}{q(\mathbf{z})}] + \mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} \log p(\mathbf{x}) \\
&= \mathcal{I}_q(\mathbf{x}, \mathbf{z}) - \mathcal{H}_p(\mathbf{x}).
\end{aligned} \tag{27}$$

563 Hence, we can achieve the following expression:

$$\mathbb{E}_{\mathbf{x} \sim p(\mathbf{x})} [\text{ELBO}(\mathbf{x})] = -\mathcal{H}_p(\mathbf{x}) - D_{\text{KL}}(q(\mathbf{z})||p(\mathbf{z})). \tag{28}$$

564 C.2 Toy Examples' Details

565 **Single-modal case setup.** In this scenario, the data distribution is determined by a standard 2-
566 dimensional Gaussian distribution $p(\mathbf{x}) = \mathcal{N}(\mathbf{x}|\mathbf{0}, \Sigma_{\mathbf{x}})$, where

$$\Sigma_{\mathbf{x}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{29}$$

567 In order to simulate the dimension-reduction property of VAE, we designate the dimension of the
 568 latent variable as 1-dimensional; that is, the variance \mathbf{I} in $p(\mathbf{z})$ reduces to 1. Under this configuration,
 569 we *i.i.d.* sample $N = 5000$ data points from the data distribution $p(\mathbf{x})$ to construct a training set.
 570 Each parameter's solutions are calculated analytically.

571 **Multi-modal case setup.** The data distribution is made by a mixture of two standard single-modal
 572 Gaussian distributions, *i.e.*, $p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, where $K = 2$, $\pi_k = 1/2$ and

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 3 \\ 3 \end{bmatrix}, \boldsymbol{\mu}_2 = \begin{bmatrix} -3 \\ -3 \end{bmatrix}, \boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \quad (30)$$

573 The training set of this multi-modal case is built by *i.i.d.* sampling from 5000 data points from each
 574 component Gaussian distribution $\mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$, *i.e.*, 10000 data points in total.

575 C.3 Derivation for Single-modal Case in Section 3.2

576 Assume we have a dataset containing N data samples $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$, $\mathbf{x}_i \in \mathbb{R}^d$, $d = 2$, and we
 577 already know the groundtruth distribution of it, *i.e.*,

$$p(\mathbf{x}) = \mathcal{N}(\mathbf{x} | \mathbf{0}, \boldsymbol{\Sigma}_x), \quad (31)$$

578 where $\boldsymbol{\Sigma}_x = \mathbf{I}$. We have a linear VAE model parameterized as:

$$p(\mathbf{z}) = \mathcal{N}(\mathbf{z} | \mathbf{0}, \mathbf{I}) \quad (32)$$

$$q_\phi(\mathbf{z} | \mathbf{x}) = \mathcal{N}(\mathbf{z} | \mathbf{A}\mathbf{x} + \mathbf{B}, \mathbf{C}) \quad (33)$$

$$p_\theta(\mathbf{x} | \mathbf{z}) = \mathcal{N}(\mathbf{x} | \mathbf{E}\mathbf{z} + \mathbf{F}, \sigma^2 \mathbf{I}), \quad (34)$$

579 where $p(\mathbf{z})$ is the prior distribution, $\mathbf{z} \in \mathbb{R}^q$, $q = 1$, $q_\phi(\mathbf{z} | \mathbf{x})$ is the approximated posterior distribution,
 580 and $p_\theta(\mathbf{x} | \mathbf{z})$ is the approximated likelihood distribution. Directly employing the knowledge from
 581 probabilistic Principal Component Analysis (pPCA) [58], we could get the maximum likelihood
 582 estimation of $p_\theta(\mathbf{x} | \mathbf{z})$:

$$\sigma_{\text{MLE}}^2 = \frac{1}{d - q} \sum_{j=q+1}^d \lambda_j \quad (35)$$

$$\mathbf{E}_{\text{MLE}} = \mathbf{U}_q (\boldsymbol{\Lambda}_q - \sigma_{\text{MLE}}^2 \mathbf{I})^{1/2} \mathbf{R} \quad (36)$$

$$\mathbf{F}_{\text{MLE}} = \mathbf{0} \quad (37)$$

583 where $\lambda_{q+1}, \dots, \lambda_d$ are the smallest eigenvalues of the sample covariance matrix $\mathbf{S} = \frac{1}{N} \sum_{n=1}^N \mathbf{x} \mathbf{x}^\top$,
 584 the $d \times q$ orthogonal matrix \mathbf{U}_q is made by the q dominant eigenvectors of \mathbf{S} , the diagonal matrix $\boldsymbol{\Lambda}_q$
 585 contains the corresponding q largest eigenvalues, and \mathbf{R} is an arbitrary $q \times q$ orthogonal matrix. Note
 586 that, when $q = 1$, we have $\mathbf{R} = \mathbf{I}$. After we get the parameters of $p_\theta(\mathbf{x} | \mathbf{z})$, we could get the $p(\mathbf{z} | \mathbf{x})$
 587 by Bayes rule:

$$\begin{aligned} p(\mathbf{z} | \mathbf{x}) &= \frac{p_\theta(\mathbf{x} | \mathbf{z}) p(\mathbf{z})}{p(\mathbf{x})} \\ &= \mathcal{N}(\mathbf{z} | \boldsymbol{\Sigma}_x^{-1} \mathbf{E}_{\text{MLE}}^\top \mathbf{x}, \sigma_{\text{MLE}}^2 \boldsymbol{\Sigma}_x^{-1}), \end{aligned} \quad (38)$$

588 where $\boldsymbol{\Sigma}_x = \mathbf{E}_{\text{MLE}}^\top \mathbf{E}_{\text{MLE}} + \sigma_{\text{MLE}}^2 \mathbf{I}$. Thus, the maximum likelihood estimates of $q_\phi(\mathbf{z} | \mathbf{x})$'s parameters
 589 are:

$$\mathbf{A}_{\text{MLE}} = \boldsymbol{\Sigma}_x^{-1} \mathbf{E}_{\text{MLE}}^\top \quad (39)$$

$$\mathbf{B}_{\text{MLE}} = \mathbf{0} \quad (40)$$

$$\mathbf{C}_{\text{MLE}} = \sigma_{\text{MLE}}^2 \boldsymbol{\Sigma}_x^{-1}. \quad (41)$$

590 Although the maximum likelihood estimations are ascertained, it remains necessary to verify whether
 591 these estimations allow the ELBO to reach the global optimum. The derivation of ELBO is as follows:

$$\begin{aligned} \log p(\mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z} | \mathbf{x})} [\log p(\mathbf{x} | \mathbf{z})] - D_{\text{KL}}(q_\phi(\mathbf{z} | \mathbf{x}) || p(\mathbf{z})) + D_{\text{KL}}(q_\phi(\mathbf{z} | \mathbf{x}) || p(\mathbf{z} | \mathbf{x})) \\ &= \text{ELBO}(\mathbf{x}) + D_{\text{KL}}(q_\phi(\mathbf{z} | \mathbf{x}) || p(\mathbf{z} | \mathbf{x})). \end{aligned} \quad (42)$$

592 Given that $q_\phi(\mathbf{z} | \mathbf{x}) = \mathcal{N}(\mathbf{z} | \boldsymbol{\Sigma}_x^{-1} \mathbf{E}_{\text{MLE}}^\top \mathbf{x}, \sigma_{\text{MLE}}^2 \boldsymbol{\Sigma}_x^{-1}) = p(\mathbf{z} | \mathbf{x})$, $D_{\text{KL}}(q_\phi(\mathbf{z} | \mathbf{x}) || p(\mathbf{z} | \mathbf{x}))$ becomes
 593 zero. Furthermore, any modifications to the parameters of q_ϕ would result in an increase of

594 $D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{x}|\mathbf{z}))$; in other words, it would result in a decrease of ELBO. Hence, the global
 595 optimum of the ELBO is attained when $\mathbf{A}_{\text{MLE}} \sim \mathbf{E}_{\text{MLE}}, \sigma_{\text{MLE}}$ are implemented in the linear VAE.
 596 Moreover, in this situation, $\log p(\mathbf{x})$ equates to ELBO.

597 Finally, we could get the expression of the aggregated posterior distribution $q(\mathbf{z})$:

$$\begin{aligned}
 q(\mathbf{z}) &= \int_{\mathbf{x}} q_\phi(\mathbf{z}|\mathbf{x})p(\mathbf{x}) \\
 &= \int_{\mathbf{x}} \mathcal{N}(\mathbf{z}|\Sigma_{\mathbf{x}}^{-1}\mathbf{E}_{\text{MLE}}^\top\mathbf{x}, \sigma_{\text{MLE}}^2\Sigma_{\mathbf{x}}^{-1})\mathcal{N}(\mathbf{x}|\mathbf{0}, \Sigma_{\mathbf{x}}) \\
 &= \int_{\mathbf{x}} \mathcal{N}(\mathbf{z}|\mathbf{I}^{-1}\mathbf{E}_{\text{MLE}}^\top\mathbf{x}, \sigma_{\text{MLE}}^2\mathbf{I}^{-1})\mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{I}) \\
 &= \int_{\mathbf{x}} \mathcal{N}(\mathbf{z}|\mathbf{E}_{\text{MLE}}^\top\mathbf{x}, \sigma_{\text{MLE}}^2\mathbf{I})\mathcal{N}(\mathbf{x}|\mathbf{0}, \mathbf{I}) \\
 &= \mathcal{N}(\mathbf{0}, \mathbf{E}_{\text{MLE}}^\top\mathbf{E}_{\text{MLE}} + \sigma_{\text{MLE}}^2\mathbf{I}) \\
 &= \mathcal{N}(\mathbf{0}, \Sigma_{\mathbf{x}}) \\
 &= \mathcal{N}(\mathbf{0}, \mathbf{I}) \\
 &= p(\mathbf{z}).
 \end{aligned} \tag{43}$$

598 In summing up the single-modal case, our assertion is that $D_{\text{KL}}[q(\mathbf{z})||p(\mathbf{z})] = 0$, indicating that the
 599 design of the prior distribution is appropriate and would not result in an *overestimation* of VAE.

600 C.4 Derivation for Multi-modal Case in Section 3.2

601 Assume we have a distribution $p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)$ and we build a dataset containing
 602 $K \times N$ data samples, which is made by sampling N data samples from each $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)$. The
 603 parameterization setting of the $p(\mathbf{z})$, $q_\phi(\mathbf{z}|\mathbf{x})$, and $p_\theta(\mathbf{x}|\mathbf{z})$ is the same as the single-modal case in
 604 Section 3.2.

605 Deriving from the single-modal scenario, an analytical formulation of $D_{\text{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}|\mathbf{x}))$ is
 606 unattainable in the multi-modal case. Thus, it necessitates a derivation directly from the ELBO.
 607 Due to the fact that the global optimum of the decoder's parameters in the ELBO coincides with
 608 the global maximum of the marginal likelihood of the observed data [57], we firstly commence
 609 with the derivation of the maximum likelihood estimation of $p_\theta(\mathbf{x}|\mathbf{z})$. Despite the feasibility of
 610 directly obtaining the maximum likelihood estimation of the parameters in $p_\theta(\mathbf{x}|\mathbf{z})$ by optimizing the
 611 integration $\hat{p}_\theta(\mathbf{x}) = \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})$ using the observed data, we propose an additional clarification
 612 connecting this integration and the ELBO. With reference to the strictly tighter importance sampling
 613 on the ELBO [36], we can derive that

$$\text{ELBO}^S(\mathbf{x}) = \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log \frac{1}{S} \sum_{s=1}^S \frac{p_\theta(\mathbf{x}|\mathbf{z}^{(s)})p(\mathbf{z}^{(s)})}{q_\phi(\mathbf{z}^{(s)}|\mathbf{x})}]. \tag{44}$$

614 Setting the number of instances $S = 1$, $\text{ELBO}^S(\mathbf{x})$ equates to the regular ELBO(\mathbf{x}). As S approaches
 615 $+\infty$, it follows that

$$\begin{aligned}
 \text{ELBO}^S(\mathbf{x}) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})}] \\
 &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log \int_{\mathbf{z}} q_\phi(\mathbf{z}|\mathbf{x}) \frac{p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x})} d\mathbf{z}] \\
 &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z}) d\mathbf{z}] \\
 &= \log \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z}) d\mathbf{z} \\
 &= \log \hat{p}_\theta(\mathbf{x}).
 \end{aligned} \tag{45}$$

616 The expression of $\hat{p}_\theta(\mathbf{x})$ is shown as:

$$\begin{aligned}\hat{p}_\theta(\mathbf{x}) &= \int_{\mathbf{z}} p_\theta(\mathbf{x}|\mathbf{z})p(\mathbf{z}) \\ &= \int_{\mathbf{z}} \mathcal{N}(\mathbf{x}|\mathbf{E}\mathbf{z} + \mathbf{F}, \sigma^2\mathbf{I})\mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I}) \\ &= \mathcal{N}(\mathbf{x}|\mathbf{F}, \mathbf{E}\mathbf{E}^\top + \sigma^2\mathbf{I}).\end{aligned}\quad (46)$$

617 Then, the joint log-likelihood of the observed dataset $\{\mathbf{x}_i^{(k)}\}_{i=1, k=1}^{N, K}$ can be formulated as:

$$\mathcal{L} = \sum_{k=1}^K \sum_{i=1}^N \log \hat{p}_\theta(\mathbf{x}_i^{(k)}) = -\frac{KNd}{2} \log(2\pi) - \frac{KN}{2} \log \det(\mathbf{M}) - \frac{KN}{2} \text{tr}[\mathbf{M}^{-1}\mathbf{S}], \quad (47)$$

618 where $\mathbf{M} = \mathbf{E}\mathbf{E}^\top + \sigma^2\mathbf{I}$ and $\mathbf{S} = \frac{1}{KN} \sum_{k=1}^K \sum_{i=1}^N (\mathbf{x}_i^{(k)} - \mathbf{F})(\mathbf{x}_i^{(k)} - \mathbf{F})^\top$.

619 Repeatedly using the knowledge in pPCA again, we could get the maximum likelihood estimation of
620 the parameters:

$$(\sigma^*)^2 = \frac{1}{d-q} \sum_{j=q+1}^d \lambda_j \quad (48)$$

$$\mathbf{E}^* = \mathbf{U}_q (\mathbf{\Lambda}_q - (\sigma^*)^2)^{1/2} \mathbf{R} \quad (49)$$

$$\mathbf{F}^* = \mathbf{0}, \quad (50)$$

621 where $\lambda_{q+1}, \dots, \lambda_d$ are the smallest eigenvalues of the sample covariance matrix $\mathbf{S} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}\mathbf{x}^\top$,
622 the $d \times q$ orthogonal matrix \mathbf{U}_q is made by the q dominant eigenvectors of \mathbf{S} , the diagonal matrix $\mathbf{\Lambda}_q$
623 contains the corresponding q largest eigenvalues, and \mathbf{R} is an arbitrary $q \times q$ orthogonal matrix. Note
624 that, when $q = 1$, we have $\mathbf{R} = \mathbf{I}$. Actually, with the same $p(\mathbf{z})$ and a decoder $p_\theta(\mathbf{x}|\mathbf{z})$ parameterized
625 by the same linear network, the expression of the maximum likelihood estimation of the $p_\theta(\mathbf{x}|\mathbf{z})$ in
626 the multi-modal case is the same as the single-modal case.

627 In order to determine $q_\phi(\mathbf{z}|\mathbf{x})$'s parameters, we can initiate the process by identifying the stationary
628 points of $q_\phi(\mathbf{z}|\mathbf{x})$ with respect to the ELBO. The ELBO can be analytically expressed as follows:

$$\text{ELBO}(\mathbf{x}) = \overbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]}^{L_1} - \overbrace{D_{\text{KL}}[q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})]}^{L_2} \quad (51)$$

$$\begin{aligned}L_1 &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[-\frac{(\mathbf{E}\mathbf{z} - \mathbf{x})^\top (\mathbf{E}\mathbf{z} - \mathbf{x})}{2\sigma^2} - \frac{d}{2} \log 2\pi\sigma^2 \right] \\ &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} \left[-\frac{(\mathbf{E}\mathbf{z})^\top (\mathbf{E}\mathbf{z}) + 2\mathbf{x}^\top \mathbf{E}\mathbf{z} - \mathbf{x}^\top \mathbf{x}}{2\sigma^2} - \frac{d}{2} \log(2\pi\sigma^2) \right] \\ &= \frac{1}{2\sigma^2} [-\text{tr}(\mathbf{E}\mathbf{C}\mathbf{E}^\top) - (\mathbf{E}\mathbf{A}\mathbf{x} + \mathbf{E}\mathbf{B})^\top (\mathbf{E}\mathbf{A}\mathbf{x} + \mathbf{E}\mathbf{B}) + 2\mathbf{x}^\top (\mathbf{E}\mathbf{A}\mathbf{x} + \mathbf{E}\mathbf{B}) - \mathbf{x}^\top \mathbf{x}] \\ &\quad - \frac{d}{2} \log(2\pi\sigma^2) \quad (52)\end{aligned}$$

$$L_2 = \frac{1}{2} [-\log \det(\mathbf{C}) + (\mathbf{A}\mathbf{x} + \mathbf{B})^\top (\mathbf{A}\mathbf{x} + \mathbf{B}) + \text{tr}(\mathbf{C}) - q] \quad (53)$$

629 For a dataset consisting of KN data samples, the stationary points with respect to the ELBO can be
630 obtained through the following expressions:

$$\frac{\partial(\sum^{KN} \text{ELBO}(\mathbf{x}))}{\partial \mathbf{A}} = KN[-\mathbf{A}\mathbf{S} - \mathbf{B}\bar{\mathbf{x}}^\top - \frac{1}{\sigma^2}(\mathbf{E}^\top \mathbf{E}\mathbf{A}\mathbf{S}) - \frac{1}{\sigma^2}(\mathbf{E}^\top \mathbf{E}\mathbf{B}\bar{\mathbf{x}}^\top - \mathbf{E}^\top \mathbf{S})] = \mathbf{0} \quad (54)$$

$$\frac{\partial(\sum^{KN} \text{ELBO}(\mathbf{x}))}{\partial \mathbf{B}} = KN[-\mathbf{A}\bar{\mathbf{x}} - \frac{1}{\sigma^2} \mathbf{E}^\top \mathbf{E}\mathbf{A}\bar{\mathbf{x}} + \frac{1}{\sigma^2} \mathbf{E}^\top \bar{\mathbf{x}} - (\mathbf{I} + \frac{\mathbf{E}^\top \mathbf{E}}{\sigma^2})\mathbf{B}] = \mathbf{0} \quad (55)$$

$$\frac{\partial(\sum^{KN} \text{ELBO}(\mathbf{x}))}{\partial \mathbf{C}} = \frac{KN}{2} ((\mathbf{C}^{-1})^\top - \mathbf{I} - \frac{1}{\sigma^2}(\mathbf{E}^\top \mathbf{E})) = \mathbf{0}, \quad (56)$$

631 where $\mathbf{S} = \frac{1}{KN} \sum^{KN} \mathbf{x}\mathbf{x}^\top$ and $\bar{\mathbf{x}} = \frac{1}{KN} \sum^{KN} \mathbf{x}$. Upon further investigation, we have discovered
 632 that the stationary points of \mathbf{A} , \mathbf{B} , and \mathbf{C} solely depend on the parameters \mathbf{E} and σ . In mathematical
 633 terms, they can be expressed as:

$$\mathbf{A}^* = \frac{(\mathbf{I} + \frac{1}{\sigma^2} \mathbf{E}^\top \mathbf{E})^{-1}}{\sigma^2} \mathbf{E}^\top \quad (57)$$

$$\mathbf{B}^* = \mathbf{0} \quad (58)$$

$$\mathbf{C}^* = ((\mathbf{I} + \frac{1}{\sigma^2} \mathbf{E}^\top \mathbf{E})^\top)^{-1}. \quad (59)$$

634 Finally, we can derive the expression of $q(\mathbf{z})$ in this multi-modal case as follows:

$$\begin{aligned} q(\mathbf{z}) &= \int_{\mathbf{x}} q_\phi(\mathbf{z}|\mathbf{x})p(\mathbf{x}) \\ &= \int_{\mathbf{x}} \mathcal{N}(\mathbf{z}|\mathbf{A}^* \mathbf{x}, \mathbf{C}^*) \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &= \sum_{k=1}^K \pi_k \int_{\mathbf{x}} \mathcal{N}(\mathbf{z}|\mathbf{A}^* \mathbf{x}, \mathbf{C}^*) \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \\ &= \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{z}|\mathbf{A}^* \boldsymbol{\mu}_k, \mathbf{A}^* \boldsymbol{\Sigma}_k (\mathbf{A}^*)^\top + \mathbf{C}^*) \\ &\neq p(\mathbf{z}). \end{aligned} \quad (60)$$

635 In conclusion, we observe that $D_{\text{KL}}[q(\mathbf{z})||p(\mathbf{z})] \neq 0$, indicating that the design of the prior distri-
 636 bution $p(\mathbf{z})$ is not appropriate in this multi-modal case and may result in *overestimation* issue of
 637 VAE.

638 C.5 Implementation Details of Deep VAE in Section 3.2

639 The non-linear deep VAE’s encoder is implemented as a 3-layer MLP, which takes the 2D data points
 640 as inputs. The encoder consists of two linear layers with a hidden dimension of 10 and LeakyReLU
 641 activation function [59]. The output layer, with a dimension of 2, does not have an activation function
 642 and provides the values for μ_z and $\log \sigma_z^2$ for each dimension of the latent variable.

643 For the decoder, it takes the sampled latent variable \mathbf{z} through reparameterization and feeds it into
 644 two linear layers with a hidden dimension of 10 and LeakyReLU activation function. The final output
 645 is obtained by a linear layer without activation function, with a dimension of 4. The reconstruction
 646 likelihood is modeled as a Gaussian distribution, where the first two dimensions represent $\boldsymbol{\mu}_x$ (the
 647 mean of the reconstruction likelihood) and the remaining dimension represents $\log \sigma_x^2$ (the log
 648 variance of the reconstruction likelihood).

649 The deep VAE is trained using the Adam optimizer [50] with a learning rate of 1e-5. The training set
 650 consists of a total of 10,000 data points.

651 D Details of the Non-scaled Entropy Calibration Method

652 We provide a pseudo code here for calculating the $\mathcal{C}_{\text{non}}(\mathbf{x})$ of a testing sample \mathbf{x} in Algorithm 1.
 653 Noted that, the maximum number of singular values N should be larger than n_{id} .

654 E Details of Experimental Setup

655 E.1 Description of all Datasets

656 For grayscale image datasets, we utilize the following datasets: FashionMNIST [40], MNIST [41],
 657 KMNIST [60], notMNIST [61], Omniglot [62], and several grayscale datasets transformed from
 658 RGB datasets. **FashionMNIST** is a dataset consisting of 60,000 grayscale images of Zalando’s article

Algorithm 1 Non-scaled dataset entropy calibration $\mathcal{C}_{\text{non}}(\mathbf{x})$ algorithm

Input: Hyperparameter n_{id} and its corresponding reconstruction error $\epsilon = \mathbb{E}_{\mathbf{x} \sim p_{\text{id}}} |\mathbf{x}_{\text{recon}} - \mathbf{x}|$, maximum number of singular values N , a testing sample \mathbf{x} .
Output: $\mathcal{C}_{\text{non}}(\mathbf{x})$.
Do SVD for the testing sample \mathbf{x} ;
for $n_i = 1$ to N **do**
 Calculate reconstruction error ϵ_i using n_i singular values;
 if $\epsilon_i \leq \epsilon$ **then**
 break;
 end if
end for
if $n_i < n_{\text{id}}$ **then**
 Calculate $\mathcal{C}_{\text{non}}(\mathbf{x}) = n_i / n_{\text{id}}$;
else
 Calculate $\mathcal{C}_{\text{non}}(\mathbf{x}) = (n_{\text{id}} - (n_i - n_{\text{id}})) / n_{\text{id}}$;
end if
return $\mathcal{C}_{\text{non}}(\mathbf{x})$

659 pictures for training, and 10,000 images for testing. Each image is 28x28 pixels and belongs to one of
660 the 10 classes. **MNIST** is a widely used dataset containing 70,000 grayscale images of handwritten
661 digits. It consists of a training set of 60,000 images and a test set of 10,000 images. Each image is
662 28x28 pixels. **KMNIST** is derived from the Kuzushiji Dataset and serves as a drop-in replacement
663 for the MNIST dataset. It includes 70,000 grayscale images, each with a resolution of 28x28 pixels.
664 **notMNIST** is a dataset composed of 547,838 grayscale images of glyphs extracted from publicly
665 available fonts. The images are 28x28 pixels in size and cover letters A to J from various fonts.
666 **Omniglot** contains 32,460 grayscale images of 1623 different handwritten characters from 50 distinct
667 alphabets. Each image has a resolution of 28x28 pixels. Additionally, we have transformed several
668 RGB datasets into grayscale versions, including CIFAR10-G, CIFAR100-G, SVHN-G, CelebA-G,
669 SUN-G, and Places365-G.

670 For RGB datasets, we utilize the following datasets: CIFAR10/CIFAR100 [42], SVHN [43], CelebA
671 [63], Places365 [64], Flower102 [65], LFWPeople [66], SUN [67], STL10 [68], GTSRB [69],
672 and DTD [70] datasets. **CIFAR10** and **CIFAR100** are datasets consisting of 32x32 color images.
673 CIFAR10 contains 50,000 training images and 10,000 testing images, with 10 different classes.
674 CIFAR100 has the same number of images but includes 100 classes. **SVHN** is a dataset obtained
675 from Google Street View images, primarily used for recognizing digits and numbers in natural scene
676 images. **CelebA** is a large-scale face attributes dataset containing over 200,000 celebrity images,
677 each annotated with 40 attribute labels. **Places365** is a dataset that includes 1.8 million training
678 images from 365 scene categories. The validation set contains 50 images per category, and the testing
679 set contains 900 images per category. **Flower102** is an image classification dataset consisting of
680 102 flower categories, with each class containing between 40 and 258 images. The selected flowers
681 are commonly found in the United Kingdom. **LFWPeople** contains more than 13,000 images of
682 faces collected from the web, making it a popular dataset for face-related tasks. **SUN** is a large-scale
683 scene recognition dataset, covering a wide range of scenes from abbey to zoo. **STL10** is an image
684 recognition dataset designed for unsupervised feature learning. It includes labeled data from 10
685 categories and unlabeled data from additional classes. **GTSRB** is a dataset specifically developed
686 for the task of German traffic sign recognition. **DTD** is an evolving collection of textured images in
687 various real-world settings. All images from these datasets are resized to the dimensions of 32x32x3
688 before being used as input for the models.

689 E.2 Description of all Baselines

690 Following the categorization in \mathcal{LLR}^{ada} [18], we provide a detailed description of each baseline
691 within the three categories:

- 692 • “**Supervised**” (Methods using in-distribution data labels y , which is the same as the “Label”
693 category in \mathcal{LLR}^{ada} [18]): maximum softmax classification probability (CP) method [1] and its
694 variants, denoted as “CP”, “CP(OOD)” with OOD as noise class, “CP(Cal)” with calibration on

695 OOD and "CP(Ent)" with entropy of softmax classification probability $p(y|x)$, and Mahalanobis
696 distance (MD) method [46], latent Mahalanobis distance (LMD) method [47], ODIN method [45],
697 VIB method [5], LogitNorm (LN) method [9], GradNorm (GN) method [49], and deep ensembles
698 (DE) method [48] with 20 classifiers;

- 699 • **“Auxiliary”** (Methods using auxiliary knowledge assumptions about ID or OOD data type, which
700 is the same as the “Prior” category in \mathcal{LLR}^{ada} [18]): Likelihood Ratio (LR) method [39] with
701 different backbones, denoted as "LR(PC)" with backbone PixcelCNN, "LR(VAE)" with VAE
702 and "LR(BC)" with binary classifier), Outlier exposure (OE) method [44] and Input complexity
703 (IC) method [13] with different backbones, denoted as "IC(PC)" with backbone PixcelCNN,
704 "IC(Glow)" with backbone Glow and "IC(HVAE)" with backbone HVAE;
- 705 • **“Unsupervised”** (Methods with no OOD-specific assumptions): Ensemble methods: WAIC
706 method [26] with different backbones, denoted as "WAIC (5Glow)" with 5 Glow models, "WAIC
707 (5VAE)" with 5 VAE models and "WAIC (5PC)" with 5 PixcelCNN models; Not ensembles
708 methods: Likelihood regret (LRe) method [14], Log-Likelihood Ratio (HVK) method [17],
709 adaptive Log-Likelihood Ratio (\mathcal{LLR}^{ada}) method [18].

710 E.3 Details of the Implementation

711 The encoder of the VAE is implemented as a 3-layer convolutional network with kernel numbers
712 of 32, 64, and 128, and strides of 1, 2, and 2, respectively. The ReLU [71] activation function is
713 applied. The output layer consists of a linear layer that outputs the mean and log-variance of the
714 latent variables, with a dimension of 200.

715 On the other hand, the decoder takes the reparameterized latent variables as input and utilizes a
716 3-layer transposed convolutional network. The network has kernel numbers of 128, 64, and 32, and
717 strides of 2, 2, and 1, respectively. The ReLU activation function is used. Finally, the output layer
718 is parameterized by a convolutional layer that models the distribution as a discretized mixture of
719 logistics.

720 In the PHP method, an LSTM is employed as the backbone [72]. The hidden size of the LSTM is
721 set to 64, and the outputted hidden state is fed into a 3-layer linear network. The hidden sizes of the
722 linear layers are 64, 32, and 2, respectively. The ReLU activation function is applied to the first two
723 layers. The optimizer used for learning the $q(z)$ distribution is Adam, and the learning rate is set to
724 $1e-4$.

725 F More Ablation Study Results on Verifying the Post-hoc Prior

We evaluate the effectiveness of the PHP method on additional datasets as shown in Table 4.

Table 4: The comparisons of the OOD detection performance of our method on more datasets. The new score function only has **post-hoc prior** part.

ID	FashionMNIST			ID	CIFAR10		
OOD	AUROC \uparrow	AUPRC \uparrow	FPR80 \downarrow	OOD	AUROC \uparrow	AUPRC \uparrow	FPR80 \downarrow
	ELBO / PHP (ours)				ELBO / PHP (ours)		
KMNIST	60.03 / 72.98	54.60 / 69.34	61.6 / 48.1	CIFAR100	52.91 / 55.00	51.15 / 54.01	77.42 / 70.23
Omniglot	99.86 / 99.90	99.89 / 99.89	0.00 / 0.00	CelebA	57.27 / 70.91	54.51 / 72.16	69.03 / 52.95
notMNIST	94.12 / 94.39	94.09 / 94.35	8.29 / 7.79	Places365	57.24 / 57.36	56.96 / 56.55	73.13 / 52.95
CIFAR10-G	98.01 / 98.84	98.24 / 99.13	1.20 / 0.30	LFWPeople	64.15 / 64.57	59.71 / 65.20	59.44 / 64.74
CIFAR100-G	98.49 / 98.50	97.49 / 97.50	1.00 / 0.90	SUN	53.14 / 53.27	54.48 / 54.67	79.52 / 78.12
SVHN-G	95.61 / 96.00	96.20 / 97.13	3.00 / 0.60	STL10	49.37 / 51.07	47.79 / 49.69	78.02 / 75.02
CelebA-G	97.33 / 97.71	94.71 / 95.62	3.00 / 2.20	Flowers102	67.68 / 67.76	64.68 / 64.75	57.94 / 57.63
SUN-G	99.16 / 99.26	99.39 / 99.40	0.00 / 0.00	GTSRB	39.50 / 52.62	41.73 / 50.81	86.61 / 75.12
Places365-G	98.92 / 98.96	98.05 / 98.95	0.80 / 0.60	DTD	37.86 / 43.38	40.93 / 43.99	82.22 / 80.12
Const	94.94 / 95.08	97.27 / 97.35	1.80 / 0.00	Const	0.001 / 15.70	30.71 / 30.78	100.0 / 86.62
Random	99.80 / 99.81	99.90 / 99.90	0.00 / 0.00	Random	71.81 / 72.52	82.89 / 83.42	85.71 / 85.00

G More Ablation Study Results on Verifying the Dataset Entropy Calibration

We evaluate the effectiveness of the DEC method on additional datasets as shown in Table 5.

Table 5: The comparisons of the OOD detection performance of our method on more datasets. The new score function only has **dataset entropy calibration** part.

ID	FashionMNIST			ID	CIFAR10		
OOD	AUROC \uparrow	AUPRC \uparrow	FPR80 \downarrow	OOD	AUROC \uparrow	AUPRC \uparrow	FPR80 \downarrow
	ELBO / DEC (ours)				ELBO / DEC (ours)		
KMNIST	60.03 / 60.54	54.60 / 55.18	61.6 / 60.3	CIFAR100	52.91 / 54.69	51.15 / 52.98	77.42 / 73.23
Omniglot	99.86 / 99.91	99.89 / 99.94	0.00 / 0.00	CelebA	57.27 / 69.00	54.51 / 61.83	69.03 / 50.93
notMNIST	94.12 / 94.50	94.09 / 93.61	8.29 / 6.89	Places365	57.24 / 68.14	56.96 / 65.16	73.13 / 64.26
CIFAR10-G	98.01 / 99.31	98.24 / 99.25	1.20 / 0.40	LFWPeople	64.15 / 67.84	59.71 / 60.28	59.44 / 54.75
CIFAR100-G	98.49 / 98.81	97.49 / 98.05	1.00 / 0.90	SUN	53.14 / 60.55	54.48 / 60.67	79.52 / 68.75
SVHN-G	95.61 / 97.06	96.20 / 97.92	3.00 / 0.00	STL10	49.37 / 64.16	47.79 / 61.76	78.02 / 67.65
CelebA-G	97.33 / 97.69	94.71 / 95.94	3.00 / 2.10	Flowers102	67.68 / 75.59	64.68 / 77.84	57.94 / 46.48
SUN-G	99.16 / 99.58	99.39 / 99.67	0.00 / 0.00	GTSRB	39.50 / 48.35	41.73 / 45.59	86.61 / 73.83
Places365-G	98.92 / 99.14	98.05 / 98.77	0.80 / 0.60	DTD	37.86 / 70.36	40.93 / 60.02	82.22 / 64.16
Const	94.94 / 99.31	97.27 / 99.25	1.80 / 0.40	Const	0.001 / 76.20	30.71 / 83.27	100.0 / 58.04
Random	99.80 / 100.0	99.90 / 100.0	0.00 / 0.00	Random	71.81 / 99.53	82.89 / 99.73	85.71 / 0.000

H Error Bar

We conduct random experiments on all grayscale and RGB datasets for 5 trials using the trainable methods (ELBO, PHP, and AVOID methods). The average error rates are presented in Table H, and it can be observed that the error rates are similar across these methods.

Datasets	Grayscale datasets			RGB datasets		
Method	ELBO	PHP	AVOID	ELBO	PHP	AVOID
Avg. error	± 0.788	± 0.512	± 0.613	± 1.408	± 1.579	± 1.649

I Broader Impact

The impact of our research can be outlined in two key aspects:

- For Unsupervised OOD Detection: Our approach stands out due to its broad applicability and versatility. Unlike many conventional methods, it does not require labeled data and it can be applied to model the distribution of diverse data types using deep generative models. This is particularly useful in applications where labeled data is scarce or unavailable. Additionally, our method provides a universal solution to enhance OOD detection performance. This is achieved by offering an innovative perspective on the *overestimation* issue in VAE, which is not predicated on the data type.
- For the development of deep generative models: Our research offers valuable insights for the progression of deep generative models. By employing the KL divergence, $D_{KL}(q(z)||p(z))$, our method can provide verification of whether a generative model has adequately learned to model the data distribution. These insights could potentially spark new developments and inspire more representative generative models, thereby furthering the field of deep learning research and applications.

In conclusion, our research holds promising potential to provide substantial contributions to both the realm of unsupervised OOD detection and the development of deep generative models.

J Limitation

The primary limitation of this paper lies in the simplicity of the developed methods, which could potentially under-explore the full capabilities of our method on unsupervised OOD detection. This

753 design choice is deliberated to allow readers to focus more on our central theme: the analysis of the
754 *overestimation* issue in VAE. We aim to create methods as straightforward as possible, with the express
755 purpose of verifying the analyzed factors, while introducing as few additional hyperparameters as
756 possible. This approach is intended to provide readers with greater insight into our analysis, as well
757 as inspire them to develop their own advanced methods based on our analysis. We provide examples
758 and helpful insights into our work through Figures 4 and 5. While acknowledging the aforementioned
759 limitation, we posit that our analysis can pave the way for the creation of more advanced methods.
760 These enhanced approaches can potentially lead to further improvements in the performance of
761 unsupervised OOD detection. Our work, therefore, should be seen as a stepping stone towards more
762 sophisticated applications in this field.