Autoencoding Under Normalization Constraints

Sangwoong Yoon [1] **Yung-Kyun Noh** [2 3] **Frank C. Park** [1 4]

Abstract

Likelihood is a standard estimate for outlier detection. The specific role of the normalization constraint is to ensure that the out-of-distribution (OOD) regime has a small likelihood when samples are learned using maximum likelihood. Because autoencoders do not possess such a process of normalization, they often fail to recognize outliers even when they are obviously OOD. We propose the Normalized Autoencoder (NAE), a normalized probabilistic model constructed from an autoencoder. The probability density of NAE is defined using the reconstruction error of an autoencoder, which is differently defined in the conventional energy-based model. In our model, normalization is enforced by suppressing the reconstruction of negative samples, significantly improving the outlier detection performance. Our experimental results confirm the efficacy of NAE, both in detecting outliers and in generating in-distribution samples.

1. Introduction

An autoencoder (Rumelhart et al., 1986) is a neural network trained to reconstruct samples from a training data distribution. Since in principle the quality of reconstruction is expected to be poor for inputs that deviate significantly from the training data, autoencoders are widely used in outlier detection (Japkowicz et al., 1995), in which an input with a large reconstruction error is classified as out-of-distribution (OOD). Autoencoders for outlier detection have been applied in domains ranging from video surveillance (Zhao et al., 2017) to medical diagnosis (Lu & Xu, 2018).

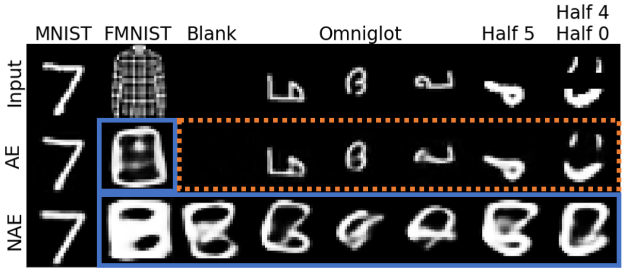

Figure 1. Examples of reconstructed outliers. The last two rows show the reconstructions from a conventional autoencoder (AE) and NAE. Both autoencoders are trained on MNIST, and other inputs are outliers. The architecture of the two autoencoders is identical. Successful detection of an outlier is highlighted with blue solid rectangles, while detection failures due to the reconstruction of outliers are denoted with an orange dotted rectangle. Note that AE is not the identity mapping, as it fails to reconstruct the shirt.

However, autoencoders have been known to reconstruct outliers consistently across a wide range of experimental settings (Lyudchik, 2016; Tong et al., 2019; Zong et al., 2018; Gong et al., 2019). We name this phenomenon *outlier reconstruction*. Figure 1 shows examples of some outliers reconstructed by an autoencoder trained with MNIST data; the autoencoder is able to reconstruct a wide range of OOD inputs, including constant black pixels, Omniglot characters, and fragments of MNIST digits. The early works on regularized autoencoders (Vincent et al., 2008; Rifai et al., 2011; Ng et al., 2011) focus for the most part on preventing the autoencoder from turning into the identity mapping that reconstructs every input. Nonetheless, outlier reconstruction can still occur even when the autoencoder is not the identity as shown by the non-identity autoencoder in Figure 1. Not surprisingly, outlier reconstruction is a leading cause of autoencoder's detection failure.

On the other hand, in a normalized probabilistic model, it is known that maximum likelihood learning suppresses the assignment of probability mass in OOD regions in order to keep the model normalized. Thus, the likelihood is widely used as a predictor for outlier detection (Bishop, 1994). Meanwhile, an autoencoder is not a probabilistic model of the data and does not have a suppression mechanism corresponding to the normalization in other probabilistic models. As a result, the reconstruction of outliers are not inhibited during training of an autoencoder.

This paper formulates an autoencoder as a normalized prob-

[1] Department of Mechanical Engineering, Seoul National University, Seoul, Republic of Korea [2] Department of Computer Science, Hanyang University, Seoul, Republic of Korea [3] Korea Institute of Advanced Studies, Seoul, Republic of Korea [4] Saige Research, Seoul, Republic of Korea. Correspondence to: Yung-Kyun Noh <nohyung@hanyang.ac.kr>, Frank C. Park <fcp@snu.ac.kr>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

abilistic model to introduce a mechanism for preventing outlier reconstruction. In our formulation, which we call the **Normalized Autoencoder (NAE)**, the reconstruction error is re-interpreted as an energy function, i.e., the unnormalized negative log-density, and defines a probabilistic model from an autoencoder. During maximum likelihood learning of NAE, outlier reconstruction is naturally suppressed by enforcing the normalization constraint, and the resulting autoencoder is significantly less prone to reconstruct outliers, as shown in Figure 1.

In each training iteration of NAE, samples generated from the model is used to update the normalization constraint which is implicitly computed as in other energy-based models. Since running a Markov Chain Monte Carlo (MCMC) sampler until convergence every iteration is computationally infeasible, an approximate sampling strategy has to be employed. We observe that training with popular sampling strategies such as Contrastive Divergence (CD; Hinton (2002)) and Persistent CD (PCD; Tieleman (2008)) may often produce poor density estimates. Instead, we propose **on-manifold initialization (OMI)**, a method of initializing an MCMC chain on manifold defined by the decoder of an autoencoder. OMI selects high-model-density initial states by leveraging the assumption that points on the decoder manifold typically have small reconstruction error, i.e. high model density. With OMI, NAE can accurately recover the data density and thus become an effective outlier detector.

Intriguingly, although technically a normalized probabilistic model, the variational autoencoder (VAE; Kingma & Welling (2014)) also reconstructs outliers and assigns a spuriously high likelihood on OOD data (Nalisnick et al., 2019; Xiao et al., 2020) for reasons that are as-yet unclear.

Our main contributions can be summarized as follows:

- We propose NAE, a novel generative model constructed from an autoencoder;
- We propose OMI, a sampling strategy tailored for NAE;
- We empirically show that NAE is highly effective for outlier detection and can perform other generative tasks.

Section 2 provides brief background on autoencoders and energy-based models. NAE is described in Section 3, and OMI is described in Section 4. Related works are reviewed in Section 5. Section 6 presents experimental results. Section 7 provide discussions and conclude the paper. Our source code and pre-trained models are publicly available online at `https://github.com/swyoon/normalized-autoencoders`.

2. Background

2.1. Autoencoders

Autoencoders are neural networks trained to reconstruct an input datum $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^{D_\mathbf{x}}$. For an input $\mathbf{x}$, the quality of its reconstruction is measured in reconstruction error $l_\theta(\mathbf{x})$, where θ denotes parameters in an autoencoder. The loss function of an autoencoder L_{AE} for training is the expected reconstruction error of training data. Gradient descent training is performed via computing the gradient of L with respect to model parameters θ:

$$L_{\text{AE}} = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[l_\theta(\mathbf{x})], \tag{1}$$

$$\nabla_\theta L_{\text{AE}} = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\nabla_\theta l_\theta(\mathbf{x})], \tag{2}$$

where ∇_θ is the gradient operator with respect to θ and $p(\mathbf{x})$ denotes the data density.

Architecture An autoencoder consists of two submodules, an encoder and a decoder. An encoder $f_e(\mathbf{x}) : \mathbb{R}^{D_\mathbf{x}} \rightarrow \mathbb{R}^{D_\mathbf{z}}$ maps an input $\mathbf{x}$ to a corresponding latent representation vector $\mathbf{z} \in \mathcal{Z} \subset \mathbb{R}^{D_\mathbf{z}}$, and a decoder $f_d(\mathbf{z}) : \mathbb{R}^{D_\mathbf{z}} \rightarrow \mathbb{R}^{D_\mathbf{x}}$ maps a latent vector $\mathbf{z}$ back to the input space. Then, the reconstruction error $l_\theta(\mathbf{x})$ is given as:

$$l_\theta(\mathbf{x}) = \text{dist}(\mathbf{x}, f_d(f_e(\mathbf{x}))), \tag{3}$$

where $\text{dist}(\cdot,\cdot)$ is a distance-like function measuring the deviation between an input $\mathbf{x}$ and a reconstruction $f_d(f_e(\mathbf{x}))$. A typical choice is the squared L^2 distance, i.e., $\text{dist}(\mathbf{x}_1,\mathbf{x}_2) = ||\mathbf{x}_1 - \mathbf{x}_2||_2^2$. Other possible choices include L^1 distance, $\text{dist}(\mathbf{x}_1,\mathbf{x}_2) = |\mathbf{x}_1 - \mathbf{x}_2|$, and the structural similarity (SSIM; Wang et al. (2004); Bergmann et al. (2018)).

Note that the reconstruction error (Eq. (3)) is *not* a likelihood of a datum, and therefore the minimization of the reconstruction error does not correspond to the maximization of the likelihood. Without modification, an autoencoder per se is not a probabilistic model.

Outlier Detection A datum is an outlier or called OOD if it lies in the ρ-sublevel set of a data density $\{\mathbf{x}|p(\mathbf{x}) \leq \rho\}$ (Steinwart et al., 2005). We particularly focus on $\rho = 0$, where an outlier is defined as an input from the outside of the data distribution's support. Most of the OOD examples which attract the attention of the research community are in fact out-of-support samples. For example, SVHN and CIFAR-10 are out-of-support to each other, as confirmed by a supervised classifier perfectly discriminating the two datasets. Note that the support-based definition provides invariant characterization of outliers, as no invertible transform defined on the data space alters whether a sample is in- or out-of-support. Meanwhile, for $\rho \neq 0$, the characterization of outliers are not invariant to the choice of coordinates Lan & Dinh (2020).

In the autoencoder-based outlier detection (Japkowicz et al., 1995), an input is classified as OOD if its reconstruction error $l_\theta(\mathbf{x})$ is greater than a threshold τ: $l_\theta(\mathbf{x}) > \tau$. The outlier reconstruction indicates that there exists an input $\mathbf{x}^*$ with $p(\mathbf{x}^*) \leq \rho$, but $l_\theta(\mathbf{x}^*) < \tau$. Appendix includes the detailed investigation on outlier reconstruction.

2.2. Energy-based Models

Unlike autoencoders, energy-based models (EBMs) are valid models for a normalized probability distribution. The EBM represents a probability distribution through the unnormalized negative log probability, also called the energy function $E_\theta(\mathbf{x})$. Here, θ denotes the model parameters.

For a continuous input $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^{D_\mathbf{x}}$, $E_\theta(\mathbf{x})$ defines the model density function $p_\theta(\mathbf{x})$ through Gibbs distribution:

$$p_\theta(\mathbf{x}) = \frac{1}{\Omega_\theta} \exp(-E_\theta(\mathbf{x})/T), \tag{4}$$

where $T \in \mathbb{R}^+$ is called the temperature and is often ignored by setting $T = 1$. Ω_θ is the normalization constant and is defined as:

$$\Omega_\theta = \int_\mathcal{X} \exp(-E_\theta(\mathbf{x})/T) \mathrm{d}\mathbf{x} < \infty. \tag{5}$$

The computation of Ω_θ is usually difficult for high-dimensional $\mathbf{x}$. However, maximum likelihood learning can still be performed without the explicit evaluation of Ω_θ. The gradient of negative log likelihood of data is given as follows (Younes, 1999):

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[-\nabla_\theta \log p_\theta(\mathbf{x})] \\
=&\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\nabla_\theta E_\theta(\mathbf{x})]/T + \nabla_\theta \log \Omega_\theta \qquad (6)\\
=&\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\nabla_\theta E_\theta(\mathbf{x})]/T - \mathbb{E}_{\mathbf{x}'\sim p_\theta(\mathbf{x})}[\nabla_\theta E_\theta(\mathbf{x}')]/T \qquad (7)
\end{aligned}$$

$\nabla_\theta \log \Omega_\theta$ in Eq. (6) is evaluated from the energy gradients of samples $\mathbf{x}'$ generated from the model in Eq. (7). The samples from $p_\theta(\mathbf{x})$ are often called ”negative” samples. The derivation of Eq. (7) is provided in Appendix.

In Eq. (7), the first term decreases the energy of the training data, or “positive” samples, while the second term increases the energy of the generated samples, or “negative” samples. The training converges when $p_\theta(\mathbf{x})$ becomes identical to $p(\mathbf{x})$, as the two gradient terms cancel out. In practice, the two expectations in Eq. (7) are approximated with a mini-batch of samples during each iteration. Figure 2 visualizes the gradients in Eq. (7).

Langevin Monte Carlo (LMC) The negative samples are generated using MCMC. LMC (Parisi (1981); Grenander & Miller (1994)) is a simple yet effective MCMC method used in recent work on deep EBMs (Du & Mordatch, 2019; Grathwohl et al., 2020; Nijkamp et al., 2019). In LMC, a starting point $\mathbf{x}_0$ is drawn from a noise distribution $p_0(\mathbf{x})$, typically a Gaussian or uniform distribution. Starting from $\mathbf{x}_0$, a Markov chain evolves as follows:

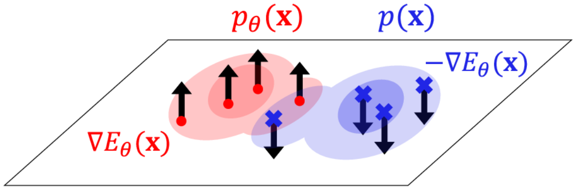

Figure 2. An illustration of the energy gradients in Eq. (7). The red and blue shades represent the model and the data density, respectively. The gradient update following Eq. (7) increases the energy of samples from $p_\theta(\mathbf{x})$ (the red dots) and decreases the energy of training data (the blue crosses).

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \lambda_\mathbf{x} \nabla_\mathbf{x} \log p_\theta(\mathbf{x}_t) + \sigma_\mathbf{x} \epsilon_t, \tag{8}$$

where $\epsilon_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. $\lambda_\mathbf{x}$ and $\sigma_\mathbf{x}$ are the step size and the noise parameters, respectively. A theoretically motivated choice is $2\lambda_\mathbf{x} = \sigma_\mathbf{x}^2$, but the parameters are often tweaked separately for better performance (Du & Mordatch, 2019; Grathwohl et al., 2020; Nijkamp et al., 2019). As $\nabla_\mathbf{x} \log p_\theta(\mathbf{x}) = -\nabla_\mathbf{x} E(\mathbf{x})/T$, tweaking the step size can be seen as adjusting the temperature T.

To ensure the convergence of the chain, either Metropolis-Hastings rejection (Roberts et al., 1996) or annealing of the noise parameter to zero (Welling & Teh, 2011) may be employed, but often omitted in practice.

We discuss specific strategies to evaluate the second term in Eq. (7) in Section 4. For a comprehensive review on various strategies for training an EBM, readers may refer to Song & Kingma (2021).

3. Normalized Autoencoders

3.1. Definition

We propose **Normalized Autoencoder (NAE)**, a normalized probabilistic model defined from an autoencoder. The probability density of NAE $p_\theta(\mathbf{x})$ is defined as a Gibbs distribution (Eq. (4)) the energy of which is defined as the reconstruction error of an autoencoder:

$$E_\theta(\mathbf{x}) = l_\theta(\mathbf{x}). \tag{9}$$

Thus, the model density of NAE is given as

$$p_\theta(\mathbf{x}) = \frac{1}{\Omega_\theta} \exp(-l_\theta(\mathbf{x})/T), \tag{10}$$

where Ω_θ is defined as in Eq. (5). Due to the normalization constant, $p_\theta(\mathbf{x})$ is a properly normalized probability density.

As a probabilistic model, NAE is trained to maximize the likelihood of data. The loss function to be minimized is the negative log-likelihood of data:

$$\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[-\log p_\theta(\mathbf{x})] = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[l_\theta(\mathbf{x})]/T + \log \Omega_\theta. \tag{11}$$

The gradient for the negative log-likelihood is evaluated as in conventional EBMs (Eq. (7)).

$$\begin{aligned}&\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[-\nabla_\theta \log p_\theta(\mathbf{x})]\\ &= \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\nabla_\theta l_\theta(\mathbf{x})]/T - \mathbb{E}_{\mathbf{x}'\sim p_\theta(\mathbf{x})}[\nabla_\theta l(\mathbf{x}')]/T. \end{aligned} \tag{12}$$

Therefore, each gradient step decreases the reconstruction error of training data $\mathbf{x}$, while increasing the reconstruction error of negative samples $\mathbf{x}'$ generated from $p_\theta(\mathbf{x})$.

3.2. Remarks

Normalization as Regularization In NAE, enforcement of normalization can be viewed as a regularizer for the reconstruction loss (1). A typical formulation for a regularized autoencoder is given as $L = L_{\text{AE}} + L_{\text{reg}}$, where L_{reg} is a regularizer. By setting the loss function of NAE as $L = T\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[-\log p_\theta(\mathbf{x})]$, we have $L = L_{\text{AE}} + T\log\Omega_\theta$. Therefore, the normalization constant contributes as a regularizer: $L_{\text{reg}} = T\log\Omega_\theta$.

Suppression of Outlier Reconstruction During the training of NAE, the reconstruction of an outlier is inhibited by enforcing the normalization constraint. Given a successful sampling process, the negative samples should cover all high density regions of $p_\theta(\mathbf{x})$. A sample from a high density region of $p_\theta(\mathbf{x})$ has a low $l_\theta(\mathbf{x})$ by definition (Eq. (9)). Hence, if there exist a reconstructable outlier, which has high $p_\theta(\mathbf{x})$ due to low $l_\theta(\mathbf{x})$, it will appear as a negative sample from MCMC. As the gradient update given in Eq. (12) increases the reconstruction error of negative samples, the reconstruction quality of a reconstructable outlier will be degraded. As a result, the reconstruction error of NAE becomes a more informative predictor that discriminates outliers from inliers than that of a conventional autoencoder.

Outlier Detection with Likelihood NAE bridges the two popular outlier detection criteria, namely, the reconstruction error (Japkowicz et al., 1995) and the likelihood (Bishop, 1994). The reconstruction error criterion classifies an input with a large reconstruction error as OOD $l_\theta(\mathbf{x}) > \tau$, whereas the likelihood criterion predicts an input as an outlier if the log-likelihood is smaller than the threshold $\log p_\theta(\mathbf{x}) < \tau'$. These two criteria are equivalent in NAE for appropriately set τ and τ', as the reconstruction error and the log-likelihood has a linear relationship: $\log p_\theta(\mathbf{x}) = -l_\theta(\mathbf{x}) - \log\Omega_\theta$. Note that the two criteria rarely coincide in other models, for example, denoising autoencoders (DAE, Vincent et al. (2008)), VAE (Kingma & Welling, 2014)), and DSEBMs (Zhao et al., 2016), causing confusion on which of the decision rules should be employed for outlier detection.

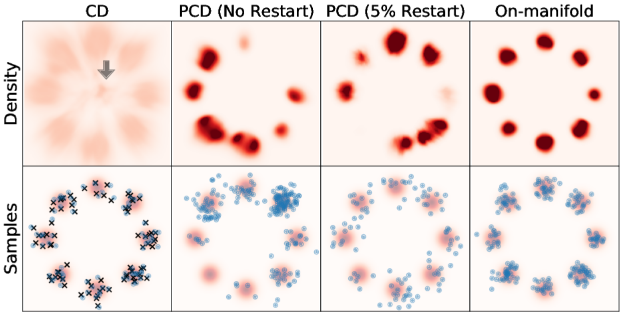

Figure 3. Density estimates and negative samples from NAEs trained by various approximate sampling methods. The generated samples (blue dots) are visualized along with the true density, a 2D mixture of 8 Gaussians. The data density is depicted in Figure 5. **CD**: The learned density has a spurious mode, marked by an arrow. The black crosses denote training data. **PCD without restart**: The highly correlated samples result in an oscillating density estimate. **PCD with restart**: Despite the good quality of sampling, the density is poorly estimated. **On-manifold**: Both density estimation and sample generation are performed well. More details are specified in Section 4.1 and Section 6.2.

Sample Generation Samples from $p_\theta(\mathbf{x})$ are generated through MCMC. Unlike VAE, the forward pass of a decoder should not be considered as sample generation.

4. On-Manifold Initialization

The main challenge in the training of NAE through Eq. (12) is that each iteration requires negative sample generation using MCMC, which is computationally expensive. In this section, we first discuss the failure modes of popular approximate sampling strategies for EBMs, namely Contrastive Divergence (CD; Hinton (2002)) and Persistent CD (PCD; Tieleman (2008)). We argue that the method on how the initial state of MCMC is chosen have incurred such failure modes. Then, we propose on-manifold initialization, an approximate sampling strategy effective in training the NAE. On-manifold initialization provides a better initial state for MCMC by leveraging the structure of an autoencoder.

There exist other training methods for EBMs which do not rely on MCMC, for example denoising score matching (Vincent, 2011) or noise contrastive estimation (Gutmann & Hyvärinen, 2010), and they may also be applicable to NAE. We leave application of such methods on NAE as future work.

4.1. Failure Modes of CD and PCD

Failure Mode of CD CD, often called CD-k, draws a negative sample by first initializing a Markov chain of MCMC at a training data point, then proceeding k steps of MCMC transitions. The strength of CD is that the number of steps k can be radically smaller, e.g., $k = 1$, than the usual number of steps required in a convergent MCMC run, significantly reducing the amount of computation.

However, when k is small, CD-k is not able to suppress a spuriously high mode in the model density $p_\theta(\mathbf{x})$ located far from the data distribution $p(\mathbf{x})$, because negative samples are only generated in the vicinity of training data. Figure 3 shows an instance of a spurious mode in the model density. Negative samples (blue dots) are close to training data (black crosses) so that they do not reach for the density mode in the middle. As a result, the mode is not suppressed. Such a spurious mode will result in outlier detection failures and, in case of NAE, reconstructed outliers. The possibility of accidentally assigning high density in the unvisited area was acknowledged in the original article (Section 3 of Hinton (2002)). Spurious modes are also observed in DAE, where a corrupted datum is located only in the neighborhood of a training data point (Alain & Bengio, 2014). Increasing k will decrease the chance of have spurious modes, but the computational advantage of CD will be lost when k is large.

Failure Mode of PCD An initial state of MCMC in PCD is given as the negative sample generated from MCMC in the previous training iteration. PCD was originally implemented using fully persistent MCMC (Tieleman, 2008). However, without a restart, MCMC chains in a mini-batch may become highly correlated to each other. When $p_\theta(\mathbf{x})$ is multi-modal, the correlated chains yield degenerate negative samples which only cover a subset of density modes as in Figure 3. The degenerate samples make the density estimate oscillatory, slowing the convergence of the model.

The degeneracy between chains can be mitigated by randomly resetting the initial state to a sample from the noise distribution $p_0(\mathbf{x})$ with a small probability (typically 5%) (Du & Mordatch, 2019; Grathwohl et al., 2020). However, learning with PCD still fails to yield an accurate density estimate (Figure 3). This failure mode can be explained by the study of Nijkamp et al. (2019): When a short MCMC chain initialized from $p_0(\mathbf{x})$ is used in training, an EBM simply learns a flow that maps $p_0(\mathbf{x})$ to $p(\mathbf{x})$, and the energy no longer models the data density. Using a restart drives an EBM to become such a flow, as restarted chains are short and start from $p_0(\mathbf{x})$.

In summary, CD initializes MCMC from the data distribution $p_\theta(\mathbf{x})$, and PCD initializes MCMC from a noise distribution $p_0(\mathbf{x})$. The convergence of MCMC is independent of its initialization in theory, but the initialization method can be crucial in practice, as shown in Figure 3. When $p_\theta(\mathbf{x})$, from which we want to sample, deviates significantly from $p_\theta(\mathbf{x})$ or $p_0(\mathbf{x})$, these initialization methods may lead to a poor density estimate and a suboptimal performance in outlier detection.

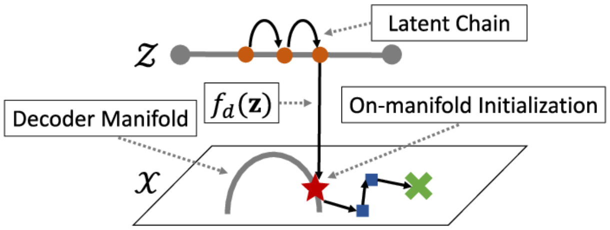

Figure 4. An illustration of the on-manifold initialization. The one-dimensional latent space $\mathcal{Z}$ and the two-dimensional input space $\mathcal{X}$ are shown. The red star is the on-manifold initialized state. The cross denotes a negative sample obtained at the end of the whole process.

4.2. On-manifold Initialization

We propose **on-manifold initialization (OMI)**, a novel MCMC initialization strategy which eventually leads to a significantly better density estimate. We aim to initialize a MCMC chain from a high-density region of $p_\theta(\mathbf{x})$ instead of $p_0(\mathbf{x})$ or $p(\mathbf{x})$. While finding a high-density region given an energy function is difficult in general, it is possible for NAE's distribution, since we can exploit the structure of an autoencoder. For a sufficiently well-trained autoencoder, a point with high $p_\theta(\mathbf{x})$, i.e., a small reconstruction error, will lie near the *decoder manifold*, which we define as:

$$\mathcal{M} = \{\mathbf{x} | \mathbf{x} = f_d(\mathbf{z}), \mathbf{z} \in \mathcal{Z}\}. \tag{13}$$

In on-manifold initialization, we initialize MCMC from a point in the decoder manifold $\mathbf{x}_0 \in \mathcal{M}$.

Not all points in $\mathcal{M}$ have high $p_\theta(\mathbf{x})$. To find points with high $p_\theta(\mathbf{x})$, we run a preliminary MCMC named as *latent chain* in the latent space $\mathcal{Z}$. The latent chain generates a sample from *on-manifold density* $q_\theta(\mathbf{z})$ defined from *on-manifold energy* $H_\theta(\mathbf{z})$.

$$\begin{aligned} q_\theta(\mathbf{z}) =& \frac{1}{\Psi_\theta} \exp(-H_\theta(\mathbf{z})/T_\mathbf{z}), & (14) \\ H_\theta(\mathbf{z}) =& E_\theta(f_d(\mathbf{z})), & (15) \end{aligned}$$

where $\Psi_\theta = \int \exp(-H_\theta(\mathbf{z})/T_\mathbf{z}) d\mathbf{z}$ is the normalization constant and $T_\mathbf{z}$ is the temperature. A latent vector $\mathbf{x}$ with a small $H_\theta(\mathbf{z})$ will result in a small $E_\theta(\mathbf{x})$ when it is mapped to the input space by $\mathbf{x} = f_d(\mathbf{z})$. Thus, $H_\theta(\mathbf{z})$ guides the latent chain to find $\mathbf{z}$ which produce $\mathbf{x}_0 \in \mathcal{M}$ which has a small energy, i.e., a small reconstruction error.

Similarly to Eq. (8), we use LMC to run the latent chain. An initial state $\mathbf{z}_0$ is drawn from a noise distribution defined on the latent space. Then the state propagates as:

$$\mathbf{z}_{t+1} = \mathbf{z}_t + \lambda_\mathbf{z} \nabla_\mathbf{z} \log q_\theta(\mathbf{z}_t) + \sigma_\mathbf{z} \epsilon_t, \tag{16}$$

where $\lambda_\mathbf{z}$ and $\sigma_\mathbf{z}$ are the step size and the noise parameters as in Eq. (8). A sample replay buffer (Du & Mordatch,

2019) is applicable in the latent chain. Figure 4 illustrates negative sample generation process using the on-manifold initialization. We also write the process as an algorithm in Appendix.

5. Related Work

Autoencoders There have been several attempts to formulate a probabilistic model from an autoencoder. VAE uses a latent variable model by introducing a prior distribution $p(\mathbf{z})$. However, the prior may deviate from the actual distribution of data in $\mathcal{Z}$, which may cause problems. GPND (Pidhorskyi et al., 2018) models probability density by factorizing into on- and off-manifold components but still requires a prior distribution. $\mathcal{M}$-flow (Brehmer & Cranmer, 2020) only defines a probability density on the decoder manifold and does not assign a likelihood to off-manifold data. DAE models a density by learning the gradient of log-density (Alain & Bengio, 2014).

MemAE (Gong et al., 2019) is a rare example that directly tackles the outlier reconstruction problem. MemAE employs a memory module that memorizes training data to prevent outlier reconstruction, but in this case, the reconstruction error for an inlier can be large because the model's generalization ability is also limited.

Design of Energy Functions Specifying the class of $E_\theta(\mathbf{x})$ not only has computational consequences but alters the inductive bias that an EBM encodes. Feed-forward convolutional networks are used in Xie et al. (2016), Du & Mordatch (2019) and Grathwohl et al. (2020) and are shown to effectively model the distribution of images. The energy can also be modeled in an auto-regressive manner (Nash & Durkan, 2019; Meng et al., 2020). Auto-regressive energy functions are very flexible and thus are capable of modeling high-frequency patterns in data. VAEBM (Xiao et al., 2021) combines VAE and a feed-forward EBM to model complicated data distribution.

The reconstruction error of an autoencoder is used as a discriminator in EBGAN (Zhao et al., 2016). Although the reconstruction error was called "energy" in EBGAN, the formulation is clearly different from NAE. EBGAN does not utilize Gibbs distribution formulation (Eq. (4)) to model a distribution, and samples are generated from a separate generator network. In DSEBM (Zhao et al., 2016), the difference between an input and its reconstruction is interpreted as the gradient of log-density.

6. Experiments

6.1. Technicalities for NAE Training

Pre-training as a Conventional Autoencoder NAE can be pre-trained as a conventional autoencoder by minimizing the reconstruction error following Eq. (2), before the main training. By providing a good initialization for network weights and the decoder manifold, pre-training greatly reduces the number of NAE training iterations (Eq. (12)) required until convergence. Pre-training is not always necessary: In our experiments, we observe that NAE can be trained successfully without pre-training for synthetic data. However, pre-training was essential to obtain decent results for larger scale data, such as MNIST and CIFAR-10.

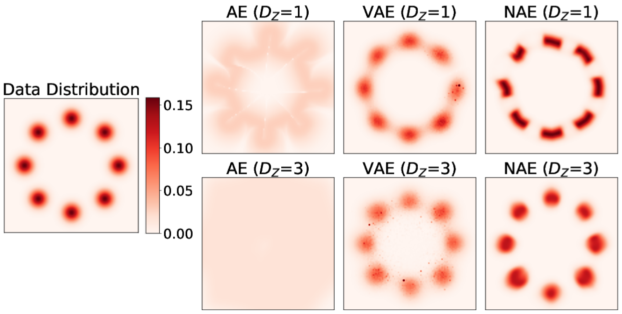

Figure 5. Estimating 8 Gaussians using various autoencoders. The density of an autoencoder (AE) is computed from Eq. (10). AE gives a significant amount of probability to low-data-density area. VAE also assigns some probability mass in between Gaussians. Meanwhile, the density estimate from NAE agrees well with the data distribution.

Latent Space Structure Two configurations for the latent space is used in experiments: the unbounded real space $\mathbb{R}^{D_\mathbf{z}}$ and the surface of a hypersphere $\mathbb{S}^{D_\mathbf{z}-1}$. When $\mathcal{Z} = \mathbb{R}^{D_\mathbf{z}}$, a linear layer is used as the output of an encoder. $q_0(\mathbf{z})$ is set as $\mathcal{N}(\mathbf{0}, \mathbf{I})$. The squared norm of the latent vectors are added to the loss function as a regularizer so that $\mathbf{z}$'s concentrate near the origin (Ghosh et al., 2020).

For the hyperspherical space $\mathcal{Z} = \mathbb{S}^{D_\mathbf{z}-1}$ (Davidson et al., 2018; Xu & Durrett, 2018; Zhao et al., 2019), the output of an encoder is projected to the surface of a unit ball through the division by its norm: $\mathbf{z} \leftarrow \mathbf{z}/||\mathbf{z}||$. In Langevin dynamics, a sample is projected to $\mathbb{S}^{D_\mathbf{z}-1}$ at the end of each step. $q_0(\mathbf{z})$ is set to a uniform distribution on $\mathbb{S}^{D_\mathbf{z}-1}$.

The hyperspherical latent space has a few advantages over $\mathbb{R}^{D_\mathbf{z}}$. First, it is impossible to draw uniformly random samples, because $\mathbb{R}^{D_\mathbf{z}}$ is not compact. Second, for large $D_\mathbf{z}$, it is difficult to draw samples near the origin, because of its exponentially decreasing volume. However, we believe more works needs to be done to completely understand the effect of hyperspherical geometry on the latent representation.

Regularizing Negative Sample Energy As introduced in Du & Mordatch (2019), we regularize the energy of negative samples to prevent its divergence. We add the average squared energy of negative samples in a mini-batch to the loss function: $L = L_{\text{NAE}} + \alpha \sum_{i=1}^{B} E(\mathbf{x}'_i)^2/B$ for the batch size B and the hyperparameter α. We set $\alpha = 1$.

Table 1. MNIST hold-out class detection AUC scores. The values in parentheses denote the standard error of mean after 10 training runs.

HOLD-OUT:	0	1	2	3	4	5	6	7	8	9	AVG
NAE-OMI	**.989**(.002)	**.919**(.013)	**.992**(.001)	**.949**(.004)	**.949**(.005)	**.978**(.003)	**.938**(.004)	**.975**(.024)	**.929**(.004)	**.934**(.005)	**.955**
NAE-CD	.799	.098	.878	.769	.656	.806	.874	.537	.876	.500	.679
NAE-PCD	.745	.114	.879	.754	.690	.813	.872	.509	.902	.544	.682
AE	.819	.131	.843	.734	.661	.755	.844	.542	.902	.537	.677
DAE	.769	.124	.872	.935	.884	.793	.865	.533	.910	.625	.731
VAE(R)	.954	.391	.978	.910	.860	.939	.916	.774	.946	.721	.839
VAE(L)	.967	.326	.976	.906	.798	.927	.928	.751	.935	.614	.813
WAE	.817	.145	.975	**.950**	.751	.942	.853	.912	.907	.799	.805
GLOW	.803	.014	.624	.625	.364	.561	.583	.326	.721	.426	.505
PXCNN++	.757	.030	.663	.663	.483	.642	.596	.307	.810	.497	.545
IGEBM	.926	.401	.642	.644	.664	.752	.851	.572	.747	.522	.672
DAGMM	.386	.304	.407	.435	.444	.429	.446	.349	.609	.420	.423

6.2. 2D Density Estimation

We demonstrate the density estimation capability of NAE with a two-dimensional mixture of 8 Gaussians. First, we benchmark negative sample generation strategies for NAE, including CD, PCD with and without restart, and on-manifold initialization. The results are shown in Figure 3 and discussed in Section 4.1 in detail.

Second, we compare NAE trained with the on-manifold initialization to a conventional autoencoder and VAE (Figure 5). An autoencoder assigns high densities on regions between Gaussian modes, meaning that an autoencoder gives a small reconstruction error from a points from the region. For the overcomplete case ($D_{\mathbf{z}} = 3 > D_{\mathbf{x}}$), an autoencoder almost becomes the identity map, and its reconstruction error is not an informative predictor for an outlier. VAE and NAE learn a non-identity function under the overcomplete setting, showing the effectiveness of their regularizers.

In the experiments, the identical network architecture is used, and the temperature is optimized by gradient descent. In on-manifold initialization, temperature values are shared by the main MCMC and the latent chain. When performing MCMC in $\mathcal{X}$, Metropolis-Hastings rejection is applied to ensure the detailed balance but is not applied in the latent chain. For visualization, the normalization constants for an autoencoder and NAE are computed by numerically integrating over the domain, $[-4, 4]^2$.

6.3. Outlier Detection

Experimental Setting We empirically demonstrate the effectiveness of NAE as an outlier detector. In outlier detection tasks, an outlier detector is trained only using inlier data and then asked to discriminate outliers from inliers during test phase. Given an input, a detector is assumed to produce a scalar decision function which indicates the outlierness of the input. We measure the detection performance in AUC, i.e., the area under the receiver operating characteristic curve. Following the protocol of Ren et al. (2019) and Hendrycks et al. (2019), we use an OOD dataset different from the datasets used in test phase to tune model hyperpamraeters. Additional details on model implementation and datasets can be found in the supplementary material.

The identical networks architectures are used for all autoencoder-based methods. The reconstruction error is used as the decision function, except for VAE. For deep generative models, PixelCNN++ (PXCNN++, Salimans et al. (2017)), Glow (Kingma & Dhariwal, 2018) and a feed-forward EBM (IGEBM, Du & Mordatch (2019)), we use the negative log-likelihood (i.e., the energy) as the decision function. For VAE, we show two results from using the reconstruction error (R) or the negative log-likelihood (L) as decision functions.

MNIST Hold-Out Class Detection One class from MNIST is set as the outlier class and the rest as the inlier class. Then, the procedure is repeated for all ten classes in MNIST. ConstantGray dataset is used for model selection.

This problem is not as easy as it seems, as confirmed in the very low performance of various algorithms in Table 1. When a class is held out from MNIST, the remaining 9 classes may contain a set of visual features sufficient to reconstruct the hold-out class, i.e., the outlier reconstruction occurs. The outlier reconstruction is particularly severe for the digit 1, 4, 7 and 9, possibly because their shape can be reconstructed from the recombination of other digits. For example, overlapping 4 and 7 produces a shape similar to 9. Interestingly, most of the other baseline algorithms also show poor performance when 1, 4, 7 or 9 are held out as the outlier. NAE shows the highest AUC score for all classes and effectively suppresses the reconstruction of the outlier class (Figure 6).

We also compare CD and PCD along with OMI in training NAEs. Using CD and PCD show poor outlier detection performance, although given the identical set of MCMC parameters.

Out-of-Distribution Detection The samples from different datasets are used as the outlier class. We test two inlier datasets, CIFAR-10 or ImageNet 32×32 (ImageNet32).

Table 2. OOD detection performance in AUC.

In: CIFAR-10	ConstantGray	FMNIST	SVHN	CelebA	Noise
NAE	**.963**	**.819**	**.920**	**.887**	1.0
AE	.006	.650	.175	.655	1.0
DAE	.001	.671	.175	.669	1.0
VAE(R)	.002	.700	.191	.662	1.0
VAE(L)	.002	.767	.185	.684	1.0
WAE	.000	.649	.168	.652	1.0
GLOW	.384	.222	.260	.419	1.0
PXCNN++	.000	.013	.074	.639	1.0
IGEBM	.192	.216	.371	.477	1.0

In: ImageNet32	ConstantGray	FMNIST	SVHN	CelebA	Noise
NAE	**.966**	**.994**	**.985**	**.949**	1.0
AE	.005	.915	.102	.325	1.0
DAE	.069	.991	.102	.426	1.0
VAE(R)	.030	.936	.132	.501	1.0
VAE(L)	.028	.950	.132	.545	1.0
WAE	.069	.991	.081	.364	1.0
GLOW	.413	.856	.169	.479	1.0
PXCNN++	.000	.004	.027	.238	1.0

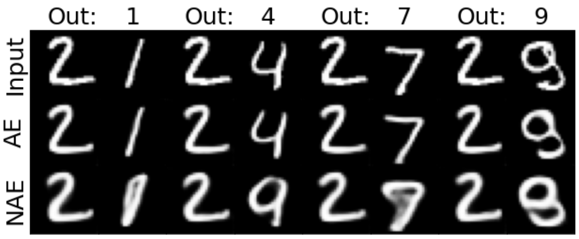

Figure 6. Reconstruction examples in MNIST hold-out class detection. Data and their reconstructions are shown for four difficult hold-out settings (1, 4, 7 and 9). Digit 2 is shown as an inlier example. The bottom two rows depict the reconstructions from four autoencoders (AE) and four NAEs trained on each setting. AEs reconstruct the outlier class well, while NAEs selectively reconstruct only inliers.

Zero-padded 32×32 MNIST images are used for model selection. Results are shown in Table 2.

It is known that constant images and SVHN images are particularly difficult outliers for generative models trained on a set of images with rich visual features (Nalisnick et al., 2019; Serrà et al., 2020). However, NAE detect such difficult outliers successfully. All models are able to discriminate noise outliers, indicating that their poor performance is not from the failure of training.

6.4. Sample Generation

Samples are generated from NAE using MCMC with OMI. Figure 7 shows the samples from NAEs trained on MNIST and on CelebA 64×64. The random initial states of the latent chain ($\mathbf{z}_0$) map to unrecognizable images. After the latent chain, OMI produces somewhat realistic images. MCMC on $\mathcal{X}$ refines the OMI images. Although quantitative image (in Appendix) quality metric for samples generated from NAE is not on a par with that of generative models which specialize in sampling, but the generated samples are indeed visually sensible.

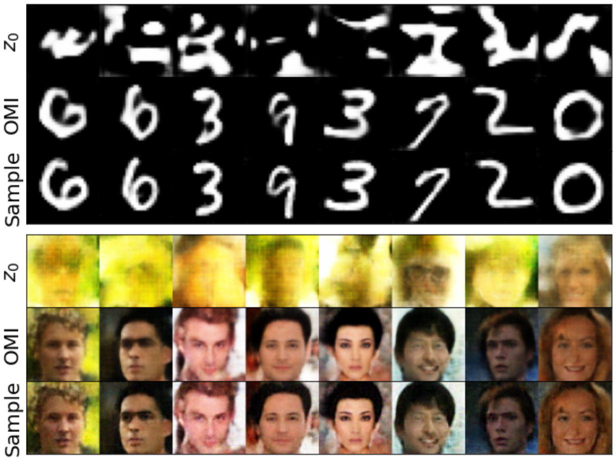

Figure 7. Sampling with NAEs trained on MNIST and CelebA 64×64. ($\mathbf{z}_0$) The random initialization of the latent chain. We visualize $f_d(\mathbf{z}_0)$. (OMI) Images after OMI. (Samples) Samples obtained after MCMC starting from OMI. OMI images and Samples corresponds to the red start and the green cross in Figure 4, respectively.

7. Discussion and Conclusion

Comparison to Other EBMs NAE uses Gibbs distribution to define a density function as in other EBMs (Eq. 4). The main difference between NAE and other EBMs is the choice of an energy function. However, this difference results in significant theoretical and practical consequences. First, we naturally incorporate the manifold hypothesis, i.e. the assumption that high-dimensional data lie on a low-dimensional manifold, into a model. Second, the energy function of NAE can be pre-trained as a conventional autoencoder. Third, more effective sampling can be performed by using OMI, leading to a more accurate density estimate.

Likelihood-based Outlier Detection and Inductive Bias The likelihood is considered as a poor decision function for outlier detection, after the failures of likelihood-based deep generative models such as VAE, PixelCNN++, and Glow (Nalisnick et al., 2019; Hendrycks et al., 2019). Those generative models fail to detect obvious outlier images which typically have low complexity. However, we believe that the failures should not be attributed to the use of the likelihood. There are likelihood-based models, particularly EBMs (Du & Mordatch, 2019; Grathwohl et al., 2020), including NAE, that show better outlier detection performance than VAE, PixelCNN++ and Glow. Instead, inductive bias of a generative model is likely to be responsible for the failure of detecting low-complexity outliers. It is reported that the likelihoods of the failed models are negatively correlated to the complexity of images (Serrà et al., 2020). Meanwhile,

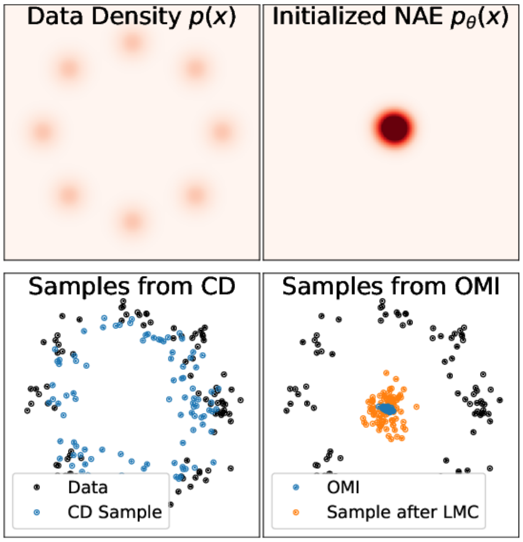

Figure 8. Sampling with randomly initialized NAE.

the reconstruction of low-complexity images are explicitly suppressed in NAE training, as the simple images tend to lie on the decoder manifold.

OMI in Early Stage of Training Sampling with OMI generates samples with high model density $p_\theta(\mathbf{x})$ even *in the early stage of training*. In fact, the early stage is where the advantage of OMI over CD is salient, because $p_\theta(\mathbf{x})$ differs from $p(\mathbf{x})$ significantly. Figure 8 visualizes samples generated via CD and OMI from a randomly initialized NAE and shows that CD fails to draw samples from $p_\theta(\mathbf{x})$.

OMI draws high-model-density proposals because it is designed to exploit the assumption that well-reconstructed points lie on the decoder manifold. We find that this assumption holds well for all experimental settings used in the paper.

Analytic Solution for Linear Case Linear NAEs reduce to Gaussian distributions. Consider $f_e(\mathbf{x}) = W\mathbf{x}$ and $f_d(\mathbf{z}) = W^\top \mathbf{z}$ with $W \in \mathbb{R}^{D_\mathbf{z} \times D_\mathbf{x}}$. Given the squared L^2 distance reconstruction error, the density of NAE is written as:

$$p_\theta(\mathbf{x}) = \exp(-\mathbf{x}^\top \Sigma^{-1}\mathbf{x}/2)/\Omega_\theta, \tag{17}$$

where $\Sigma^{-1} = 2(I - W^\top W)^2/T$. When the determinant of $I - W^\top W$ is non-zero, $p_\theta(\mathbf{x})$ becomes a well-defined Gaussian. Under certain conditions (see Appendix), the maximum likelihood estimate of Σ becomes the empirical covariance of data, as in a usual Gaussian distribution.

It is interesting to note that a linear VAE also reduces into a Gaussian, as it is equivalent to probabilistic PCA(Kingma & Welling, 2014). On the other hand, a linear autoencoder is equivalent to PCA (Bourlard & Kamp, 1988), which is not a generative model.

Conclusion We have introduced a novel interpretation of the reconstruction error as an energy function. Our interpretation leads to a novel class of probabilistic autoencoders, which shows impressive OOD detection performance and bridges EBMs and autoencoders.

Acknowledgements

S. Yoon and F.C. Park were supported in part by the SNU Artificial Intelligence Institute (AIIS), NRF Grant 2016R1A5A1938472 (SRRC), the SNU BK21+ Program in Mechanical Engineering, and SNU-IAMD. Y.-K. Noh was partly supported by NRF/MSIT (No. 2018R1A5A7059549, 2021M3E5D2A01019545), IITP/MSIT Artificial Intelligence Graduate School Program for Hanyang University (2020-0-01373), and Hanyang University (HY-2019).

References

Alain, G. and Bengio, Y. What regularized auto-encoders learn from the data-generating distribution. *The Journal of Machine Learning Research*, 15(1):3563–3593, 2014.

Bergmann, P., Löwe, S., Fauser, M., Sattlegger, D., and Steger, C. Improving unsupervised defect segmentation by applying structural similarity to autoencoders. *arXiv preprint arXiv:1807.02011*, 2018.

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

Bourlard, H. and Kamp, Y. Auto-association by multilayer perceptrons and singular value decomposition. *Biological cybernetics*, 59(4-5):291–294, 1988.

Brehmer, J. and Cranmer, K. Flows for simultaneous manifold learning and density estimation. *arXiv preprint arXiv:2003.13913*, 2020.

Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M. Hyperspherical variational auto-encoders. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pp. 856–865. Association For Uncertainty in Artificial Intelligence (AUAI), 2018.

Du, Y. and Mordatch, I. Implicit generation and modeling with energy based models. In Wallach, H., Larochelle, H., Beygelzimer, A., d Alche-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 3608–3618. Curran Associates, Inc., 2019.

Ghosh, P., Sajjadi, M. S. M., Vergari, A., Black, M., and Scholkopf, B. From variational to deterministic autoencoders. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1g7tpEYDS.

Gong, D., Liu, L., Le, V., Saha, B., Mansour, M. R., Venkatesh, S., and Hengel, A. v. d. Memorizing normality to detect anomaly: Memory-augmented deep autoencoder for unsupervised anomaly detection. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hkxzx0NtDB.

Grenander, U. and Miller, M. I. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(4):549–581, 1994.

Gutmann, M. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 297–304, 2010.

Hendrycks, D., Mazeika, M., and Dietterich, T. Deep anomaly detection with outlier exposure. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyxCxhRcY7.

Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

Japkowicz, N., Myers, C., Gluck, M., et al. A novelty detection approach to classification. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 1, pp. 518–523, 1995.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pp. 10215–10224, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *International Conference on Learning Representations (ICLR)*, 2014.

Lan, C. L. and Dinh, L. Perfect density models cannot guarantee anomaly detection. *arXiv preprint arXiv:2012.03808*, 2020.

Lu, Y. and Xu, P. Anomaly detection for skin disease images using variational autoencoder. *arXiv preprint arXiv:1807.01349*, 2018.

Lyudchik, O. Outlier detection using autoencoders. Technical report, 2016.

Meng, C., Yu, L., Song, Y., Song, J., and Ermon, S. Autoregressive score matching. *arXiv preprint arXiv:2010.12810*, 2020.

Nalisnick, E., Matsukawa, A., Teh, Y. W., Gorur, D., and Lakshminarayanan, B. Do deep generative models know what they don't know? In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1xwNhCcYm.

Nash, C. and Durkan, C. Autoregressive energy machines. In *International Conference on Machine Learning*, pp. 1735–1744. PMLR, 2019.

Ng, A. et al. Sparse autoencoder. *CS294A Lecture notes*, 2011.

Nijkamp, E., Hill, M., Zhu, S.-C., and Wu, Y. N. Learning non-convergent non-persistent short-run mcmc toward energy-based model. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32, pp. 5232–5242. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/2bc8ae25856bc2a6a1333d1331a3b7a6-Paper.pdf.

Parisi, G. Correlation functions and computer simulations. *Nuclear Physics B*, 180(3):378–384, 1981.

Pidhorskyi, S., Almohsen, R., and Doretto, G. Generative probabilistic novelty detection with adversarial autoencoders. In *Advances in neural information processing systems*, pp. 6822–6833, 2018.

Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., Depristo, M., Dillon, J., and Lakshminarayanan, B. Likelihood ratios for out-of-distribution detection. In *Advances in Neural Information Processing Systems*, pp. 14680–14691, 2019.

Rifai, S., Vincent, P., Muller, X., Glorot, X., and Bengio, Y. Contractive auto-encoders: explicit invariance during feature extraction. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pp. 833–840, 2011.

Roberts, G. O., Tweedie, R. L., et al. Exponential convergence of langevin distributions and their discrete approximations. *Bernoulli*, 2(4):341–363, 1996.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. *Learning Internal Representations by Error Propagation*, pp. 318–362. MIT Press, Cambridge, MA, USA, 1986. ISBN 026268053X.

Salimans, T., Karpathy, A., Chen, X., and Kingma, D. P. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

Serrà, J., Álvarez, D., Gómez, V., Slizovskaia, O., Núñez, J. F., and Luque, J. Input complexity and out-of-distribution detection with likelihood-based generative models. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SyxIWpVYvr.

Song, Y. and Kingma, D. P. How to train your energy-based models. *arXiv preprint arXiv:2101.03288*, 2021.

Steinwart, I., Hush, D., and Scovel, C. A classification framework for anomaly detection. *Journal of Machine Learning Research*, 6(Feb):211–232, 2005.

Tieleman, T. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pp. 1064–1071, 2008.

Tong, A., Yousefzadeh, R., Wolf, G., and Krishnaswamy, S. Fixing bias in reconstruction-based anomaly detection with lipschitz discriminators. *arXiv preprint arXiv:1905.10710*, 2019.

Vincent, P. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Vincent, P., Larochelle, H., Bengio, Y., and Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pp. 1096–1103, 2008.

Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pp. 681–688, 2011.

Xiao, Z., Yan, Q., and Amit, Y. Likelihood regret: An out-of-distribution detection score for variational auto-encoder. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 20685–20696. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/eddea82ad2755b24c4e168c5fc2ebd40-Paper.pdf.

Xiao, Z., Kreis, K., Kautz, J., and Vahdat, A. {VAEBM}: A symbiosis between variational autoencoders and energy-based models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=5m3SEczOV8L.

Xie, J., Lu, Y., Zhu, S.-C., and Wu, Y. A theory of generative convnet. In *International Conference on Machine Learning*, pp. 2635–2644. PMLR, 2016.

Xu, J. and Durrett, G. Spherical latent spaces for stable variational autoencoders. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 4503–4513, 2018.

Younes, L. On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates. *Stochastics: An International Journal of Probability and Stochastic Processes*, 65(3-4):177–228, 1999.

Zhao, D., Zhu, J., and Zhang, B. Latent variables on spheres for autoencoders in high dimensions. *arXiv*, pp. arXiv–1912, 2019.

Zhao, J., Mathieu, M., and LeCun, Y. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

Zhao, Y., Deng, B., Shen, C., Liu, Y., Lu, H., and Hua, X.-S. Spatio-temporal autoencoder for video anomaly detection. In *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1933–1941, 2017.

Zong, B., Song, Q., Min, M. R., Cheng, W., Lumezanu, C., Cho, D., and Chen, H. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=BJJLHbb0-.