# Learning Conditionally Untangled Latent Spaces using Fixed Point Iteration

Victor Enescu
victor.enescu@lip6.fr
Hichem Sahbi
hichem.sahbi@lip6.fr

Sorbonne University
CNRS, LIP6
F-75005, Paris, France

**Abstract**

Normalizing flows (NFs) are powerful generative models that map arbitrary complex (ambient) distributions to simple (latent) ones such as the monomodal gaussian. Despite their ability in modeling and sampling highly nonlinear manifolds, NFs are less effective in assigning labels to the generated data. This stems from the insufficient expressivity of monomodal gaussians, and also the difficulty in learning multimodal distributions in the latent spaces.

In this paper, we devise a multimodal NF-based approach suitable both for image generation and classification. The particularity of our method resides in its ability to design multimodal gaussian distributions as a part of NF training using an objective function that mixes a likelihood term and a Kullback-Leibler Divergence (KLD) criterion. The parameters of the trained gaussians (namely means and covariance matrices) are obtained as an interpretable fixed-point solution of this objective function. Besides, our proposed method avoids the overwhelming and sensitive process of tuning the learning rates as required by gradient descent. Extensive experiments conducted on different datasets, including CIFAR10, CIFAR100 and ImageNet, show competitive performances of our method against different baselines as well as the related work. Code is available in this link https://github.com/vic-ene/NFPI.

## 1 Introduction

Deep generative models have attracted unprecedented attention in computer vision [45], by training models capable of mapping complex *ambient* spaces into simpler *latent* ones (and vice-versa). Ambient spaces refer to input data drawn from existing but unknown probability distributions (possibly sitting on top of complex nonlinear manifolds) whereas latent spaces correspond to learned representations, lying on notoriously more tractable distributions such as the gaussian. Generative modeling has also been accelerated thanks to the improvement of computational resources that allow training larger and increasingly more accurate generative models, most notably Normalizing Flows (NFs) [12, 13, 29]. The latter are unique in their ability to learn bijective (invertible) transformations, useful for *exact* density estimation and image generation. Nonetheless, in their standard form, NFs coerce the data to follow a single monomodal gaussian in the latent space, which makes them relatively unsuitable for class-dependent image generation. Existing variants built upon gaussian mixture models

(GMMs) for label conditioning are more expressive, and allow enhancing not only the generative properties of NFs, but also their discrimination power. Furthermore, in order to increase their expressiveness, gaussian components can also be learned using backpropagation and gradient descent. However, training covariance matrices with gradient descent does not guarantee the inherent properties of these matrices, namely their positive-definiteness and their symmetry; besides, the obtained solutions are not necessarily interpretable. As a result, most of the existing NF frameworks fix (or handcraft) the means and the covariance matrices resulting into suboptimal models. In this context, recent work [2, 3, 18, 38] addresses these issues by training gaussian components using a plain likelihood loss, and this may result in overlapping components, detrimental to image generation and classification.

In order to circumvent these limitations, we propose a new method that learns expressive gaussian components (namely means and covariance matrices). Our solution is based on the optimization of an objective function mixing a data term, that maximizes the likelihood of training samples conditional to their class-labels, and a Kullback-Leibler Divergence (KLD) criterion which maximizes the separability of the gaussian components. This separation process is achieved while preserving the continuum of data through different classes; in other words, visually similar (resp. dissimilar) classes are mapped to separate *but nearby (resp. distant)* gaussians in the latent space. The solution of this objective function is obtained using a fixed-point iteration (FPI) framework that allows obtaining more accurate and interpretable analytic solutions compared to black box gradient descent, which requires a cumbersome tuning of learning rate policies among other hyperparameters. Indeed, the proposed FPI framework has a unique *step-size* hyperparameter which also controls the KLD criterion, and its setting is more intuitive and conditioned, in practice, by the convergence, the discriminative and the generative properties of the trained NFs. To the best of our knowledge, this work is the first FPI framework that learns multimodal gaussian distributions as a part of NF design.

Considering all the aforementioned issues, the main contributions of this work include

(i) A novel method that accurately learns non-overlapping multimodal gaussians using a fixed-point iteration framework that overcomes the cumbersome "learning-rate policy design" which is crucial for the success of gradient descent.

(ii) The first NF-based framework that learns anisotropic multimodal gaussians (their means and covariances) in the latent space using an objective function mixing a maximum likelihood term and a KLD criterion. The solution of this objective function, obtained with FPI, is interpretable and guarantees the properties of the trained covariance matrices (mainly their symmetry and positive-definiteness).

(iii) Extensive experiments involving NF models, on different datasets, show high discrimination and generation performances of the proposed method against different baselines as well as the related work.

## 2 Related Work

Conditioning in NFs [13, 29] has been extensively studied using GMMs [2, 3, 7, 18, 23, 27, 30, 38, 48, 50, 53, 54] where labels are associated to unique gaussian components, and this allows achieving simultaneous conditional image generation and classification while increasing the expressiveness of NFs. In all these existing approaches, GMM training is achieved with standard gradient descent; for instance, [3, 18] learn portions of discriminative features,

and [2, 38, 50, 54] learn only the means and use fixed isotropic identity covariance matrices whereas [7, 23, 27, 30, 50] proceed differently by randomly initializing the means, and keeping them fixed during training. As gradient descent cannot guarantee the positive definiteness of the trained covariance matrices, most of the existing approaches consider these matrices as fixed and isotropic, and only very few works [18, 29] train these covariances. In contrast, our NF training is achieved on top of anisotropic gaussians which are more expressive compared to the isotropic ones. Besides, similarly to information bottleneck [2, 38], our approach relies on a Kullback-Leibler Divergence (KLD) criterion that separates gaussians, but differs in the way this criterion is defined; our criterion is based on exponentially decaying KLD terms that provide more stable and convergent solutions.

To the best of our knowledge, most of the existing NF models are reliant on gradient descent and our proposed method is the first to learn multimodal gaussians in the latent space (as a part of NF training) using a Fixed-Point Iteration (FPI) framework. FPI-based methods have received less attention in the literature compared to gradient-based methods when optimizing generative models in general. For instance, [25] uses FPI to speedup very long sampling processes based on gaussian noise prediction [39], and [4, 21] rely on single fixed-point iteration that greatly reduces model size and computational requirements. FPI has also been incorporated in the loss of GANs [22], with the work of [40, 41] in an effort to speedup convergence and diversity as well as image quality.

## 3 A Glimpse on Normalizing Flows

Let $\mathbf{X}$ be a random variable standing for all possible images taken from an existing but *unknown* probability distribution $P_{\mathbf{X}}$ in an ambient space $\mathcal{X} \subseteq \mathbb{R}^d$. Considering $\mathbf{Z}$ as a latent representation associated to $\mathbf{X}$ drawn from a *known* probability distribution $P_{\mathbf{Z}}$ in a latent space $\mathcal{Z} \subseteq \mathbb{R}^d$; normalizing flows aim at learning a diffeomorphism $f$ from $\mathcal{X}$ to $\mathcal{Z}$ together with its inverse $g$, where $f$ (resp. $g$) is used for classification (resp. generation) and is referred to as normalizing (resp. generative) direction. Given $\mathbf{x} \in \mathcal{X}$, one may write

$$P_{\mathbf{X}}(\mathbf{x}) = P_{\mathbf{Z}}(f(\mathbf{x})) \left| \det \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \right| = P_{\mathbf{Z}}(f(\mathbf{x})) \left| \det \mathbf{J}_{f(\mathbf{x})} \right|, \tag{1}$$

where $\mathbf{J}_{f(\mathbf{x})} \in \mathbb{R}^{d \times d}$ is the Jacobian of $f$ w.r.t. $\mathbf{x}$ and $|\det(.)|$ stands for determinant magnitude. In practice, $f$ is a neural network composed of several smaller invertible flows chosen to make $\mathbf{J}_{f(\mathbf{x})}$ computationally efficient. As defined in [12, 29], each flow is usually made of an actnorm layer, an invertible $1 \times 1$ convolution, and a coupling Layer stacked together. By rewriting the d-dimensional vector $\mathbf{x}$ as $\mathbf{x}_{1:d}$, a coupling Layer maps $\mathbf{x}_{1:d}$ to two subvectors $\tilde{\mathbf{x}}_{1:d/2}$ and $\tilde{\mathbf{x}}_{d/2+1:d}$ being $\tilde{\mathbf{x}}_{1:d/2} = \mathbf{x}_{1:d/2}$ and $\tilde{\mathbf{x}}_{d/2+1:d} = \mathbf{x}_{d/2+1:d} \odot \exp(s(\mathbf{x}_{1:d/2})) + b(\mathbf{x}_{1:d/2})$, $s(.)$, $b(.)$ are two neural networks, $\odot$ the Hadamard product and $\exp(.)$ is applied entrywise. Invertible $1 \times 1$ convolutions are generalized permutation layers that enhance expressivity by allowing permutations between image channels to be learned [29]. An actnorm layer is an invertible equivalent of batch normalization [26] that increases stability and performance. NFs are usually trained to minimize the negative log-likelihood of Eq. 1. From transport theory point of view [49], NFs pushforward a complex ambient distribution into a simpler latent one as the monomodal normal. Subsequently, we take a step further to make the latent distribution multimodal while also being able to model the continuum between different classes, and this balances the generation and the discrimination power of the resulting NFs as also shown in experiments.

# 4 Proposed Method

Let's consider a collection $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N} \subset \mathcal{X} \times \mathcal{Y}$ of labeled images originating from an ambient space $\mathcal{X}$, and the underlying class labels taken from a discrete set $\mathcal{Y} = \{1, \ldots, K\}$. Given a pair $(\mathbf{x}, \mathbf{y}) \in \mathcal{X} \times \mathcal{Y}$, one may write the conditional form of Eq. 1 as

$$P_{\mathbf{X}}(\mathbf{x}|\mathbf{y}) = P_{\mathbf{Z}}(f(\mathbf{x})|\mathbf{y}) \left| \det \mathbf{J}_{f(\mathbf{x})} \right|, \tag{2}$$

where $P_{\mathbf{Z}}(.|\mathbf{y})$ is set a priori to a given distribution, namely a gaussian mixture. The purpose here is to train the parameters of the NF (denoted as $\Theta$) together with the hyperparameters of the underlying gaussians (referred to as $\Psi = \{(\mu_{\mathbf{y}}, \Sigma_{\mathbf{y}})\}_{\mathbf{y} \in \mathcal{Y}}$) while ensuring better discriminative and generative performances of the resulting NF. As previous works [2, 18, 27] have noted that gaussians separated using only a likelihood loss produce rich continuum, but highly overlapping gaussians (that are unfit for controlled image generation), we add a repulsing criterion based on Kullblack-Leibler Divergence (KLD) in order to control the tradeoff between separability and continuity of the gaussians as defined in Eq. 3

$$\mathcal{L}_{KLD}(\Psi) = \sum_{\mathbf{y}, \mathbf{y}' \in \mathcal{Y} : \mathbf{y} \neq \mathbf{y}'} \exp\left( - \frac{\mathrm{KLD}(\mathcal{N}_{\mathbf{y}} \,||\, \mathcal{N}_{\mathbf{y}'})}{\pi_{KLD}(.) \cdot \alpha} \right), \tag{3}$$

here $\pi_{KLD}(.)$ is a customizable function which returns a scalar depending on all the pairwise KLD terms (set as shown later), and $\alpha$ is a scalar that regulates the impact of the KLD, chosen in practice between 0.1 and 10. The double sum of forward and reverse KLD is similar to the negative of the symmetrized Kullback-Leibler Divergence [1] [42], but differs in the way the exponential and negative sign are applied individually to each pair $(\mathbf{y}, \mathbf{y}')$. This is a major distinction which ensures that the repulsing force, associated to all gaussian pairs, is lower bounded and thereby vanishes once a certain degree of separation has been reached. In contrast, the negative of the standard symmetrized Kullback-Leibler Divergence [42] is not lower bounded and decreases continuously even when gaussians are well separated, eventually yielding unstable (non-convergent) solution. Besides, $\mathcal{L}_{KLD}(\Psi)$ ensures that all non-separated classes $\{(\mathbf{y}, \mathbf{y}')\}$ are *evenly* impactful compared to other KLD variants that commute the exponential and the sum in Eq. 3; these variants are dominated by distant classes leading to large KLD values and small $\mathcal{L}_{KLD}(\Psi)$ even when some gaussian pairs are not sufficiently well separated.

## 4.1 Fixed-Point Iteration (FPI)

Considering the definition of the KLD loss in Eq. 3, we define the global loss used to optimize the NF and gaussian parameters as following

$$\mathcal{L}(\Theta, \Psi) = \mathcal{L}_{NF}(\Theta, \Psi) + \lambda \mathcal{L}_{KLD}(\Psi), \tag{4}$$

being $\mathcal{L}_{NF}(\Theta, \Psi)$ the standard negative log-likelihood loss (see for instance [27]) and $\lambda$ is a scalar balancing the impact of the KLD loss. Unlike gradient based methods, that heavily rely on *learning-rate-update-policy* to reach an optimal solution, our proposed method in this paper is more effective, and it is based on a fixed-point iteration (FPI) framework whose behavior is controlled with $\lambda$. This framework iteratively updates the hyperparameters $\Psi$, rewritten with a superscript $t$ as $\Psi^{(t)}$ at a given timestep $(t)$, by taking into account $\Psi^{(t-1)}$

[1] the symmetrized KLD is known as Jensen–Shannon Divergence.

at the previous timestep $(t-1)$, till reaching convergence. Equations used to obtain these hyperparameters correspond to the stationary solution of Eq. 4 as the limit of the fixed-point iterations as shown in the following proposition.

**Proposition 1** *Let $N_{\mathbf{y}}$ denote the number of training samples belonging to class $\mathbf{y}$ and define a kernel between classes $\{(\mathbf{y},\mathbf{y}')\}$ as $K_{\mathbf{y}\mathbf{y}'} = \frac{1}{\pi_{KLD}(.)\cdot\alpha}\cdot\exp\left(-\frac{KLD(\mathcal{N}_{\mathbf{y}}\,||\,\mathcal{N}_{\mathbf{y}'})}{\pi_{KLD}(.)\cdot\alpha}\right)$. The optimality conditions of Eq. 4 lead to the solution*

$$\mu_{\mathbf{y}}^{(t)} = \frac{1}{N_{\mathbf{y}}}\sum_{i=1}^{N_{\mathbf{y}}} f(\mathbf{x}_i) + \frac{\lambda}{K(K-1)}\,\Sigma_{\mathbf{y}}^{(t-1)}\sum_{\mathbf{y}'\neq\mathbf{y}}^{K}\left(K_{\mathbf{y}\mathbf{y}'}\left(\Sigma_{\mathbf{y}'}^{(t-1)}\right)^{-1} + K_{\mathbf{y}'\mathbf{y}}\left(\Sigma_{\mathbf{y}}^{(t-1)}\right)^{-1}\right)\left(\mu_{\mathbf{y}}^{(t-1)} - \mu_{\mathbf{y}'}^{(t-1)}\right) \quad (5)$$

$$\Sigma_{\mathbf{y}}^{(t)} = A^{-1/2}\,(A^{1/2}\,B\,A^{1/2})^{1/2}\,A^{-1/2}, \quad (6)$$

*being*

$$\begin{aligned} A &= \left(\Sigma_{\mathbf{y}}^{(t-1)}\right)^{-1}\left(1+\frac{\lambda}{K(K-1)}\sum_{\mathbf{y}'\neq\mathbf{y}}\left(K_{\mathbf{y}\mathbf{y}'}-K_{\mathbf{y}'\mathbf{y}}\right)\right) \\ &\quad + \frac{\lambda}{K(K-1)}\left(\Sigma_{\mathbf{y}}^{(t-1)}\right)^{-1}\sum_{\mathbf{y}'\neq\mathbf{y}} K_{\mathbf{y}'\mathbf{y}}\left[(\mu_{\mathbf{y}}^{(t-1)}-\mu_{\mathbf{y}'}^{(t-1)})(\mu_{\mathbf{y}}^{(t-1)}-\mu_{\mathbf{y}'}^{(t-1)})^{\top}+\Sigma_{\mathbf{y}'}^{(t-1)}\right]\left(\Sigma_{\mathbf{y}}^{(t-1)}\right)^{-1} \\ B &= \frac{1}{N_{\mathbf{y}}}\sum_{i=1}^{N_{\mathbf{y}}}\left(f(\mathbf{x}_i)-\mu_{\mathbf{y}}^{(t-1)}\right)\left(f(\mathbf{x}_i)-\mu_{\mathbf{y}}^{(t-1)}\right)^{\top} + \frac{\lambda}{K(K-1)}\sum_{\mathbf{y}'\neq\mathbf{y}} K_{\mathbf{y}\mathbf{y}'}\Sigma_{\mathbf{y}}^{(t-1)}\left(\Sigma_{\mathbf{y}'}^{(t-1)}\right)^{-1}\Sigma_{\mathbf{y}}^{(t-1)}. \end{aligned} \quad (7)$$

Details of the proof are omitted and result from the gradient's optimality conditions of Eq. 4. Considering the above proposition, the optimal solution is obtained iteratively as a fixed point of Eqs. 5 and 6 with $\mu_{\mathbf{y}}^{(0)}$ and $\Sigma_{\mathbf{y}}^{(0)}$ initially set to the mean and the covariance of training data in $\mathcal{Z}$ belonging to class $\mathbf{y}$. Note that convergence is observed in practice in few iterations, and the underlying fixed points, denoted as $\{(\tilde{\mu}_{\mathbf{y}},\tilde{\Sigma}_{\mathbf{y}})\}_{\mathbf{y}}$, correspond to the parameters of well separated gaussians with a better modeling of the underlying continuum (i.e., a better generative and discriminative properties) as shown later in the experiments.

## 4.2 Model Analysis & Settings

It can be noted that both expressions in Eqs. 5 and 6 rely on the *initial* means and covariances of data mapped by the NF. These terms originate from the likelihood loss, and are the exact solution when $\lambda = 0$. Optimizing this loss puts emphasis on relevant data continuum, and acts as an attracting force between gaussian pairs (see supplementary material). When $\lambda > 0$, the KLD term avoids overlapping class distributions leading (not only) to continuum modeling but also to class distribution separability. Its effect on the learned gaussians leads to (1) a repulsion of their centroids directed by the mean difference at the previous timestep (see Eq. 5), and also leads to (2) the deformation and shrinkage of the covariances in order to make classes well separated (as shown later in the experiments; see Fig. 1).
By balancing both these antagonist terms using FPI, enhanced discriminative and generative properties — and analytically more interpretable compared to gradient-based solutions — have been observed in our experiments. Additionally, one may show that covariance matrices in Eq. 6 remain symmetric positive definite (PD) provided that the initialization is also symmetric PD since the geometric mean of two PD matrices is PD [19], and this has also been observed experimentally for all the chosen $\lambda$ values (see again the supplementary material for more details).

**Asymptotic Behavior & Convergence.** As the impact of the KLD term decays exponentially, *viz.* tends to zero as gaussians become well separated, the asymptotic behavior of Eq. 5 and 6 can further be analyzed. Indeed, the fixed point of $\mu_{\mathbf{y}}^{(t)}$ becomes equivalent to class-wise means of training data mapped by the NF from the ambient to the latent space. As for the covariance, the expression becomes equivalent to Eq. 8 which is the identity $\Sigma_{\mathbf{y}}^{(t)} = \Sigma_{\mathbf{y}}^{(t-1)}$ provided that the covariance of data mapped by the NF to the latent space equates the covariance of the gaussian at timestep $t-1$. This asymptotic property is very interesting as it shows that the balance of both losses has been directly learned by the generative model, which then follows a more stable and convergent training process (see more details about convergence in the supplementary material).

$$\Sigma_{\mathbf{y}}^{(t)} = \left( \frac{1}{N_{\mathbf{y}}} \sum_{i=1}^{N_{\mathbf{y}}} (f(\mathbf{x}_i) - \mu_{\mathbf{y}}^{(t-1)})(f(\mathbf{x}_i) - \mu_{\mathbf{y}}^{(t-1)})^\top \cdot \left(\Sigma_{\mathbf{y}}^{(t-1)}\right)^{-1} \right)^{-1/2} \Sigma_{\mathbf{y}}^{(t-1)}. \tag{8}$$

**Control on $\pi_{KLD}$.** As noted previously, the function $\pi_{KLD}(.)$ is customizable, and can have a different impact on the overall KLD term. Most notably, by setting its value to the forward KLD of each pair $(\mathbf{y}, \mathbf{y}')$ as shown in Eq. 9

$$\pi_{KLD}(.) = \text{KLD}(\mathcal{N}_{\mathbf{y}} \,||\, \mathcal{N}_{\mathbf{y}'}), \tag{9}$$

the loss in Eq. 3 becomes equivalent to the negative log-likelihood of the pairwise KLD (instead of the exponential). We experimentally found that this variant converges with a much higher $\lambda$ than its averaged counterpart defined in Eq. 10 (which uses the mean of all pairwise KLDs), and leads to a better gaussian separation in the latent space, thus benefiting from better discriminative properties.

$$\pi_{KLD}(.) = \frac{1}{K(K-1)} \sum_{\mathbf{y},\mathbf{y}' \in \mathcal{Y}: \mathbf{y} \neq \mathbf{y}'} \text{KLD}(\mathcal{N}_{\mathbf{y}} \,||\, \mathcal{N}_{\mathbf{y}'}). \tag{10}$$

Furthermore, using only a single fixed point iteration at early training stages of the NF, a flexible and high quality latent space can be learned. This helps the NF converge much faster (because gaussian parameters no longer need to be trained at the subsequent iterations), and preserves a relevant continuum in the latent space as shown in experiments.

**Anisotropic Diagonal Covariances.** It is widely admitted that training fully dense covariance matrices in high dimensional spaces without enough data samples results in high instabilities and may also lead to overfitting [10, 14, 47]. Hence, we only consider anisotropic diagonal covariances as the best tradeoff between highly expressive (but overparametrized) dense covariances, and less expressive (underparametrized) isotropic ones.

# 5 Experiments

In this section, we study the performance of our proposed fixed-point iteration framework, by training NFs on the FPI based latent spaces, that are referred to as $NF^{\pi}$ (NFPI). We also compare it against other multimodal baselines described in section 5.2. Additionally, we train ConvNets and Transformers for classification using images generated with our model and also image augmentation. Finally, we show ablation study and closely related work comparisons.

## 5.1 Datasets and Evaluation Metrics

Experiments have been conducted using three standard image datasets: CIFAR10 [31], CIFAR100 [31] and ImageNet [11]. CIFAR10-100 include 50k images for training and 10k for testing while ImageNet includes 1281k for training and 50k for testing. The NF backbone used in our experiments is taken from the Generative Matrix Exponential Flow [52] which is a matrix exponential variant of affine coupling layers and invertible 1x1 convolutions built on top of Glow [29]. For image classification, we use a ResNet18 [24], a vision transformer adapted to mid-scale datasets from [20], and a very large pretrained transformer [15] from the timm [51] library. Classification performances are measured using accuracy as the percentage of correctly classified images. In the case of the NF, the classification accuracy (referred to as NF-Acc) is calculated with the labels obtained using $\mathbf{argmax}_{\mathbf{y}} P_{\mathbf{Z}}(\mathbf{z}|\mathbf{y})$ which corresponds to the gaussian with the highest likelihood, for a given sample in the latent space.

## 5.2 Baselines & Settings

**Baseline 1 (Exact Likelihood).** For the first baseline, gaussians are initialized with their classwise data mean and covariance, which is equivalent to the direct solution of the fixed point iteration without the KLD term in Eq. 4 (i.e., $\lambda = 0$). This baseline corresponds to Fig. 1(b), and it leverages the inductive bias in NFs in order to obtain an accurate initialization in the latent space.

**Baseline 2 (Gradient Descent).** For this baseline, we use gradient descent (GD) when optimizing Eq. 4: as gradient descent does not guarantee that the obtained anisotropic covariances are symmetric positive-definite, we guarantee this property by constraining the trained covariance eigenvalues to follow a sigmoid reparametrization, and the gradient of the loss 4 is rewritten as the product of the original gradient and the Jacobian of the sigmoid reparametrization. In this baseline, different learning rate policies are also investigated for comparison: sublinear, linear, and superlinear. In what follows, this baseline is dubbed as $NF^{GD}$. In view of space, more details about the settings of this baseline are reported in the supplementary material.

**Implementation Details.** The optimizer used to train the NFs is Adamax [28] with a learning rate of $10^{-2}$ linearly warmed up for the first 1000 iterations, and trained for 500 epochs using an exponential moving average[2]. The augmentations used are the same as in [2], i.e., horizontal flip, padding, cropping, rotation, and color jitter. The ResNet18 has 11.2M parameters, and it is trained with an SGD optimizer with a learning rate of $10^{-3}$, a momentum of 0.9, a weight decay of $5 \cdot 10^{-4}$, and a one cycle scheduler [46] with a maximum learning rate of 0.1. The batch size is 100, and the models are respectively trained for 30 and 50 epochs on CIFAR10 and CIFAR100, using cropping and horizontal flip as augmentations. For the mid-scale transformer, we use a Swin [35] architecture optimized for mid-scale datasets [20], with an Adam optimizer and a learning rate of $2 \cdot 10^{-3}$, a weight decay of $5 \cdot 10^{-2}$, and a cosine scheduler [36]. Besides horizontal flip and random cropping for image augmentation, we also use cutmix [56], mixup [57], auto-augment [8] and random augment [9] as well as random erasing [58]. The transformer has 7M parameters and is trained for 1000 epochs with a batch size of 256. The large pretrained transformer corresponds to EVA02 [15, 16] with 87.12M parameters pretrained on ImageNet21k [44] and then fine-tuned on ImageNet. SGD is used to train this transformer with a global gradient clipping at norm 1, a learning rate of $1 \cdot 10^{-5}$, and a batch size of 64 for a total of 3 epochs. Standard augmentations (SA) (com-

[2] implemented using the following github repository https://github.com/lucidrains/ema-pytorch

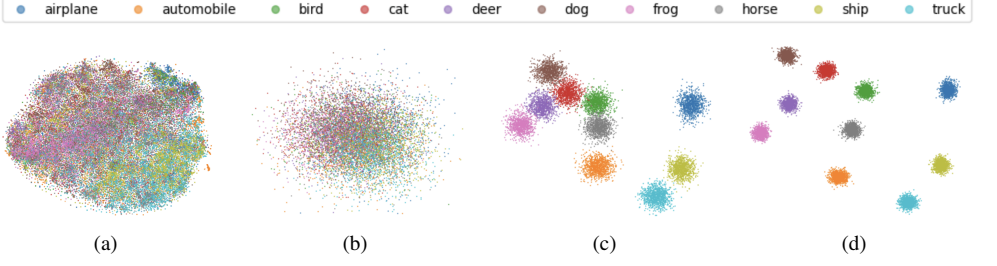
monly used in the state-of-the-art attention based models [34]) are also considered, including random augment and random erasing.

Table 1: Accuracy of different NF models where NF-Acc stand for the classification accuracy. It can be noted that $NF^{\pi}$ achieves the highest score, outperforming similar methods by up to 3 accuracy points on CIFAR100. More details concerning the learning rate policies can be found in the supplementary material.

| Model | CIFAR10 | CIFAR100 |
|---|---|---|
| | NF-Acc | NF-Acc |
| Baseline 1 (Data Initialization) | 77.31 | 39.16 |
| Baseline 2 (GD + Sublinear v1) | 92.34 | 68.28 |
| Baseline 2 (GD + Sublinear v2) | 91.94 | 60.68 |
| Baseline 2 (GD + Linear) | 93.1 | 60.58 |
| Baseline 2 (GD + Superlinear v1) | 92.41 | 57.21 |
| Baseline 2 (GD + Superlinear v2) | 93.25 | 58.56 |
| Our ($NF^{\pi}$) | **93.63** | **71.65** |

## 5.3 Results

**Model Analysis: Visualization.** Fig. 1 illustrates the behavior of FPI using TSNE visualization. Fig 1(a) shows all the embeddings obtained by projecting CIFAR10 images to a 2D space, whilst Fig. 1(b) shows samples from those gaussians (classwise estimated) which are highly overlapping. Figs. 1(c) and 1(d) show samples from gaussians obtained with FPI (following Proposition 1) using different $\lambda$ values. Small $\lambda$ values (see Fig. 1(c)) lead to smooth continuum whereas larger $\lambda$ (see Fig. 1(d)) result in well separated clusters in the latent space. For FPI, gaussians are initialized with the data means and covariances corresponding to Fig. 1(b). Note that upon convergence of FPI, the obtained gaussians are much more distinguishable than the initial ones.




(a) (b) (c) (d)

Figure 1: Fig. 1(a) shows 2D embeddings obtained by projecting CIFAR10 images to a lower dimensional space using TSNE. Fig. 1(b) shows samples from the classwise gaussians estimated from the embeddings in Fig. 1(a), and corresponds to the Baseline 1. Figs. 1(c) and 1(d) show samples from gaussians obtained using the proposed FPI, for small $\lambda$ values (Fig. 1(c)) and large $\lambda$ values (Fig. 1(d)). Although the FPI gaussians were initialized as in Fig. 1(b), they are much more dinstinguishable upon convergence of the FPI.

**Performances.** According to Table 1, $\text{NF}^{\pi}$ clearly outperforms $\text{NF}^{GD}$ for different learning rate policies considered (namely linear, two sublinear and superlinear variants – v1 and v2 – with different learning rate update policies; see more details in the supplementary material). On CIFAR100, the accuracy reaches 71.65% outperforming $\text{NF}^{GD}$ by more than 3 accuracy points. Additionally on CIFAR10, it reaches a high accuracy of 93.63%. Table 2 shows the accuracy of the NF together with the accuracy of the underlying CNNs (trained on top of the NF-generated images) for increasing values of $\lambda$; it's worth noticing that NF-acc measures the discriminative properties of our trained NF while CNN-acc measures their generative performances (i.e., to what extent the generated images by the NF, together with their conditioned labels, are fine for CNN training). When $\lambda$ is small, neither generative nor discriminative properties reach high values, and as $\lambda$ grows, a good balance is reached. Finally, as $\lambda$ keeps increasing, the generative properties are traded for discriminative ones,

| $\lambda$ | 10 | 50 | 100 | 200 | 300 | 500 |
|---|---|---|---|---|---|---|
| NF-Acc | 39.48 | 63.58 | 66.87 | 69.29 | **70.42** | 70.08 |
| CNN-Acc | 31.37 | 44.57 | **46.38** | 43.72 | 42.65 | 37.27 |

Table 2: Results for different values of $\lambda$ on CIFAR100.

| | CIFAR10 | | CIFAR100 | |
|---|---|---|---|---|
| Setting | NF-A | CNN-A | NF-A | CNN-A |
| #1: w/o KLD ($\lambda = 0$) | 77.31 | 62.1 | 39.16 | 23.03 |
| #2: w/ KLD on $\mu$ only | 93.2 | **74.76** | **71.65** | 37.02 |
| #3: w/ KLD on $\Sigma$ only | 88.75 | 61.94 | 57.76 | 30.58 |
| #4: w/ KLD on $\mu$ and $\Sigma$ | **93.63** | 73.03 | 70.42 | **42.65** |

Table 3: Ablation study of $NF^{\pi}$ when learning different KLD components (means, covariances, and both). All configurations initialize means and covariances using data in the latent space, excepting #2 which considers an identity covariance instead.

which eventually no longer increase.

**Ablations.** Table 3 shows an ablation study when means and covariances are trained separately and jointly. All means and covariances are initialized on data in the latent space excepting setting # 2 which uses isotropic identity covariances instead. Setting #2 confirms that it is indeed more important to learn the means. Learning the covariances only (setting #3) does not provide enough benefit to the CNN accuracy on CIFAR10. The rational resides in the fact that data means (at initialization) could be overlapping and gaussian separability cannot be achieved simply by shrinking the covariances, and this leads to erroneous labels when sampling images. On the other hand, learning the means provides enough flexibility to the learned gaussians in the latent space. When learning both means and covariances simultaneously (setting # 4), the highest accuracy is reached on CIFAR10, but not on CIFAR100, which is behind setting #2 that learns only the means. This could be explained by the larger number of classes and the reduced amount of training data per class on CIFAR100.

**SOTA Comparison.** We compare the results of $NF^{\pi}$ with other generative models used for classification (see Table 4). The proposed method is better than equivalent NFs using GMMs in the latent space (blue rows). It also matches the accuracy of Invertible ResNets [1, 5, 6, 37, 43] that constrain ResNet architectures to be invertible classifiers, and closes the gap with other generative models that do not preserve bijection (JEM++, SHOT-VAE), and are notoriously easier to train.

**Extra Comparison (Image Augmentation).** Finally, we further investigate the use of our NFs to train larger models, namely transformers [20], which suffer more from the lack of labeled data. In order to enhance the generalization performances of these transformers, we enrich training data by mapping the underlying images from the ambient to the latent space, and by disrupting the latent coordinates before mapping them back in the ambient space. This process implements linear (resp. nonlinear) data augmentation in the latent (resp. ambient) space. Results for CIFAR10-100 are available in Table 5, and results for ImageNet are

| Method | Accuracy CIFAR10 | Accuracy CIFAR100 | Model | GOA |
|---|---|---|---|---|
| $NF^{\pi}$ (ours) | **93.63** | **71.65** | NF | FPI |
| $NF^{GD}$ | 93.25 | 68.28 | NF | GD |
| IB-INN [1] | 91.28 | 66.22 | NF | GD |
| IB-INN + KLJ [43] | 88.6 | ✗ | NF | GD |
| FLOWGMM [33] | 88.44 | | NF | None |
| ULCGM [43] | 84.0 | | NF | GD |
| Monotone Flow [1] | 93.4 | ✗ | NF | None |
| i-DenseNets [43] | 92.40 | | NF | None |
| Residual NF [1] | 91.78 | | NF | None |
| Invertible ResNets [1] | 93.22 | **75.42** | NF | None |
| Implicit NF [33] | 92.71 | 70.94 | NF | Non |
| JEM++ [33] | **94.1** | 74.5 | EBM | GD |
| SHOT-VAE [33] | 93.89 | 74.70 | VAE | None |

Table 4: Comparison of $NF^{\pi}$ against closely related works. Blue rows shows gaussian based multimodal NFs. Excepting our method that uses fixed point iteration, other methods either use gradient descent, or fixed random initialization of gaussians. It can be noted that our method outperforms closest related works by more than 3 accuracy points on CIFAR100. The lower part of the table (white rows) shows other generative models used for classification. It should be noted that those NFs [1, 5, 6, 37, 43] are classifiers trained to be invertible, and not fit for conditional image generation. GOA stands for gaussian optimization algorithm.

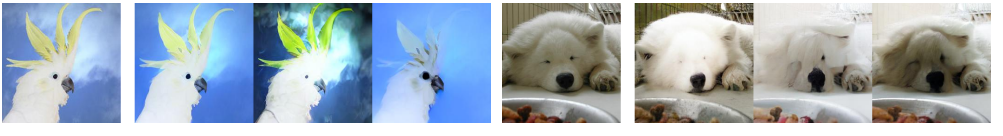
available in Table 6. In both cases, the proposed NF augmentations outperform the baseline, and show benefits when jointly used with other augmentations. Concerning the perturbation, we achieve a mixup on image pairs with identical labels, in the span of the PCA axes of their classwise gaussians, estimated on the underlying training data, in the latent space. We found this produces more visually diverse and realistic images, than simply adding noise in the latent space, since it takes into account their principal axes (i.e., modes with the highest variances). This is confirmed by the augmented images in Figure 2. Further results, visualizations and implementations details are available in the supplementary material.

Table 5: This table shows the impact of NF-based image augmentation on the accuracy of a trained transformer in [20]. When no data augmentation is used, accuracy increase of up to 1% is observed. When it is used jointly with other data augmentations, a small increase in accuracy is still observed.

| Model Used | P#(M) | CIFAR-10 | CIFAR-100 |
|---|---|---|---|
| Swin (scratch) | 7.1 | 93.37 | 77.32 |
| SL-Swin [63] | 10.2 | 94.93 | 79.99 |
| Swin-Drloc [63] | 7.7 | 86.07 | 65.32 |
| Swin Baseline [20] - w/o Augmentations | 7.1 | 93.06 | 72.84 |
| Swin Baseline [20] - w/o Augmentations + NF Augmentations (ours) | 7.1 | **94.06** | **73.50** |
| Swin Baseline [20] - w/ Augmentation | 7.1 | 96.89 | 80.38 |
| Swin Baseline [20] - w Augmentations + NF Augmentations (ours) | 7.1 | **96.96** | **80.45** |

| Top1 Acc | | | | Top5 Acc | | | |
|---|---|---|---|---|---|---|---|
| Baseline | SA | NFA (ours) | NFA + SA (ours) | Baseline | SA | NFA (ours) | NFA + SA (ours) |
| 90.896 | 90.986 | 91.031 | **91.082** | 98.802 | 98.834 | 98.849 | **98.854** |

Table 6: Comparison of our NF augmentations (NFA) against Standard Augmentations (SA), when fine-tuning a pretrained EVA02 transformer Baseline on ImageNet. We also combine NFA + SA, and this combination achieves the best results.



(a) Class 89 from ImageNet. (b) Class 258 from ImageNet.

Figure 2: Original Image from ImageNet (left), and 3 augmented variants (right), using the latent space of $NF^{\pi}$ when doing perturbations, for class 89 (a) and 258 (b) from ImageNet.

# 6 Conclusion

In this paper, we propose a novel method that learns multimodal gaussians as a part of NF training. The strength of the proposed method resides in its ability to learn more flexible and expressive distributions, well suited for conditional image generation which also balances between the discriminative and the generative properties of the resulting NFs. Our proposed solution relies on a fixed point iteration framework that (i) overcomes the tedious and sensitive process of tuning the learning rate policy, (ii) allows obtaining interpretable means and covariance matrices, and also (iii) preserves their symmetry and positive-definiteness. Extensive experiments conducted on different datasets show the positive impact of our method when training NF models for different classification and augmentation tasks involving convolutional networks and transformers.

**Acknowledgment.** This work has been supported with computer and storage resources by GENCI at IDRIS thanks to the grant 2024-AD011013954R1 on the supercomputer Jean Zay with the V100 and A100 partition.

# References

[1] Byeongkeun Ahn, Chiyoon Kim, Youngjoon Hong, and Hyunwoo J Kim. Invertible monotone operators for normalizing flows. *Advances in Neural Information Processing Systems*, 35:16836–16848, 2022.

[2] Lynton Ardizzone, Radek Mackowiak, Carsten Rother, and Ullrich Köthe. Training normalizing flows with the information bottleneck for competitive generative classification. *Advances in Neural Information Processing Systems*, 33:7828–7840, 2020.

[3] Andrei Atanov, Alexandra Volokhova, Arsenii Ashukha, Ivan Sosnovik, and Dmitry Vetrov. Semi-conditional normalizing flows for semi-supervised learning. *arXiv preprint arXiv:1905.00505*, 2019.

[4] Xingjian Bai and Luke Melas-Kyriazi. Fixed point diffusion models. *arXiv preprint arXiv:2401.08741*, 2024.

[5] Jens Behrmann, Will Grathwohl, Ricky TQ Chen, David Duvenaud, and Jörn-Henrik Jacobsen. Invertible residual networks. In *International conference on machine learning*, pages 573–582. PMLR, 2019.

[6] Ricky TQ Chen, Jens Behrmann, David K Duvenaud, and Jörn-Henrik Jacobsen. Residual flows for invertible generative modeling. *Advances in Neural Information Processing Systems*, 32, 2019.

[7] Sebastian Ciobanu. Mixtures of normalizing flows. In *Proceedings of ISCA 34th International Conference on*, volume 79, pages 82–90, 2021.

[8] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[10] Michael J Daniels and Robert E Kass. Shrinkage estimators for covariance matrices. *Biometrics*, 57(4):1173–1184, 2001. doi: 10.1111/j.0006-341x.2001.01173.x. URL https://doi.org/10.1111/j.0006-341x.2001.01173.x.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[12] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[13] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[14] David L Donoho, Matan Gavish, and Iain M Johnstone. Optimal shrinkage of eigenvalues in the spiked covariance model. *Annals of statistics*, 46(4):1742, 2018.

[15] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*, 2023.

[16] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19358–19369, 2023.

[17] Hao-Zhe Feng, Kezhi Kong, Minghao Chen, Tianye Zhang, Minfeng Zhu, and Wei Chen. Shot-vae: semi-supervised deep generative models with label-aware elbo approximations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 7413–7421, 2021.

[18] Ethan Fetaya, Jörn-Henrik Jacobsen, Will Grathwohl, and Richard Zemel. Understanding the limitations of conditional generative models. *arXiv preprint arXiv:1906.01171*, 2019.

[19] Luyining Gan and Sejong Kim. Revisit on spectral geometric mean. *Linear and Multilinear Algebra*, 72(6):944–955, 2024.

[20] Hanan Gani, Muzammal Naseer, and Mohammad Yaqub. How to train vision transformer on small-scale datasets? In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022. URL https://bmvc2022.mpi-inf.mpg.de/0731.pdf.

[21] Zhengyang Geng, Ashwini Pokle, and J Zico Kolter. One-step diffusion distillation via deep equilibrium models. *Advances in Neural Information Processing Systems*, 36, 2024.

[22] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[23] Paul Hagemann and Sebastian Neumayer. Stabilizing invertible neural networks using mixture models. *Inverse Problems*, 37(8):085002, 2021.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[25] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[26] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/ioffe15.html.

[27] Pavel Izmailov, Polina Kirichenko, Marc Finzi, and Andrew Gordon Wilson. Semi-supervised learning with normalizing flows. In *International Conference on Machine Learning*, pages 4615–4630. PMLR, 2020.

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[29] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

[30] Polina Kirichenko, Mehrdad Farajtabar, Dushyant Rao, Balaji Lakshminarayanan, Nir Levine, Ang Li, Huiyi Hu, Andrew Gordon Wilson, and Razvan Pascanu. Task-agnostic continual learning with hybrid probabilistic models. *arXiv preprint arXiv:2106.12772*, 2021.

[31] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images.(2009), 2009.

[32] Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. Vision transformer for small-size datasets. *arXiv preprint arXiv:2112.13492*, 2021.

[33] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco De Nadai. Efficient training of visual transformers with small datasets. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[34] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024.

[35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[36] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[37] Cheng Lu, Jianfei Chen, Chongxuan Li, Qiuhao Wang, and Jun Zhu. Implicit normalizing flows. *arXiv preprint arXiv:2103.09527*, 2021.

[38] Radek Mackowiak, Lynton Ardizzone, Ullrich Kothe, and Carsten Rother. Generative classifiers as a basis for trustworthy image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2971–2981, 2021.

[39] Barak Meiri, Dvir Samuel, Nir Darshan, Gal Chechik, Shai Avidan, and Rami Ben-Ari. Fixed-point inversion for text-to-image diffusion models. *arXiv preprint arXiv:2312.12540*, 2023.

[40] Lars Mescheder, Sebastian Nowozin, and Andreas Geiger. The numerics of gans. *Advances in neural information processing systems*, 30, 2017.

[41] Neel Mishra, Bamdev Mishra, Pratik Jawanpuria, and Pawan Kumar. A gauss-newton approach for min-max optimization in generative adversarial networks. *arXiv preprint arXiv:2404.07172*, 2024.

[42] Frank Nielsen. On the jensen–shannon symmetrization of distances relying on abstract means. *Entropy*, 21(5):485, 2019.

[43] Yura Perugachi-Diaz, Jakub Tomczak, and Sandjai Bhulai. Invertible densenets with concatenated lipswish. *Advances in Neural Information Processing Systems*, 34:17246–17257, 2021.

[44] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

[45] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[46] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019.

[47] Kun Song, Ruben Solozabal, Hao Li, Martin Takáč, Lu Ren, and Fakhri Karray. Robustly train normalizing flows via kl divergence regularization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(13):15047–15055, Mar. 2024. doi: 10.1609/aaai.v38i13.29426. URL https://ojs.aaai.org/index.php/AAAI/article/view/29426.

[48] Vincent Stimper, Bernhard Scholkopf, and Jose Miguel Hernandez-Lobato. Resampling Base Distributions of Normalizing Flows.

[49] Cédric Villani and American Mathematical Society. *Topics in Optimal Transportation*. Graduate studies in mathematics. American Mathematical Society, 2003. ISBN 9781470418045. URL https://books.google.fr/books?id=MyPjjgEACAAJ.

[50] Tianchun Wang, Farzaneh Mirzazadeh, Xiang Zhang, and Jie Chen. Gc-flow: A graph-based flow network for effective clustering. *arXiv preprint arXiv:2305.17284*, 2023.

[51] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[52] Changyi Xiao and Ligang Liu. Generative flows with matrix exponential. In *International Conference on Machine Learning*, pages 10452–10461. PMLR, 2020.

[53] Chen Xu, Xiuyuan Cheng, and Yao Xie. Invertible neural networks for graph prediction. *IEEE Journal on Selected Areas in Information Theory*, 3(3):454–467, 2022.

[54] Chen Xu, Xiuyuan Cheng, and Yao Xie. Normalizing flow neural networks by jko scheme. *Advances in Neural Information Processing Systems*, 36, 2024.

[55] Xiulong Yang and Shihao Ji. JEM++: Improved Techniques for Training JEM. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6474–6483. IEEE. ISBN 978-1-66542-812-5. doi: 10.1109/ICCV48922.2021.00643. URL https://ieeexplore.ieee.org/document/9710663/.

[56] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[57] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[58] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.