Robustly Train Normalizing Flows via KL Divergence Regularization

Kun Song[1], Ruben Solozabal[1], Hao Li[2], Martin Takáč[1], Lu Ren[2*], Fakhri Karray[1]

[1]Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE
[2]Anhui University, Heifei City, Anhui, China
{kun.song, ruben.solozabal, martin.takac, fakhri.karray}@mbzuai.ac.ae, lihao6897@gmail.com, penny_lu@ahu.edu.cn

Abstract

In this paper, we find that the training of Normalizing Flows (NFs) are easily affected by the outliers and a small number (or high dimensionality) of training samples. To solve this problem, we propose a Kullback–Leibler (KL) divergence regularization on the Jacobian matrix of NFs. We prove that such regularization is equivalent to adding a set of samples whose covariance matrix is the identity matrix to the training set. Thus, it reduces the negative influence of the outliers and the small sample number on the estimation of the covariance matrix, simultaneously. Therefore, our regularization makes the training of NFs robust. Ultimately, we evaluate the performance of NFs on out-of-distribution (OoD) detection tasks. The excellent results obtained demonstrate the effectiveness of the proposed regularization term. For example, with the help of the proposed regularization, the OoD detection score increases at most 30% compared with the one without the regularization.

Introduction

Normalizing flows (NFs) provide a general mechanism for defining expressive probability distributions only requiring the specification of a base (usually simple) distribution and an invertible neural network f_θ. Due to their effectiveness, NFs have been widely applied in a range of applications, such as providing posterior for VAEs (Ardizzone et al. 2020; Mackowiak et al. 2021; Rudolph, Wandt, and Rosenhahn 2021), image generation (Yao et al. 2023; Zhang, Zhang, and McDonagh 2021), and out-of-distribution detection (Gudovskiy, Ishizaka, and Kozuka 2022; Rudolph, Wandt, and Rosenhahn 2021), etc.

It is well known that deep neural networks (DNNs) thrive on a large volume of training samples. In the case of limited training samples, leveraging prior information becomes vital for improving performance. An effective approach is initializing the parameters of DNNs using pre-trained models on extensive image datasets. However, this technique is challenging for NFs due to the lack of available pre-trained models, primarily because of the complexity of training NFs on large datasets. Notably, NFs' progress is often demonstrated using simpler datasets like Cifar10/100 or MNIST. Thus, the task of economically providing prior information to NFs is crucial. Alongside pre-trained models, regularization on learnable parameters stands out as a feasible and effective method (Liu et al. 2018; Dai et al. 2019) for enhancing Neural Network training. Therefore, we aim to explore regularization techniques to enhance NFs.

In this paper, we unveil that the training process of NFs is sensitive to both the quantity (dimensionality) of training samples and the presence of outliers. This inherent sensitivity poses challenges for NFs when dealing with a small number of samples or datasets containing outliers, which offers us insights for providing prior information to enhance its performance. For better demonstrating the aforementioned sensitivity issue, we use the example of a linear normalizing flow ($\theta = (\mathbf{U}, \mathbf{b})$, $f_\theta(\mathbf{x}) = \mathbf{U}^T\mathbf{x} + \mathbf{b}$) with Gaussian target distribution as an example. For linear NFs, $\mathbf{U}$ is solved by maximum likelihood estimation (MLE) which performs Singular Value Decomposition (SVD) on the inverse of the estimated covariance matrix $\mathbf{T} = \sum_{i=1}^{N} \frac{(\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T}{N} \in \mathbb{R}^{d\times d}$ of train samples, i.e., $\mathbf{U}\mathbf{U}^T = \mathbf{T}^{-1}$. Consequently, we can deduce that the training of linear NFs is affected by two key factors. First, when the sample number N is smaller than the dimension d, $\mathbf{T}$ lacks an inverse matrix, resulting in the failure to calculate $\mathbf{U}$. Second, outliers significantly bias the estimation of the covariance matrix $\mathbf{T}$ from its true value, as their contribution to $\mathbf{T}$ outweighs normal samples. Consequently, without prior information to address the sensitivity issues, the solution of $\mathbf{U}$ will be compromised significantly.

As for the non-linear case, the sensitivity problem referred still exists. We explain this using the conclusion obtained from a linear model. As well known, a complex distribution can be approximated by a Gaussian mixture model (GMM). Because a linear projection is enough to connect two Gaussian distributions, training non-linear NFs can be approximated by finding a series of linear projections parameterized by non-linear NFs, which project each Gaussian component to the target Gaussian distribution. If the non-linear NFs are powerful enough, those linear projections for different Gaussian components may be less related. In this way, we can consider that samples in each neighborhood only contribute to their own projection. Suppose that there is an outlier, it can be considered a Gaussian component consisting of one sample. Thus, the linear projection for

*Corresponding author.

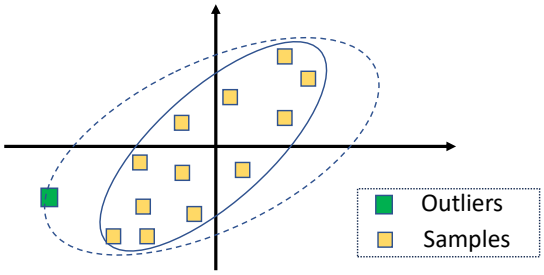


Figure 1: Illustration of the covariance matrix. The two axes of the ellipse correspond to the magnitudes of the eigenvalues of the covariance matrix. As observed, even a single outlier can change the covariance matrix significantly.

this one-sample component only has one sample to train. In this case, the NFs suffer from the lack of training samples. For a non-strict analysis, this assumption is reasonable because deep neural networks with ReLu activation are piecewise linear projections. We have experimentally validated this claim, and the results are presented in the Supplementary material.

To address this issue, we introduce a regularization that uses the Kullback-Leibler (KL) divergence between the Jacobian matrix ∇f_θ and a specified positive definite matrix $\mathbf{G}$ to regulate the training of NFs. This regularization term is equivalent to introducing a set of samples with a covariance matrix $\mathbf{G}$ into the training datasets. Consequently, by appropriately selecting $\mathbf{G}$, these supplementary samples reduce the impact of outliers on the covariance matrix estimation while increasing the number of training samples. This mitigates the sensitivity to outliers and the scarcity of training samples simultaneously. If we opt for $\mathbf{G}$ as an identity matrix, our proposed regularization is computationally efficient to implement on various types of NFs.

Furthermore, we proceed to conduct experiments to evaluate the efficacy of our approach, particularly in the context of out-of-distribution (OoD) detection. Our regularization term yields a performance improvement of 30% at most. Extensive ablation experiments validate our analysis and show the effectiveness of the introduced regularization term.

Sensitivity Problem of Normalizing Flows

Normalizing Flow

Given an invertible neural network $f_\theta : \mathbb{R}^d \to \mathbb{R}^d$, NFs aim at finding the parameters θ that project an arbitrary distribution $p_x(\mathbf{x})$ to a base distribution (such as the normal distribution) $p_z(\mathbf{z})$ where $\mathbf{z} = f_\theta(\mathbf{x})$. By using the change of variables formula, the probability of a sample $\mathbf{x} \in \mathbb{R}^d$ extracted from the complex distribution can be calculated as

$$p_x(\mathbf{x}) = p_z(f_\theta(\mathbf{x}))|\det \nabla f_\theta(\mathbf{x})|, \tag{1}$$

where $\nabla f_\theta(\mathbf{x}) \in \mathbb{R}^{d\times d}$ is the jacobian matrix of the projection $f_\theta(\mathbf{x})$.

Using the maximum likelihood estimation, the above equation can also be used to train NFs. Given N training samples $\{\mathbf{x}_i\}_{i=1}^N$, the objective of NFs can be described as follows,

$$\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \underbrace{[\log p_z(f_\theta(\mathbf{x}_i)) + \log|\det \nabla f_\theta(\mathbf{x}_i)|]}_{likelihood \quad function} + \alpha \underbrace{\log(q(\theta))}_{prior} \tag{2}$$

In Eq.(2), the first term is the likelihood and the second term is the prior information of θ, which is considered as a set of random variables described by the distribution $q(\theta)$. In MLE, $q(\theta)$ is useful because it can prevent the overfitting of outliers and make the calculation of the solutions tractable in ill-conditions (Mackowiak et al. 2021). For deep learning, there are mainly two methods to provide such prior information, i.e., pre-trained models on large dataset (Qiu et al. 2020) or designing regularization terms (Girosi, Jones, and Poggio 1995; Blanc et al. 2020; Liu et al. 2018). In this work, we focus on the last, and design a regularization term suitable for NFs.

Shortcomings of Normalizing Flows

Linear NFs. We begin with solving a linear NF $\mathbf{z} = \mathbf{U}^T\mathbf{x}+\mathbf{b}$ without prior information, which transforms a set of samples $\mathcal{X} = \{\mathbf{x}_i \in \mathbb{R}^d\}_{i=1}^N$ to a standard Gaussian distribution $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$. The optimal $\mathbf{U}$ and $\mathbf{b}$ can be obtained by minimizing the following objective function:

$$\mathcal{L}(\mathbf{U}, \mathbf{b}) = \sum_{i=1}^{N} |(\mathbf{U}^T\mathbf{x}_i + \mathbf{b})|_2^2/2 - N\log(\det(\mathbf{U})) \tag{3}$$

Theorem 1. *If the estimated covariance matrix of $\mathcal{X}$, i.e, $\boldsymbol{T} = \frac{\sum_{i=1}^N (\boldsymbol{x}_i-\bar{\boldsymbol{x}})(\boldsymbol{x}_i-\bar{\boldsymbol{x}})^T}{N}$, is full rank, the solution of the linear normalizing flow $\boldsymbol{z} = \boldsymbol{U}^T\boldsymbol{x}$ satisfies $(\boldsymbol{U}\boldsymbol{U}^T)^{-1} = \boldsymbol{T}$ and $\boldsymbol{b} = \boldsymbol{U}^T\bar{\boldsymbol{x}}$ where $\bar{\boldsymbol{x}} = \frac{\sum_{i=1}^N \boldsymbol{x}_i}{N}$.*

The proof is provided in the Supplementary material.

Theorem 1 indicates that the solution of $\mathbf{U}$ depends on the estimation of the covariance matrix $\mathbf{T}$ of $\mathcal{X}$. The covariance matrix estimation is a typical high dimension and small sample size (HDSS) problem (Wang et al. 2020; Daniels and Kass 2001; Donoho, Gavish, and Johnstone 2018). Therefore, the shortcomings of estimating the covariance matrix also become the ones of linear NFs. We call them the sensitivity problems, which consists of two aspects:

- If the dimension of samples d is larger than the number of samples N, the estimated covariance matrix $\mathbf{T}$ becomes a low-rank matrix, leading to a failure to calculate $\mathbf{U}$. This shortcoming prevents NFs from applications in which the number of samples is smaller than the dimensionality. This condition is very common in computer vision (Wang et al. 2017, 2019).
- As shown in Figure 1, the covariance matrix $\mathbf{T}$ is easily disturbed by outliers (corrupted by noises), which may cause the singular values of $\mathbf{T}$ larger than its true value. According to Theorem 1, the singular values of $\mathbf{U}$ would

be smaller than ground truth, and the distribution estimation of $\mathbf{x}$ presents a larger bias.

Deep NFs. As described in the introduction, we can convert the sensitivity problem of NFs to outliers into the problem of the low-rank covariance matrix in the linear case by using a GMM. In the following, a detailed analysis is provided.
Firstly, we introduce how to approximate the deep neural network to linear projections. For a deep NF f_θ, we perform the Taylor expansion on it, i.e. $f_\theta(\mathbf{x}_i) = f_\theta(\mathbf{x}_0) + \frac{\partial f_\theta(\mathbf{x})}{\partial \mathbf{x}}(\mathbf{x}_i - \mathbf{x}_0)$ where $\mathbf{x}_i \in O(\mathbf{x}_0)$. This means f_θ can be considered as a linear projection in the neighborhood $O(\mathbf{x}_0) = \{\mathbf{x} || \mathbf{x} - \mathbf{x}_0|_2^2 < a\}$ with the projecting matrix $\mathbf{L}(\mathbf{x}_0) = \frac{\partial f_\theta(\mathbf{x})}{\partial \mathbf{x}}(\mathbf{x}_0)$. A powerful neural network can fit complex functions, therefore, the values of Jacobian matrix $\mathbf{L}(\mathbf{x})$ in different neighborhoods could be quite different. Thus, the Jacobian matrix may only describe the information of samples in its own neighborhood. Considering that outliers are far away from the other samples, the dependence of outliers on other samples is minimum. So, if one Gaussian component belongs to an outlier, the Jacobian matrix converting this component to the target distributions will only depend on the outlier itself. Thus, the training of this Jacobian matrix suffers from the low-rank problem.

Because the sensitivity problem with outliers is described by the Jacobian matrix, the regularization of deep NFs to solve the sensitivity problem should be implemented on the Jacobian matrix for each sample. To the best of our knowledge, there is no work approaching the sensitivity problems of NFs. This motivates us to design a regularization that enhances the training of NFs.

Regularization for Deep Normalizing Flows

Motivation

Following the above analysis, we use the linear models to design a regularization term for NFs. First, we consider the problem where the covariance matrix $\mathbf{T}$ is low-rank. For easy analysis, we convert the objective function into the one whose learnable parameters are eigenvalues of $\mathbf{T}$ and $\mathbf{U}\mathbf{U}^T$.

According to Theorem 1, suppose $\sum_{i=1}^N \mathbf{x}_i/N = \bar{\mathbf{x}}$, the term $|(\mathbf{U}^T\mathbf{x}_i + \mathbf{b})|_2^2 = (\mathbf{x}_i - \bar{\mathbf{x}})^T\mathbf{U}\mathbf{U}^T(\mathbf{x}_i - \bar{\mathbf{x}})$ where $\mathbf{b} = \mathbf{U}^T\bar{\mathbf{x}}$. Here, we replace $\mathbf{U}\mathbf{U}^T$ with $\mathbf{W}\Gamma\mathbf{W}^T$, where $\mathbf{W}\mathbf{W}^T = \mathbf{I}$ and Γ is a diagonal matrix with positive diagonal elements. Then, we reformulate the objective function for linear NFs and present it as follows,

$$\mathcal{J}(\Gamma) = \max_{\mathbf{W}\mathbf{W}^T=\mathbf{I}} \left[Tr(\mathbf{T}\mathbf{W}^T\Gamma\mathbf{W}) - \frac{1}{2}\log(\det(\mathbf{W}^T\Gamma\mathbf{W})) \right] \tag{4}$$

where $\mathbf{T} = \frac{1}{N}\sum_{i=1}^N (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$ is the estimation of the covariance matrix of $\mathcal{X}$.

Theorem 2. $\max_{\boldsymbol{W}\boldsymbol{W}^T=\boldsymbol{I}} Tr(\boldsymbol{T}\boldsymbol{W}^T\Gamma\boldsymbol{W}) = Tr(\Lambda\Gamma)$*, where* $\boldsymbol{W}$ *and* Λ *are the results of the eigen-decomposition of* $\boldsymbol{T}$*, i.e.,* $\boldsymbol{W}\Lambda\boldsymbol{W} = \boldsymbol{T}$*.*

The proof can be found in the Supplementary material.

Denote the diagonal elements in Λ and Γ as $\Lambda_{ii} = \lambda_i$ and $\Gamma_{ii} = \gamma_i$, we can rewrite the objective function as

$$\mathcal{L}(\Gamma) = \sum_{i=1}^d \lambda_i\gamma_i - \sum_{i=1}^d \log(\gamma_i) \tag{5}$$

Thus, $\gamma_i > 0$ can be solved by the equation $\lambda_i = \frac{1}{\gamma_i}$. Because $\mathbf{W}$ are made of the eigenvectors of $\mathbf{T}$, the solution of Eq.(5) is the same as the one of Eq.(3), if $\lambda_i > 0$. This implies that the solution to $\{\gamma_i\}_{i=1}^d$ in Eq.(5) are independent. Therefore, we can ignore the solutions of λ_i if $\gamma_i = 0$, and give any positive number to them.

One way of solving low-rank problems is to find the zero eigenvalues of $\mathbf{T}$. However, this cannot be done in this case due to two reasons. First, this is impractical for NFs as they are solved by gradient descent. To approximate this solution. we can adopt the regularization strategy. For example, if we use the L_2-norm regularization, i.e., $reg(\mathbf{U}) = \|\mathbf{U}\|_F^2$, the solution of $\mathbf{U}$ equals performing an inverse matrix operation on $\hat{\mathbf{T}} = \mathbf{T} + \alpha\mathbf{I}$, where α is the regularization coefficient. The deduction is presented in the Supplementary material.
When we set any positive number to λ_i for $\gamma_i = 0$, there is a new covariance matrix $\hat{\mathbf{T}} = \mathbf{T} + \sum_i \frac{1}{\lambda_i}u_i u_i^T$. It means extracting samples from the orthogonal space of the one spanned by $\mathcal{X}$, whose covariance matrix is $\sum_i \frac{1}{\lambda_i}u_i u_i^T$. This motivates us to solve the outlier problem by drawing some in-distribution samples from the space of $\mathcal{X}$ to reduce the impact of outliers. Suppose we can select N_1 samples denoted by $\hat{\mathcal{X}} = \{\hat{\mathbf{x}}_t\}_{t=1}^{N_1}$ which have the same center of $\mathcal{X}$. To ensure $\hat{\mathcal{X}}$ being in-distribution, the variance of $\hat{\mathcal{X}}$ should be small. Thus the influence of the outliers in the new sample set $\mathcal{X} + \hat{\mathcal{X}}$ would kept small. Because the covariance matrix of $\hat{\mathcal{X}}$ is calculated as $\mathbf{T}^s = \sum_{i=1}^{N_1}(\hat{\mathbf{x}}_t - \bar{\mathbf{x}})(\hat{\mathbf{x}}_t - \bar{\mathbf{x}})^T/N_1$, the covariance matrix of $\mathcal{X} + \hat{\mathcal{X}}$ denoted by $\hat{\mathbf{T}}$ is expressed as

$$\hat{\mathbf{T}} = \frac{N * \mathbf{T} + N_1\mathbf{T}^s}{N + N_1} \tag{6}$$

According to the above analysis, if $\mathbf{T}^s$ is full rank, there is no need to find the zero eigenvalues of $\hat{\mathbf{T}}$. Considering the new solution is robust to outliers, this strategy can solve the sensitivity problems at the same time.

Theorem 3. *Suppose* $\boldsymbol{M} = \boldsymbol{U}\boldsymbol{U}^T$*, by adopting the KL-divergence* $KL(\boldsymbol{M}, \boldsymbol{G}^{-1}) = Tr(\boldsymbol{M}\boldsymbol{G}) - \log(\det(\boldsymbol{M}\boldsymbol{G}))$ *as the regularization to the linear normalizing flows, the solution of Eq.(3) is:*

$$\mathbf{M}^{-1} = \frac{N * \mathbf{T} + \alpha\mathbf{G}}{N + \alpha} \tag{7}$$

where α *is the coefficient of the regularization term.*

The proof is attached in the Supplementary material.

Because by setting $\alpha = N_1$ and $\mathbf{G} = \mathbf{T}^s$, Eq.(6) equals Eq.(7). This indicates that our regularization on $\mathbf{M} = \mathbf{U}\mathbf{U}^T$ can robustly estimate the covariance matrix of samples even the number is small and there are some outliers.

Remark 1. *As both, the KL divergence regularization and* L_2 *regularization have closed-form solutions for linear NF, one can calculate the estimated probability according to*

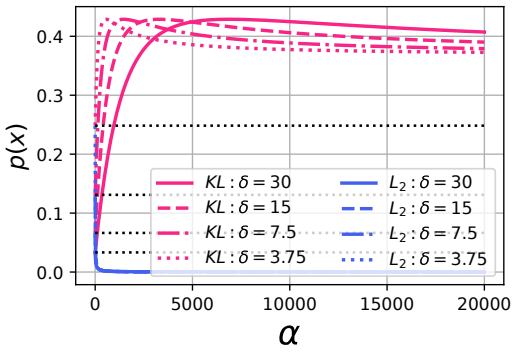


Figure 2: The probability estimated by linear normalizing flows solved with KL divergence regularization and L_2 regularization. δ is the variance of the original distribution. When the coefficient parameter increases, L_2 regularization makes the estimation collapse. The four black dot lines (different δ) are the probability without regularization, i.e., $\alpha = 0$.

the change of variables in Eq.(1). We use one dimension Gaussian distribution to demonstrate the difference between both regularizations. The results for different regularization coefficients α are presented in Figure 2. We can find that L_2 regularization lets the distribution estimation collapse. This validates the effectiveness of the proposed KL divergence regularization.

Remark 2. *In the linear case, our regularization term is similar to the objective function used in information-theorem distance metric learning (ITML) (Davis et al. 2007). However, ITML only highlights the linear invariant property of KL divergence between two matrices; it does not provide a perspective to explain why its objective function is good or bad. Our paper obtains the KL divergence regularization from a completely different motivation from ITML. We prove that the regularization term solves the sensitivity problems on the number of training samples and the outliers. This explains why our method is so good. Besides, our work provides a regularization term valid on deep neural networks rather than only for linear models.*

Regularization on Deep NFs

Because the sensitivity problem analysis approximates deep NFs as a set of linear projections, the KL divergence regularization of deep NFs should be implemented on the jacobian matrix $\mathbf{L}(\mathbf{x}) = \nabla f_\theta$. However, we can not do this in practice for two reasons:

(1) $\mathbf{L}(\mathbf{x})$ is the Jacobian matrix of the whole neural network at $\mathbf{x}$, which is computational expensive. Suppose the jacobian matrix of the i-th layer is $\mathbf{L}_i(\mathbf{x})$, there is $\mathbf{L}(\mathbf{x}) = \prod_{i=1}^{N} \mathbf{L}_i(\mathbf{x})$. For a small image, the Jacobian matrix is very large. For example, the size of Cifar10 is size $3 \times 32 \times 32$, the corresponding Jacobian matrix is 3072×3072. Therefore, multiplications of $\mathbf{L}(\mathbf{x})$ are very expensive.

(2) Many works (Allen-Zhu, Li, and Liang 2019; Razin and Cohen 2020; Cooper 2018; Liu et al. 2018) attributes the success of deep learning to the combination of the over-parameter property of neural networks and the stochastic gradient descent optimizer. For the same target, it is better to design objectives in the over-parameter way. For example, for a desirable matrix $\mathbf{W} = \mathbf{W}_1\mathbf{W}_2\mathbf{W}_3\cdots\mathbf{W}_l$, (Arora et al. 2019) proves that directly solving a matrix $\mathbf{W}$ is worse than solving the decomposed matrices $\{\mathbf{W}_i\}_{i=1}^{l}$.

In this way, we should regularize the Jacobian matrix of each layer rather than the whole neural network. We suppose there are R coupling blocks in the considered invertible neural network. Because $Tr(\mathbf{L}(\mathbf{x})\mathbf{L}(\mathbf{x})^T) = \|\mathbf{L}(\mathbf{x})\|_F^2$ and $\det(\mathbf{L}(\mathbf{x})\mathbf{L}(\mathbf{x})^T) = (\det(\mathbf{L}(\mathbf{x})))^2$, the regularization term can be formulated as

$$\mathbb{KLJ} = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} \sum_{i=1}^{R} (\|\mathbf{L}_i(\mathbf{x})\|_F^2 - 2\log(|\det(\mathbf{L}_i(\mathbf{x}))|) \quad (8)$$

In this paper, we adopt the coupling structure-based NFs to formulate our proposed KL divergence regularization.

As seen from Figure. 3, the jacobian matrix of the i-th coupling block is a lower triangle matrix presented as follows,

$$\mathbf{L}_i = \begin{bmatrix} diag(\mathbf{s}_i^1) & \mathbf{0} \\ \hat{\mathbf{L}}_i & \mathbf{I} \end{bmatrix} \quad (9)$$

where $\mathbf{s}_i \in \mathbb{R}^{d/2\times 1}$ is the output of the first neural networks NN_s^i and the $\hat{\mathbf{L}}_i \in \mathbb{R}^{d/2\times d/2}$ is the jacobian matrix of the second neural networks NN_t^i. As discussed before, the calculation of $\hat{\mathbf{L}}_i$ is computationally complex if the dimension of inputs and the number of layers of NN_t^i are large. In this way, we should find an efficient way to estimate it. Suppose the number of layers of NN_t^i is R_1, $\hat{\mathbf{L}}_i = \mathbf{J}_i^1\mathbf{J}_i^2\cdots\mathbf{J}_i^{R_1}$, where $\mathbf{J}_i^k$ is the jacobian matrix of the k-th layer of NN_t. Because $\|\mathbf{AB}\|_F^2 \le \|\mathbf{A}\|_F^2\|\mathbf{B}\|_F^2$, then $\|\hat{\mathbf{L}}_i\|_F^2 \le \prod_{k=1}^{R_1}\|\mathbf{J}_i^k\|_F^2$. Suppose the weight matrix of k-th layer is $\mathbf{W}_i^k$, the jacobian matrix $\mathbf{J}_i^k = \phi'((\mathbf{W}_i^k)^T\mathbf{x})\mathbf{W}_i^k$, where ϕ is the activation. Normally, $\phi'(t) \le 1$, thus $\|\mathbf{J}_i^k\|_F^2 < \|\mathbf{W}_i^k\|_F^2$. Therefore, there is an upper bound: $\|\hat{\mathbf{L}}_i\|_F^2 \le \prod_{R_1=1}^{r}\|\mathbf{W}_i^k\|_F^2$. The update of gradient of the term $\prod_{k=1}^{R_1}\|\mathbf{W}_i^k\|_F^2$ is cumbersome, another upper bound can be obtained by the inequality: $\prod_{i=1}^{R_1} a_i < (\frac{\sum_{i=1}^{R_1} a_i}{R_1})^{R_1}$. Thus, reducing the term $\sum_{k=1}^{R_1}|\mathbf{W}_i^k|_F^2$, the term $\|\hat{\mathbf{L}}_i\|_F^2$ is also reduced. Therefore, the KL divergence regularization can be reformulated as:

$$\begin{aligned}\mathbb{KLJ} = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})} & \underbrace{\tau\sum_{i=1}^{R}(\sum_{k=1}^{r}\|\mathbf{W}_i^k\|_F^2)}_{\mathbb{R}_1} + \underbrace{\sum_{i=1}^{R}|\mathbf{s}_i|_F^2}_{\mathbb{R}_2} \\ & - \underbrace{2\log(|\det(\mathbf{L}(\mathbf{x}))|)}_{\mathbb{R}_3}\end{aligned} \quad (10)$$

where τ is a coefficient to mitigate the impact of the boundary relaxation. $\mathbb{R}_1$ can be achieved by tuning the weight decay parameter. $\mathbb{R}_3$ equals the second term in objectives of

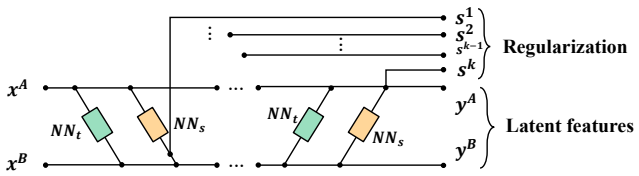


Figure 3: The illustration of the structure for implementing our regularization. Our regularization requires NFs to return the output of NN_s.

the normalizing flow. Only $\mathbb{R}_2$ needs to calculate. Since the dimension of $\mathbf{s}_i$ is d, the computation is very cheap. Thus, the objective of the coupling-based NFs can be formulated as

$$J = \mathcal{L}(\theta, \mathcal{X}) + \gamma \left(\sum_{i=1}^{R} |\mathbf{s}_i|_F^2 - 2\log\left(|\det(\mathbf{L}(\mathbf{x}))|\right) \right) \quad (11)$$

Remark 3. *Although the above discussions are about the coupling structure-based deep normalizing flows. The proposed regularization can be easily applied to other forms of NFs. This is because the term is* $\log(\det(\boldsymbol{L}(\boldsymbol{x})))$ *can be efficiently obtained on normalizing flows.*

Experimentation Details

We evaluate our proposed regularization on the generative classification task, as normalizing flows are proven to be one of the state-of-the-art techniques for building generative classifiers (GC) (Ovadia et al. 2019). Therefore, OoD detection using GC is an ideal task to validate the effectiveness of our proposed regularization. Code is available on GitHub[1].

Experimental Setting

In brief, we construct an invertible structure using Glow affine coupling blocks (Kingma and Dhariwal 2018) combined with random permutation layers[2]. The model is comprised of 60 coupling blocks performing convolutional operations. A detailed description of the architecture is presented in the Supplementary material. The classical class-conditional negative log-likelihood (cNNL) and the information bottleneck (IB) (Ardizzone et al. 2020) objective are adopted to test the model's OoD detection capabilities.

- **Class-NLL objective.** As a GC objective, the class-conditional negative log-likelihood (cNLL) loss is a common loss function for NF-based generative models. It is defined as $\mathcal{L}_{cNLL} = -\mathbb{E}\log(q_\theta(x|y))$ where x is the training samples and the y is the class conditioned. Many works in the literature, out the pure GC model can not deal with the complex distribution (Mackowiak et al. 2021).
- **Information Bottleneck.** The IB objective (Tishby and Zaslavsky 2015; Alemi et al. 2016) balances between discriminative and generative properties (Ng and Jordan 2001; Bernardo et al. 2007). It attempts to maximize the mutual information $I(Y, Z)$ between Y and the latent space Z, while minimizing the mutual information $I(X, Z)$ between X and Z. This forces the model to ignore irrelevant aspects of the input that can lead to overfitting. The IB objective was formulated for INNs generative models in (Ardizzone et al. 2020), which is proved to overcome the shortcomings of GCs. The loss function $\mathcal{L}_{IB} = \mathcal{L}_{I(X,Z)} - \gamma\mathcal{L}_{I(Z,Y)}$ the accuracy of DC models and the better uncertain estimation of GC models balances with the parameter γ. According to (Ardizzone et al. 2020), we set $\gamma = 1$ in this paper, empirically. We also evaluate $\gamma \to \infty$, which corresponds to a discriminative objective.

Datasets. We follow the experimental protocols in (Ardizzone et al. 2020). The models are trained in CIFAR10/100, featuring colored images of 3x32x32 pixel distributed across 10/100 categories, respectively. To examine the model's OoD capabilities, four additional datasets were employed. The first two are derived from CIFAR10: one incorporates random rotations and color shift, while the other integrates uniform random noise. The third dataset included is QuickDraw (Ha and Eck 2017), which consists of hand-drawn objects filtered to match the categories in CIFAR10. Lastly, the Tiny-ImageNet (Deng et al. 2009) validation dataset is also tested.

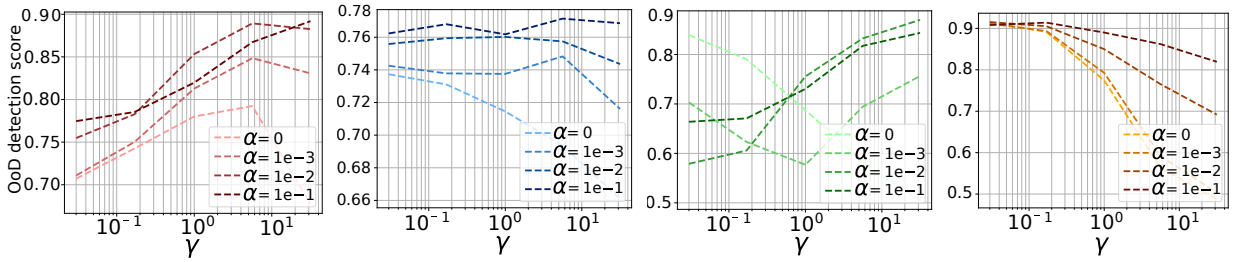
Metrics. We measure the OoD detection capabilities of the model as described in (Nalisnick et al. 2019). This work sets a threshold on the estimated likelihood to identify the OoD samples. Varying the detection threshold we obtain an ROC-AUC curve that it is used as a proxy for OoD detection. We also evaluate the entropy the model outputs on OoD samples, particularly, we measure the ratio between OoD entropy and in-distribution (ID) entropy measured as $(H(Y|X_{OoD}) - H(Y|X_{ID}))/H(Y|X_{ID})$, which is more robust to the discrete entropy of class prediction outputs $H(Y|X_{OoD})$ (Ardizzone et al. 2020). Finally, the model's ID accuracy is also reported.

Results

A summary of the results for the different objectives is presented in Table 1. For those experiments, we set the coefficient parameter of the proposed regularization to $\alpha = 0.01$ since it produces the best outcome. As seen from Table 1, we observe that our proposed regularization term consistently enhances the OoD detection capabilities across different objectives and different testing datasets. For the information bottleneck, if $\gamma = 1$, our regularization improves the OoD detection score on 8.3% on Rot-RGB at most and to 4.6% at least on TinyImageNet. If $\gamma = \infty$, the best improvement is 36.5% achieved on QuickDraw, and the smallest improvement is 8.2% achieved on TinyImaget. For cNLL, our regularization term can achieve an increment at most 6.5% on Rot-RGB. This demonstrates the effectiveness of our proposed regularization. Overall, our regularization shows

[1]https://github.com/Optimization-and-Machine-Learning-Lab/NFs_InformationBottleneck

[2]The model is implemented using the Framework for Easily Invertible Architectures (FrEIA) (Ardizzone et al. 2018).



(a) Rot-RGB Cifar10 (b) Tiny Imagenet (c) QuickDraw (d) Noisy Cifar10

Figure 4: OoD detection score for the different testing datasets. The model is trained on CIFAR10 using the IB objective varying γ to balance the generative versus discriminate capabilities of the model. Different levels of α for $\mathbb{KLJ}$ regularization are plotted. We find the regularization term increases the detection score significantly.

Method	OoD detection score (↑) Rot	Qui	Tin	Noi	Acc.
IB ($\gamma = 1$)	78.0	68.7	71.4	77.4	89.3
IB+$\mathbb{KLJ}$ ($\gamma = 1$)	85.3	75.5	76.0	84.9	85.6
IB ($\gamma \to \infty$)	67.7	50.5	66.1	48.7	**92.0**
IB+$\mathbb{KLJ}$ ($\gamma \to \infty$)	**88.2**	**87.0**	74.3	69.1	88.6
cNLL	69.7	84.7	73.8	**91.7**	47.1
cNLL+$\mathbb{KLJ}$	76.2	86.9	**76.4**	91.2	43.2
Method	OoD/ID Entropy ratio (↑) Rot	Qui	Tin	Noi	
IB ($\gamma = 1$)	1.45	0.76	2.30	0.15	
IB+$\mathbb{KLJ}$ ($\gamma = 1$)	**1.69**	**1.11**	**2.45**	**0.31**	
IB ($\gamma \to \infty$)	-0.07	-0.42	0.32	-0.23	
IB+$\mathbb{KLJ}$ ($\gamma \to \infty$)	0.03	-0.13	0.07	-0.19	
cNLL	0.09	-0.29	0.11	0.19	
cNLL+$\mathbb{KLJ}$	-0.06	-0.10	-0.19	0.12	

Table 1: The model's OoD capabilities were assessed through training on the CIFAR10 dataset. The same invertible neural network based on Normalizing Flows was used in all domains. Results are reported for $\alpha = 0.001$ in the $\mathbb{KLJ}$ regularization.

the best results when applied to the discriminant classifier ($\gamma = \infty$), achieving state-of-the-art results on two datasets. This may be because the discriminant classifier is more easily affected by the outliers.

On the other hand, although our regularization yields to good detection scores, it reduces in-distribution accuracy, which is a prevalent phenomenon in GC (Nalisnick et al. 2018). This may be because the discriminant classifier tends to fit any information correlated to the labels, even if the information is provided by the noises in the data. However, such information may not be good for the OoD detection, because similar noises can be observed in the OoD samples. The generative classifier tends to fit the inherent information in the dataset. So our regularization prevents the fit on noises and lets the discriminant classifier focus the inherent information on the dataset. As a result, it improves the OoD detection capabilities.

Specifically, the accuracy is calculated using a linear decision boundary. However, the proposed regularization is performed on the Jacobian matrix, which can be equivalently transformed to a distance metric. For a learnable distance metric, it prefers the decision boundary of K-nearest neighbor decision rule. To validate this, we also conduct experiments on distance metric learning. Due to the lack of space, we present the results in the Supplementary material. In that setting, the results show that our proposed regularization can enhance the discriminant information among the different classes.

Finally, we quantify the uncertainty of the model by studying the entropy of the model's prediction. Consistent with the findings in (Ardizzone et al. 2020), we observe nearly identical levels of entropy in ID and OoD predictions when training on the IB$_{(\gamma \to \infty)}$ and cNLL objectives. Worthy of note, our regularization term can increase the entropy noticeably on IB$_{(\gamma = 1)}$ on all datasets (see Figure 5).

Ablation Experiments

Parameter determination. As the coefficient parameter α is important, we show the results on different α, which are presented in Fig. 4. In the experiments, we set $\alpha = \{0, 10^{-3}, 10^{-2}, 10^{-1}\}$. As seen from Figure 4, we change the $\gamma = \{0.03, 0.17, 1, 5.62, 31.6\}$ (Tishby and Zaslavsky 2015) to control the generative and discriminant, and observe the performance with different α. From the results, we can find that the perfect α varies with the different datasets. But the method with regularization achieves the best performance in the searching grid of α, which indicates α is stable in the search grid. This gives the instruction to search α.

Outliers. We also test our regularization under noise conditions. Table 2 shows the OoD detection score for different levels of outliers which are corrupted by Gaussian additive noise $\sigma = 1$ in the training dataset. r means the percentage of the outliers in the training datasets. We observe that under relatively large levels of outliers, the performance of our methods is relatively stable. The methods without our regularization are easier to be influenced by the outliers, especially when IB($\gamma = \infty$). Such observations support our claim that our proposed algorithm makes the model robust

METHOD	$r = 0.05$					$r = 0.1$					$r = 0.15$				
	Rot	Qui	Tin	Noi	ACC.	Rot	Qui	Tin	Noisy	ACC.	Rot	Qui	Tin	Noisy	ACC.
IB ($\gamma = 1$)	75.9	67.6	66.4	74.1	85.9	70.6	63.4	67.8	73.1	80.5	74.6	66.6	59.7	61.4	65.0
IB + $\mathbb{KLJ}$ ($\gamma = 1$)	84.7	74.9	75.6	83.2	84.8	83.5	74.3	74.2	83.5	83.5	79.2	82.8	72.9	71.9	79.7
IB ($\gamma \to \infty$)	64.9	59.3	63.1	46.2	**89.2**	60.5	57.4	62.7	47.8	**85.1**	77.9	54.4	54.9	57.4	46.5
IB + $\mathbb{KLJ}$($\gamma \to \infty$)	**86.8**	**86.1**	74.3	68.4	87.1	**86.5**	**85.6**	72.2	66.8	84.6	**80.3**	**84.0**	**83.2**	70.8	62.6
cNLL	67.9	82.3	73.8	**90.2**	45.0	62.5	77.7	70.6	87.1	46.7	43.8	57.4	75.3	65.8	71.5
cNLL + $\mathbb{KLJ}$	75.8	85.2	**76.4**	90.1	41.4	72.8	84.6	**75.3**	**88.1**	39.5	39.4	71.6	81.5	**73.1**	**84.9**

Table 2: OoD detection score on the four testing datasets and ID accuracy across various levels of Gaussian noise σ added to the training images. Results are reported for $\alpha = 0.01$ in the KLJ regularization.

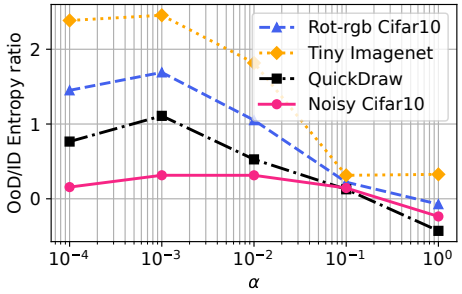
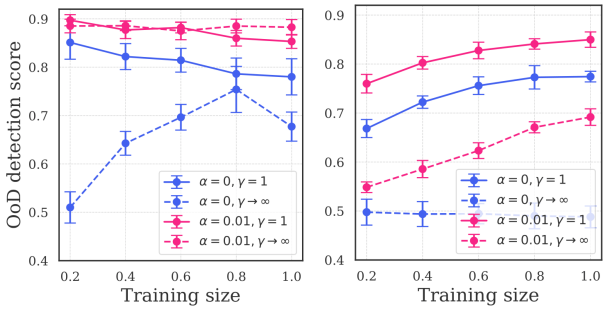


Figure 5: Entropy ratio between the out-of-distribution and in-distribution domains. Results are depicted for the IB loss with $\gamma = 1$ under different levels of $\mathbb{KLJ}$ regularization.



(a) Rot-RGB Cifar10 (b) Noisy Cifar10

Figure 6: OoD detection score for different sizes on the training CIFAR10 dataset, indicated as the ratio w.r.t. the original size.

to noises (or outliers).

Number of training samples. Because we claim our proposed regularization can prevent the negative influence of insufficient training samples, we test our regularization term for different training sample sizes. As seen from Figure 6, the models are trained with on a reduced percentage of the initial training dataset $p = \{0.2, 0.4, 0.6, 0.8\}$. We find that, when the training number is small, the methods with our regularization surpass the methods without the regularization. For example, in Rot-RGB, the results of and $\gamma = \infty$ without regularization is very bad at $p = 0.2$. But, if it adopts our regularization, the performance increase significantly. For Noise Cifar10, we can also observe the same phenomenon. For example, our regularization increases $\gamma = 1$ about 10% and $\gamma = \infty$ about 5% at $p = 0.2$, respectively. Although the incrase on the method $\gamma = \infty$ is not too much, our regularization lets its performance increase when training samples increase. Because the generative classifier is more robust than the DC for OoD detection, the performance increase may be caused by our regularization to prevent the negative influence of nosies. This may be the evidence to support our regularization makes the methods robust to outliers. Based on the perspective of robust to outliers, we can explain why the performance of methods with our regularization terms is stable in Figure 6, and the methods without our regularization tend to drop their performance when the number of training sample increase.

Conclusions

In this paper, we reveal that the training of the normalizing flows is sensitive to the number of training samples and outliers. Especially, when the number of training samples is very small, the training of NFs may fail. This is because the Maximal likelihood strategy needs prior information on the learnable parameters. To solve this problem, we propose a regularization term based on the KL divergence to regularize the learnable parameters. The regularization term lets estimated covariance matrix close to a given distribution. In this way, if the given distribution is selected well, the outliers and the low-rank problem will be addressed simultaneously. At last, we conduct extensive experiments to validate the effectiveness of the proposed regularization term.

This demonstrates that the value of our regularization term can be used to detect overfitting. And as proposed in this paper, this term can be used to successfully regularize the DML loss function to overcome overfitting.

Acknowledgments

This research was partially supported by the National Natural Science Foundation of China No.62106215.

References

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. 2016. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*.

Allen-Zhu, Z.; Li, Y.; and Liang, Y. 2019. Learning and generalization in overparameterized neural networks, going beyond two layers. *Advances in neural information processing systems*, 32.

Ardizzone, L.; Bungert, T.; Draxler, F.; Köthe, U.; Kruse, J.; Schmier, R.; and Sorrenson, P. 2018. Framework for Easily Invertible Architectures (FrEIA).

Ardizzone, L.; Mackowiak, R.; Rother, C.; and Köthe, U. 2020. Training normalizing flows with the information bottleneck for competitive generative classification. *Advances in Neural Information Processing Systems*, 33: 7828–7840.

Arora, S.; Cohen, N.; Hu, W.; and Luo, Y. 2019. Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems*, 32.

Bernardo, J.; Bayarri, M.; Berger, J.; Dawid, A.; Heckerman, D.; Smith, A.; and West, M. 2007. Generative or discriminative? getting the best of both worlds. *Bayesian statistics*, 8(3): 3–24.

Blanc, G.; Gupta, N.; Valiant, G.; and Valiant, P. 2020. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. In *Conference on learning theory*, 483–513. PMLR.

Cooper, Y. 2018. The loss landscape of overparameterized neural networks. *arXiv preprint arXiv:1804.10200*.

Dai, B.; Liu, Z.; Dai, H.; He, N.; Gretton, A.; Song, L.; and Schuurmans, D. 2019. Exponential family estimation via adversarial dynamics embedding. *Advances in Neural Information Processing Systems*, 32.

Daniels, M. J.; and Kass, R. E. 2001. Shrinkage estimators for covariance matrices. *Biometrics*, 57(4): 1173–1184.

Davis, J. V.; Kulis, B.; Jain, P.; Sra, S.; and Dhillon, I. S. 2007. Information-theoretic metric learning. In *Proceedings of the 24th international conference on Machine learning*, 209–216.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Donoho, D. L.; Gavish, M.; and Johnstone, I. M. 2018. Optimal shrinkage of eigenvalues in the spiked covariance model. *Annals of statistics*, 46(4): 1742.

Girosi, F.; Jones, M.; and Poggio, T. 1995. Regularization theory and neural networks architectures. *Neural computation*, 7(2): 219–269.

Gudovskiy, D.; Ishizaka, S.; and Kozuka, K. 2022. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 98–107.

Ha, D.; and Eck, D. 2017. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*.

Kingma, D. P.; and Dhariwal, P. 2018. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31.

Liu, W.; Lin, R.; Liu, Z.; Liu, L.; Yu, Z.; Dai, B.; and Song, L. 2018. Learning towards minimum hyperspherical energy. *Advances in neural information processing systems*, 31.

Mackowiak, R.; Ardizzone, L.; Kothe, U.; and Rother, C. 2021. Generative classifiers as a basis for trustworthy image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2971–2981.

Nalisnick, E.; Matsukawa, A.; Teh, Y. W.; Gorur, D.; and Lakshminarayanan, B. 2018. Do deep generative models know what they don't know? *arXiv preprint arXiv:1810.09136*.

Nalisnick, E.; Matsukawa, A.; Teh, Y. W.; and Lakshminarayanan, B. 2019. Detecting out-of-distribution inputs to deep generative models using typicality. *arXiv preprint arXiv:1906.02994*.

Ng, A.; and Jordan, M. 2001. On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Advances in neural information processing systems*, 14.

Ovadia, Y.; Fertig, E.; Ren, J.; Nado, Z.; Sculley, D.; Nowozin, S.; Dillon, J.; Lakshminarayanan, B.; and Snoek, J. 2019. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32.

Qiu, X.; Sun, T.; Xu, Y.; Shao, Y.; Dai, N.; and Huang, X. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10): 1872–1897.

Razin, N.; and Cohen, N. 2020. Implicit regularization in deep learning may not be explainable by norms. *Advances in neural information processing systems*, 33: 21174–21187.

Rudolph, M.; Wandt, B.; and Rosenhahn, B. 2021. Same same but differnet: Semi-supervised defect detection with normalizing flows. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 1907–1916.

Tishby, N.; and Zaslavsky, N. 2015. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, 1–5. IEEE.

Wang, J.; Zhou, F.; Wen, S.; Liu, X.; and Lin, Y. 2017. Deep metric learning with angular loss. In *Proceedings of the IEEE international conference on computer vision*, 2593–2601.

Wang, Q.; Xie, J.; Zuo, W.; Zhang, L.; and Li, P. 2020. Deep cnns meet global covariance pooling: Better representation and generalization. *IEEE transactions on pattern analysis and machine intelligence*, 43(8): 2582–2597.

Wang, X.; Han, X.; Huang, W.; Dong, D.; and Scott, M. R. 2019. Multi-similarity loss with general pair weighting for deep metric learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5022–5030.

Yao, J.-E.; Tsao, L.-Y.; Lo, Y.-C.; Tseng, R.; Chang, C.-C.; and Lee, C.-Y. 2023. Local Implicit Normalizing Flow for

Arbitrary-Scale Image Super-Resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1776–1785.

Zhang, M.; Zhang, A.; and McDonagh, S. 2021. On the out-of-distribution generalization of probabilistic image modelling. *Advances in Neural Information Processing Systems*, 34: 3811–3823.