Learning Classwise Untangled Continuums for Conditional Normalizing Flows

Victor Enescu, Hichem Sahbi

Learning Classwise Untangled Continuums for Conditional Normalizing Flows

Victor Enescu and Hichem Sahbi

Sorbonne University, CNRS, LIP6, F-75005, Paris, France
victor.enescu@lip6.fr

Abstract. Normalizing flows (NFs) are invertible and bijective generative models, capable of performing exact density estimation of complex data by mapping them from highly nonlinear ambient spaces to simpler latent ones. These mappings hold many promises for images since capturing their true distribution could greatly enhance the performance of downstream tasks such as image classification. In this paper, we devise a novel conditional normalizing flow model that achieves both conditional image generation and classification. The main contribution of our method consists in learning untangled continuums of gaussian distributions in the latent space that maximize the discrimination power of the learned NFs together with the quality, diversity and label reliability of the underlying generated images. This results into highly effective NF classifiers as well as convolutional and transformer networks built on top of the generated images. Extensive experiments conducted on different challenging datasets, including CIFAR100 and ImageNet show the highly balanced discrimination and generative properties of our proposed NF models and their outperformance w.r.t. the closely related work.

Keywords: Generative Modeling · Normalizing Flow · Classification.

1 Introduction

Deep generative models are currently witnessing a major success in computer vision [44] and several neighboring fields [8,28]. These models allow capturing data distributions by learning mappings between *ambient* and *latent* spaces. 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. Amongst existing generative models, normalizing flows (NFs) [12,13,31] have particularly attracted a lot of attention due to their ability to exactly learn highly intricate distributions. Compared to alternative generative models [37,20,30,44], NFs are unique in their ability to learn bijective invertible transformations useful for *exact* density estimation and also image generation. Nonetheless, NFs in their standard form coerce the data, in the latent space, to follow monomodal gaussians, and this makes them powerless to condition image generation on class-labels. Other variants consider

instead gaussian mixture models (GMMs) to achieve both image generation and label conditioning. With these variants, it becomes possible to train not only generative but also discriminative models (see Fig. 1).

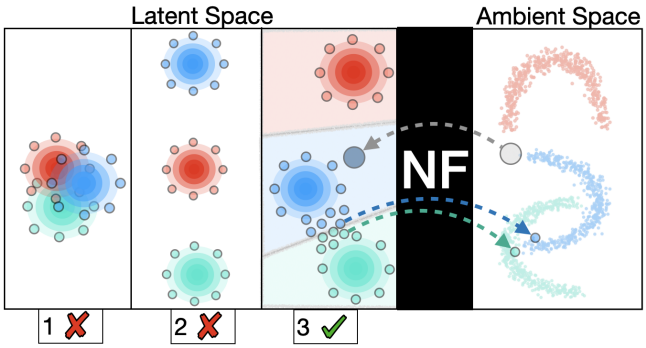


Fig. 1: This figure shows three different configurations of gaussians in the latent space. In (1), the gaussians preserve the continuum between classes and the generative properties only. In (2), gaussians are well separated and preserve discriminative properties only. In (3), both discriminative and generative properties of the gaussians are preserved. Image classification (resp. generation) is achieved by mapping data from the ambient to the latent space (resp. vice versa).

However, a few existing works questioned the effectiveness of training mixture models (MMs) with a plain likelihood loss for label conditioning [18,4,3,40], and conjectured that the latter, while conceptually attractive, may hinder the discriminative properties in favor of the generative abilities of the resulting NFs (see Fig. 1, config-1). On the other hand, directly constraining MM to include *a priori fixed* and *well separated* gaussians [26] is clearly suboptimal, and makes NFs discontinuous and less expressive (see Fig. 1, config-2). Alternatively, learning how to separate gaussians in the latent space can give very good discriminative properties and label conditioning (using, for instance, information bottleneck [3,40,52]), *but* at the expense of a significant degradation in the visual quality of the generated images, due to strong discontinuities and instabilities in the learned latent representations. Put differently, data belonging to visually similar classes could be mapped to distant (fixed or learned) gaussians in the latent space (and vice versa), and this (i) prevents the resulting NFs from consistently modeling the continuum between data across classes (see again Fig. 1), and (ii) ultimately leads to unusable generated images when training classifiers. In contrast to the aforementioned related work, our main contribution, in this paper, seeks to learn gaussians (hyperparameters) in the latent space while guaranteeing their separability and their ability to consistently model the continuum of data (see Fig. 1, config-3).

In this paper, we devise a novel approach that trains normalizing flows to-

gether with the hyperparameters of the underlying gaussian distributions. The proposed approach finds *an optimal placement* of gaussians (in the latent space) while balancing the generative and the discriminative properties of the learned NFs. This is achieved by optimizing an objective function which mixes a likelihood term that maximizes the expressivity (i.e., generative capacity) of the learned NFs, and a Kullback-Leibler divergence (KLD) criterion which enhances their discrimination power. Learning the hyperparameters consists in finding the best configuration of means and covariances associated to the gaussians. Whereas the optimization of gaussian means is straightforward and feasible with a direct application of gradient descent, the optimization of the covariance matrices requires additional constraints. This is achieved using a suitable reparametrization that constrains the learned covariance matrices to be symmetric and positive definite; by multiplying the Jacobian of the proposed reparametrization function with the gradient of the loss, one may guarantee that the learned covariance matrices are indeed symmetric and positive definite.

In order to assess our proposed NFs particularly against standard monomodal ones, we upgrade the latter with an encoding mechanism that conditions image generation with labels. As shown in experiments, the proposed NF model clearly outperforms standard ones, and other related work, using different metrics mainly classification accuracy as well as other proxies including image/label quality and diversity. These proxies allow us to further analyze the behavior of the proposed NFs and to understand why a better accuracy is reached. To the best of our knowledge, this is the first comprehensive study of the impact of image/label quality as well as diversity — *all together* — on classification performances, when using normalizing flows.

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

(i) A novel method, in section 4, that learns NFs together with the underlying gaussian placements. Our method seeks to maximize the discrimination power of our NFs by *bringing closer gaussians belonging to visually similar classes (and vice versa)* while also maximizing the generative capacity of our NF models.
(ii) A novel encoding mechanism, in section 5.2, that allows conditionally generating data from different NFs (including standard ones). This encoding allows training NFs not only to generate images but also their labels.
(iii) A comprehensive study and comparison of our NF models in section 5 — using staple metrics (namely accuracy) as well as proxy ones including image/label quality and diversity — show the competitiveness of the proposed NF models against the related work.

2 Related Work

Normalizing Flows & Split Priors. One of the earliest normalizing flow models, namely Glow [31], is based on the multi-scale architecture of RealNVP [13]. It includes several levels built upon invertible neural networks such as *actnorms*

[31], *invertible 1×1 convolutions* [31], and *coupling layers* [12]. At each level, part of the dimensions are removed, using split priors [31], which estimate the gaussian hyperparameters (means and covariances) with a neural network. Many existing works have implemented conditioning on top of these architectures, including [31,18,4] which learn a conditional prior at the lowest level using a classification loss. In particular, [18,31,50] also use split priors to estimate gaussian hyperparameters through different mini-batches. However, the resulting gaussians are very likely to overlap, and this makes label-conditioning challenging. Our alternative solution, in this paper, does not rely on split priors but instead on a novel reparametrization that allows training more suitable gaussian hyperparameters for a better NF conditioning.

Label-Conditioning with GMMs. Mixture models condition image generation in NFs [26,32,4,18,3,40,21,54,10,59,60] by assigning a unique component (gaussian mean and covariance) to each label. Existing variants [4,18] condition only some dimensions of GMMs, with the idea that only a portion of the dimensions are discriminative. Other methods [26,32,21,54,10] condition all dimensions by breaking the gaussian distribution into multiple ones, and fixing the underlying means to random values, and covariance matrices to be diagonal isotropic; in particular, [26] further recalibrate those matrices on validation sets. The works in [7,33] consider handcrafted means and learn different NFs through clusters in the latent space with limited generalization performances [7]. Alternative solutions [3,40,59,60] learn instead the means but fix the covariance matrices to identity using information bottleneck [3,40], or by adding other losses including Wasserstein [59,60]. The works in [26,18,3,40] are the most related to our proposed method with the differences being that (i) gaussian means are learned using a repulsive pairwise KLD criterion in contrast to [18,26], and (ii) the covariance matrices are anisotropic, thereby more expressive compared to [3,40,26,60,59]. Finally, our proposed method does not leverage GMMs but instead multi-gaussians, and this reduces the number of training parameters, and makes the joint optimization of NFs and gaussians more tractable.

Label-Conditioning with Encoding. Another category of methods focuses on directly conditioning images in ambient and latent spaces. Xiao et al. [58] trains NFs on latent representations of VAEs, and concatenates one hot encoding vectors to the latent samples. Ardizzone et al. [2] use a one hot encoding for conditioning, but rely on a complex loss that does not scale well to high dimensional data. The methods in [43,58] directly condition the coupling Layers by concatenating a higher dimensional embedding conditioned on labels issued by a neural network. In contrast to these methods, our proposed conditional encoding is *only* applied to the ambient space, and neither *enforced* in the latent space *nor* in the NF architecture. Put differently, it acts as a preprocessing step that encodes labels in images, and allows NFs to learn not only images, but also label generation. Our proposed encoding is also related to augmented normalizing flows [24] that make a model more expressive by embedding data in high dimensional spaces using noise sampled from a normal distribution. Instead of noise, we consider a learned class-dependent conditional encoding.

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 [31,12], each flow is usually made of an actnorm layer, an invertible 1×1 convolution, and a coupling layer stacked together. Let $\mathbf{x}_{1:d}$ be a d-dimensional vector, 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 1×1 convolutions are generalized permutation layers that enhance expressivity by allowing permutations between image channels to be learned [31]. An actnorm layer is an invertible equivalent of batch normalization [25] that increases stability and performance. Multi-scale architectures [13] are also used in NFs and allow better expressivity by progressively removing half of the dimensions through different flows; in other words, a NF (made of L levels) discards and appends (at the end of each level) half of the dimensions to the output of the normalizing function f, in order to extract more meaningful intermediate representations from the remaining half. NFs are usually trained to minimize the negative log-likelihood of Eq. 1. From transport theory point of view [53], 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 $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_i \subset \mathcal{X} \times \mathcal{Y}$ denote a collection of labeled images with $\mathbf{x}_i$ belonging to an ambient space $\mathcal{X}$ and $\mathbf{y}_i$ its underlying class-label 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

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

here $P_{\mathbf{Z}}(.|\mathbf{y})$ is set a priori to a given distribution, *viz.*, gaussian mixture. Our goal here is to train the parameters of the NF (denoted as Θ) together with the

hyperparameters of the underlying gaussians (referred to as $\Psi = \{(\mu_{\mathbf{y}}, \Sigma_{\mathbf{y}})\}_{\mathbf{y}\in\mathcal{Y}}$) while guaranteeing better generation and classification performances of the resulting NF.

4.1 Learning Continuums

By assigning different gaussians across classes, one may train the parameters Θ together with the hyperparameters Ψ using the log-likelihood loss of Eq. 2

$$\mathcal{L}_{NF}(\Theta, \Psi) = \sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} -\log \mathcal{N}(f(\mathbf{x}); \mu_{\mathbf{y}}, \Sigma_{\mathbf{y}}) - \log \left|\det \mathbf{J}_{f(\mathbf{x})}\right|. \tag{3}$$

Following Jensen's inequality, Eq. 3 is an upper bound of the loss widely used in GMM-based NFs [26,18,3] — provided that GMM components (gaussians) are well separated. This form has several advantages; on the one hand, it is more flexible compared to monomodal gaussian NFs which are class-oblivious. On the other hand, while gathering the upsides of GMM based NFs (mainly conditional image generation), it is more tractable and also convex when only means (or covariances) are allowed to vary during training. Our proposed loss in Eq. 3 relies on multiple gaussians with a constrained one-to-one mapping between classes and gaussians. Our goal is to make conditional generation providing (i) *visually plausible* images and (ii) *accurate labels* on the generated images. As shown subsequently (and also in proposition 1 in the supplementary material), the optimization of Eq. 3 alone guarantees condition (i), i.e., a *smooth* placement of gaussians forming a continuum of gradually similar data in the latent space. In other words, as we traverse this continuum, images vary smoothly across classes thereby making images generated with this conditioning visually plausible (see later experiments). Considering this issue, our introduced proposition (in the supplementary material) shows that only condition (i) is achieved when optimizing Eq. 3 at the detriment of condition (ii). This results into highly confounded gaussians which makes label conditioning error-prone.

4.2 Learning Untangled Continuums

In order to mitigate the aforementioned issue (ii), we also consider a Kullback-Leibler Divergence (KLD) term as a *repulsion criterion* between different gaussians that aims at pushing them apart from each others. This term equates

$$\mathcal{L}_{KLD}(\Psi) = -\sum_{\mathbf{y},\mathbf{y}'\in\mathcal{Y}:\mathbf{y}\neq\mathbf{y}'} \mathrm{KLD}(\mathcal{N}_{\mathbf{y}} \,||\, \mathcal{N}_{\mathbf{y}'}). \tag{4}$$

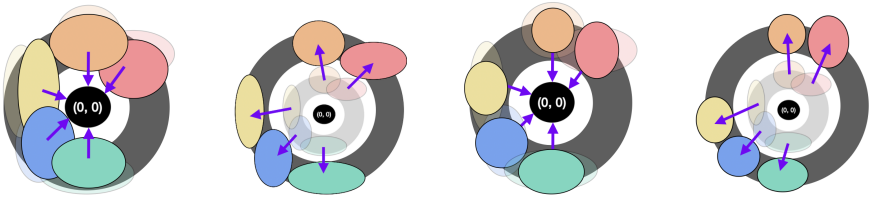
With the above term, we define our global loss as

$$\mathcal{L}(\Theta, \Psi) = \mathcal{L}_{NF}(\Theta, \Psi) + \lambda \mathcal{L}_{KLD}(\Psi), \tag{5}$$

where $\lambda \geq 0$ controls the impact of KLD. On the one hand, large λ makes gaussians well separated, and thereby label conditioning more accurate, however, the continuum between visually similar classes will not be correctly modeled resulting into abrupt changes between these classes. On the other hand, small λ makes

transition between classes smoother, and image visually plausible, but label conditioning becomes erroneous. Hence, the setting of λ is critical and should be carefully achieved in order to balance image quality and label accuracy. In practice, λ (rewritten as λ_t) is annealed through iterations $t \in \{0, \dots, \textbf{maxepoch}\}$; initially, λ_0 is overestimated to favor the discrimination power of the trained NF, and then λ_t is gradually decreased to put more emphasis on its generation properties, which also improves training stability.

Following proposition 1 in the supplementary material, and as shown in Fig. 2 and later in ablation, the two losses $\mathcal{L}_{NF}$ and $\mathcal{L}_{KLD}$ in equation 5 are complementary and show different behaviors. $\mathcal{L}_{NF}$ naturally places the means of similar classes close one to another, and it also expands the covariance matrices in order to capture partial overlaps between classes sharing similar visual aspects. Fig. 2a shows an illustration of the partial overlap of 2D gaussians using $\mathcal{L}_{NF}$ only. When $\mathcal{L}_{KLD}$ is added, and when optimizing only the means, the distance between gaussians is increasing particularly between visually different classes (see Fig. 2b); this separability cannot be obtained when using $\mathcal{L}_{NF}$ only. On another hand, optimizing only the covariance matrices shrinks the gaussians at intersecting areas which makes them separated without moving the means (see Fig. 2c). Finally, when optimizing both the means and the covariances, the model reaches a more optimal design allowing both the means and the covariances to change, and yielding a continuum between gaussians without overlapping (see Fig. 2d). In sum, the whole model balances two *antagonist behaviors*, associated to the NF and the KLD losses, that respectively attract and repulse the trained gaussians.



(a) $\mathcal{L}_{NF}$ loss only (b) $\mathcal{L}^{\mu}_{NF}$ + $\mathcal{L}^{\mu}_{KLD}$ loss (c) $\mathcal{L}^{\Sigma}_{NF}$ + $\mathcal{L}^{\Sigma}_{KLD}$ loss (d) $\mathcal{L}_{NF}$ + $\mathcal{L}_{KLD}$ loss

Fig. 2: Influence of the losses $\mathcal{L}_{NF}$ and $\mathcal{L}_{KLD}$ on the trained gaussians.

4.3 Optimization

Training NF parameters together with gaussian hyperparameters, using Eq. 5, is achieved by backpropagation and gradient descent of Θ and Ψ. In practice, the covariance matrices are taken as diagonal anisotropic. Whereas this setting makes the gaussians relatively flexible (compared to isotropic ones) and their training more tractable (w.r.t. fully dense gaussians), optimizing Ψ still requires

additional constraints to guarantee that the resulting $\{\Sigma_{\mathbf{y}}\}_{\mathbf{y}}$ are positive definite. In order to implement these constraints, we consider a reparametrization in Eq. 5 — particularly the covariance matrices — as $\Sigma_{\mathbf{y}} = \psi(\hat{\Sigma}_{\mathbf{y}})$ for some $\hat{\Sigma}_{\mathbf{y}} \in \mathbb{R}^{d\times d}$ with ψ applied entrywise, and this allows free settings of $\{\hat{\Sigma}_{\mathbf{y}}\}_{\mathbf{y}}$ during optimization while guaranteeing the positive definiteness of $\Sigma_{\mathbf{y}}$. During backpropagation, the gradient of the loss $\mathcal{L}$ (now w.r.t. $\hat{\Sigma}_{\mathbf{y}}$) is updated using the chain rule as

$$\frac{\partial \mathcal{L}}{\partial \mathbf{vec}(\hat{\Sigma}_{\mathbf{y}})} = \mathbf{J}_{\psi} . \frac{\partial \mathcal{L}}{\partial \mathbf{vec}(\Sigma_{\mathbf{y}})}, \tag{6}$$

here $\mathbf{vec}(\Sigma_{\mathbf{y}})$ is a columnwise vectorization of $\Sigma_{\mathbf{y}}$ and $\mathbf{J}_{\psi}$ is a diagonal Jacobian whose i^{th} diagonal element equates $\psi'([\hat{\Sigma}_{\mathbf{y}}]_{ii})$. In practice, $\psi(.) = a(1 + \exp\{-\beta(.)\})^{-1} + c$ with a, b and c being positive values that respectively control the amplitude (scale) and the slope (smoothness) as well as the shift of the reparametrization ψ. Besides, $\frac{a+c}{c}$ controls the conditioning of the trained covariance matrices and thereby the shape of the underlying gaussians in the latent space (see Fig. 3-top).

Algorithm 1: GLEM

Input: Images with labels in $\mathcal{D} = \mathcal{D}_{\text{nf}} \cup \mathcal{D}_{\text{g}}$.
Output: NF parameters Θ and gaussian hyperparameters Ψ.
Initialization: Set $\{\mu_{\mathbf{y}}\}_{\mathbf{y}}$, $\{\hat{\Sigma}_{\mathbf{y}}\}$ entries to zeros ; *// Equivalently, each $\Sigma_{\mathbf{y}}$ is set to $I_d \cdot (\frac{a}{2} + c)$; this corresponds to (overlapping) standard monomodal and isotropic gaussian initializations which become well separated and anisotropic as training evolves.*
for $t := 1$ **to** *MaxIterations* **do**
  Fix Ψ and train NF parameters Θ on $\mathcal{D}_{\text{nf}}$; `// E-step`
  Fix Θ and train gaussian hyperparameters Ψ on $\mathcal{D}_{\text{g}}$; `// M-step`
  Compute the gradients $\{(\frac{\partial \mathcal{L}}{\partial \mu_{\mathbf{y}}}, \frac{\partial \mathcal{L}}{\partial \mathbf{vec}(\Sigma_{\mathbf{y}})})\}_{\mathbf{y}\in\mathcal{Y}}$;
  For each $\mathbf{y} \in \mathcal{Y}$
    $\mu_{\mathbf{y}} \leftarrow \mu_{\mathbf{y}} - \delta_t \frac{\partial \mathcal{L}}{\partial \mu_{\mathbf{y}}}$;
    $\hat{\Sigma}_{\mathbf{y}} \leftarrow \hat{\Sigma}_{\mathbf{y}} - \delta_t \mathbf{vec}^{-1}\left(\frac{\partial \mathcal{L}}{\partial \mathbf{vec}(\hat{\Sigma}_{\mathbf{y}})}\right)$ as shown in Eq. 6;
    Update $\Sigma_{\mathbf{y}} \leftarrow \psi(\hat{\Sigma}_{\mathbf{y}})$;
  Update the learning rate δ_t and λ_t.
With optimized Ψ^*, retrain the NF parameters Θ on $\mathcal{D}_{\text{nf}} \cup \mathcal{D}_{\text{g}}$.

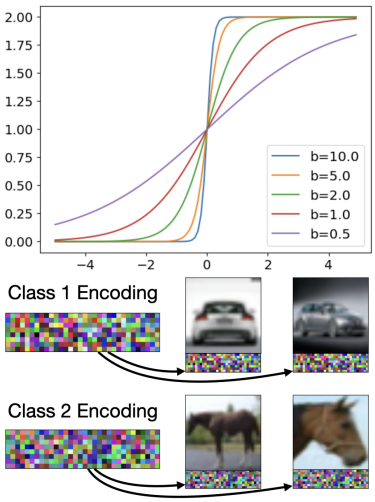


Fig. 3: (Top) Visualization of the reparametrization for varying values of $b \in \{0.05, 0.1, 0.2, 0.5, 1.0\}$, with $a = 2$ and $c = 0$. (Bottom) Class conditional encoding concatenated with images.

In order to optimize the loss in Eq. 5, we consider an EM-like procedure. Two steps are alternately applied: in the E-step, we fix the hyperparameters Ψ, and we train our NF while in the M-step we fix the NF parameters Θ and we train only Ψ using gradient descent. These two steps are run using two disjoint subsets $\mathcal{D}_{\text{nf}}$ and $\mathcal{D}_{\text{g}}$ (taken from $\mathcal{D}$) in order to mitigate the co-adaptation between Θ and Ψ. In practice, $|\mathcal{D}_{\text{g}}| = 0.1 \times |\mathcal{D}|$ and $|\mathcal{D}_{\text{nf}}| = 0.9 \times |\mathcal{D}|$. Algorithm 1, dubbed as

GLEM (Gaussians Learned using Expectation Maximization), shows the whole optimization procedure.

5 Experiments

In this section, we study the impact of GLEM on image classification when used separately and jointly with ConvNets and Transformers. First, we introduce different monomodal and multimodal NF baselines. Then, we extend the former using a novel mechanism that conditions image generation. We also study the impact of KLD weighting and sampling temperature on GLEM. Finally, we show an ablation study and comparison against closely related works.

5.1 Datasets and Evaluation Metrics

Experiments have been conducted using three standard datasets: CIFAR10 [34], CIFAR100 [34] and later ImageNet [11]. CIFAR10-100 include 50k images for training, and 10k images for testing, whereas ImageNet includes 1281k for training and 50k for testing. The NF backbone used in our experiments is taken from the Generative Matrix Exponential [57] which is a generalization of affine coupling layers built on top of Glow, while for image classification, we use ResNet18[22] and a vision transformer [15,51] from the timm library. Classification performances are measured using accuracy as the percentage of correctly classified images. However, to further understand the behavior of our model, we use three other proxy metrics namely FID [23], Coverage [41], and Label Reliability which respectively assess the visual quality of images, their diversity (as an improved variant of the recall [45,35] used in generative modeling) and the accuracy of their labels measured by a simulated oracle. The latter corresponds to a highly accurate ResNet18 trained on CIFAR100. We also use this oracle — precisely its penultimate layer — in order to assess the FID and the Coverage (more details are available in the supplementary material).

5.2 Baselines & Model Analysis

Baseline 1 (Monomodal NFs with CE). We upgrade standard monomodal NFs [57] with a conditional encoding (CE) mechanism that allows, *not only*, sampling images but also predicting their class-labels. Our CE is achieved as a part of NF training; each image $\mathbf{x}$ (used to train the NF) is padded with a class-dependent code (subimage) along its vertical axis, so the height of $\mathbf{x}$ increases by 2^L (where L is the number of NF-levels) while its width and depth remain identical (see Fig. 3-bottom). Ground-truth codes assigned to training images are obtained by first designing class-dependent prototypes whose placements are obtained by minimizing a regularized mean squared error, and then by perturbing these prototypes with a uniformly generated noise whose amplitude is smaller compared to the designed prototypes.

Baseline 2 (Multimodal NFs). This setting corresponds to GLEM (without

CE) whose conditioning is based on gaussians trained with $\lambda = 0$.
Baseline 3 (CEGLEM). This setting combines both CE and GLEM, and consists in learning NFs and gaussians on images endowed with class-dependent codes. This variant allows a double conditioning mechanism; in practice, we found that both — CE and gaussian conditioning — provide identical class-labels during image generation.
Ours (GLEM). This setting corresponds to GLEM with multi-gaussians trained by optimizing both the likelihood and the KLD losses (i.e., $\lambda > 0$).

Implementation Details. The optimizer used to train the NFs is Adamax [29] with a learning rate of 10^{-2} linearly warmed up for the first 1000 iterations, and divided by a factor 2 at several intervals — until it reaches a minimum of $2.5 \cdot 10^{-3}$ where stochastic weight averaging is also used [27]. The augmentations used are the same as in [3], i.e., horizontal flip, padding, cropping, rotation, and color jitter. For ResNet18, the optimizer used is SGD 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 [47] with a maximum learning rate of 0.1. The augmentations used are cropping and horizontal flip. For the transformers, a cosine scheduler [38] with an Adam optimizer is used; we use standard augmentations for small scale transformers (see [19] for extra details). More implementation details can be found in the supplementary material. Table

Table 1: Table Accuracy of different models on CIFAR100. NF-A is the classification accuracy of NFs, and CNN-A is the accuracy of Resnet18 trained on the generated images.

Model	CIFAR10		CIFAR100	
	NF-A	CNN-A	NF-A	CNN-A
Baseline 1 (CE)	✗	75.54	✗	37.40
Baseline 2 (GLEM, $\lambda = 0$)	57.83	60.17	19.12	26.7
Baseline 3 (CEGLEM)	12.82	76.03	2.04	38.85
Our (GLEM)	**92.02**	**77.39**	**67.67**	**45.22**

Table 2: Comparison of different methods on CIFAR100 with the selected metrics. According to these results, GLEM overtakes the baselines.

Method \ Metric	NF-A ↑	CNN-A ↑	FID ↓	Cov ↑	L-R ↑
Baseline 1 (CE)	✗	37.40	37.62	17.70	12.58
Baseline 2 (GLEM, $\lambda = 0$)	19.12	26.7	25.68	19.77	9.65
Baseline 3.1 (CEGLEM $\lambda_0 = 3$)	2.23	39.40	27.34	17.83	16.38
Baseline 3.2 (CEGLEM $\lambda_0 = 10$)	2.04	38.85	26.38	17.97	16.25
Our (GLEM $\lambda_0 = 3$)	63.40	**45.95**	**19.96**	**25.93**	**38.36**
Our (GLEM $\lambda_0 = 10$)	**67.67**	45.22	26.33	19.15	32.51

Performances. According to Table 1, GLEM outperforms the three aforementioned baselines, as the gaussians are well separated, and this provides better image quality (see also Table 2) and more accurate labels; this is challenging to achieve with baseline 1 as classes are totally intricate in the latent space. Note that CEGLEM cannot be used alone (without CNNs [36]) for classification, as test images should be appended with class-dependent codes which are unknown on test data; furthermore, appending irrelevant (e.g., random) code values creates a domain shift between training and test data which significantly degrades classification performances (as displayed in Table 1). According to these results, GLEM clearly outperforms all other baselines, which is corroborated by the proxy metrics in Table 2. Subsequently, we study the behavior of our GLEM model in the remaining experiments, mostly on the challenging dataset CIFAR100.

KLD Weighting. Table 3 shows the increasingly positive impact of GLEM when λ takes sufficiently (but not very) large values. For small λ values, gaussians are still intricate and this results into low classification performances. Conversely, when λ is large, gaussians are highly separated but classification performances decrease due to discontinuities in the learned gaussians (as shown through FID scores). The proxy metrics show low visual quality (high FID), low coverage (diversity), and also low label reliability when λ takes extreme values. By taking into account those observations, gaussians that are closer to each other (but *"separated enough"*) produce images of higher quality (for image classification) than highly separated gaussians.

Table 3: Behavior of GLEM for different mixing coefficients λ_0 of $\mathcal{L}_{KLD}$ on CIFAR100. BPD is a metric that stands for bits per dimension, and is commonly used to see how well an NF has learned.

Method	KLD for different λ_0 values					
	1	3	5	10	15	20
NF-A ↑	26.45	63.40	66.53	67.67	**68.23**	67.39
CNN-A ↑	30.95	45.95	**47.21**	45.22	43.51	42.85
FID ↓	21.54	**19.96**	23.71	26.33	31.36	35.83
Cov ↑	23.86	**25.93**	22.0	19.14	16.17	14.35
L-R ↑	21.04	**38.36**	35.39	32.52	30.10	24.66
BPD (5bits) ↓	**1.64**	1.74	1.87	2.07	2.19	2.25

Table 4: Accuracy of CNN (ResNet18) w.r.t. the number of NF generated training images (see extra details in the supplementary material).

Sizes	1×	2×	5×	10×	20×	50×	100×
CIFAR10	81.02	82.48	84.25	84.45	85.77	86.59	**87.14**
CIFAR100	55.21	57.66	59.47	61.11	61.84	63.42	**63.52**

Table 5: Metrics obtained at different sampling temperatures and RFCM (Reestimated Full Covariance Matrix) on CIFAR100. RFCM is obtained by projecting training data belonging to a given class $\mathbf{y}$ and reestimating its mean $\mu_{\mathbf{y}}$ and its full covariance matrix $\Sigma_{\mathbf{y}}$. Note that RFCM is achieved at the end of the EM training procedure shown in Algorithm 1. RFCM clearly makes covariance matrices more expressive and thereby yields globally better accuracy and proxy metric performances. The reparametrization factors used are $a = 0.5$, $b = 0.2$ and $c = 0.05$.

τ	1.2	1.1	1.0	0.9	0.8	0.7	RFCM
CNN-A ↑	42.99	43.63	44.64	**45.22**	44.99	44.17	**49.74**
FID ↓	30.15	28.17	26.90	**26.33**	26.45	27.21	**25.55**
Cov ↑	16.13	17.21	18.47	19.15	**19.33**	19.19	**22.47**
L-R ↑	26.82	28.92	30.86	32.51	33.67	**34.69**	33.56

Sampling. Gaussians in the latent space are sampled within a range τ w.r.t. their centroids; this range τ is dubbed as *temperature*. This is obtained by rescaling the covariance matrices by τ prior to sampling (as introduced in [31] for NFs). Large values of τ imply diverse samples but visually less plausible, and vice versa. Table 5 shows the impact of τ on the classification accuracy as well as the other proxy metrics. As τ gets sufficiently (but not very) small, classification performances improve, however, smaller values of τ degrade classification performances since the quality and the diversity of the generated samples degrades too (see FID and Coverage); this clearly shows that the actual distribution of plausible and diverse samples do not lie near the gaussian centers, but instead in a mid-distance between the centers and the extent of the gaussian (i.e., temperature). Nonetheless, label reliability continues to improve since samples close to the gaussian centroids have the least uncertain labels.

Cardinality. All classes are equally sampled to construct artificial datasets whose sizes are multiples of the size of the original set. As shown in Table 4, the

accuracy of the CNNs — trained on top of the generated samples — improve significantly as more images are sampled from the NFs.

Table 6: This table shows an ablation study on CIFAR100, where different criteria (KLD term, Rep: reparametrization, and RFCM) are gradually added. From these results, we observe a high impact of KLD in config #2 (better gaussian separability). When adding the reparametrization (configs #3 and #4), we observe a lower FID (better visual quality), higher coverage/Cov (better diversity) and higher L-R (better Label Reliability) particularly when λ is sufficiently (but not very) large (i.e., $\lambda = 3$) suggesting that a better continuum and discrimination power are reached when λ is appropriately set.

Setting	NF-A ↑	CNN-A ↑	FID ↓	Cov. ↑	L-R ↑
#1: w/o KLD/Rep/RFCM	19.12	26.7	25.68	19.78	9.66
#2: w $\text{KLD}_{\lambda_0} = 10$	38.21	27.87	**22.96**	21.96	24.59
#3: w $\text{KLD}_{\lambda_0} = 10$ + Rep	**67.67**	45.22	26.33	19.15	32.51
#4: w $\text{KLD}_{\lambda_0} = 3$ + Rep	63.40	45.95	**19.96**	**25.93**	**38.36**
#5: w $\text{KLD}_{\lambda_0} = 10$ + Rep + RFCM	67.50	**49.74**	25.55	**22.47**	**33.56**

Table 7: Ablation study on CIFAR100 w.r.t. the learned gaussian hyperparameters. Bold and red scores respectively show the best and the second best values in the table. The best (overall) accuracy scores are reached when learning both μ and Σ. If only Σ is learned, the NF reaches a higher accuracy, but both the continuum between classes (proxy metrics) and CNN accuracy are degraded. In these experiments, a, b and c of the reparametrization are respectively set to 5.0, 0.2, 0.05 which leads to a better observed trade-off on all the used metrics. More details can be found in the supplementary material.

Setting	NF-A ↑	CNN-A ↑	FID ↓	Cov. ↑	L-R ↑
Only μ ($\lambda_0 = 10$)	64.31	45.29	**20.29**	**25.74**	35.70
Only Σ ($\lambda_0 = 10$)	**66.91**	41.16	36.02	15.76	30.18
Both μ and Σ ($\lambda_0 = 10$)	66.49	**46.12**	21.43	24.39	**36.38**
Random Initialization of μ and Σ	59.9	43.34	25.0	18.6	25.32

5.3 Ablation

Table 6 shows an ablation study and impact of different criteria used in GLEM. From these results, we observe that KLD improves accuracy of NFs (and the underlying CNNs) with a clear margin. We observe a similar behavior on the proxy metrics (FID and Coverage as well as Label Reliability). Reparametrization brings an extra substantial gain to accuracy and label reliability, but not to FID and Coverage when λ is overestimated; this is mainly due to high gaussian separability which *breaks the continuum* (and thereby visual quality and diversity) between the underlying classes. In contrast, a reasonably large λ value (equal to 3) maintains comparable accuracy while improving significantly all the proxy metrics. Note that without reparametrization, negative covariance matrix entries are rounded — after each gradient descent step — to a lower bound 10 times smaller than the initial positive values of these entries (to avoid bad conditioning). Table 6 also shows the impact of RFCM which globally makes GLEM more effective. Finally, Table 7 shows an ablation study when μ and Σ are separately and jointly learned. We observe that the best trade-off between NF/CNN accuracy and the proxy metrics is reached when both μ and Σ are jointly learned.

5.4 Extra Comparison

We compare the results of GLEM with other generative models used for classification (see Table 8). The proposed method is better than equivalent NFs using GMMs in the latent space (blue rows). Furthermore, it can train classifiers reaching higher accuracy (using artificial images) beside producing more meaningful interpolations in the latent space (see Figure 4). It also matches the accuracy of Invertible ResNets [6,9,39] that constrain ResNet architectures to be invertible classifiers, or other generative models that do not preserve bijection (JEM++, SHOT-VAE). To the best of our knowledge, our proposed method is also the first to study the quality of both labels and images generated with conditional NFs, using the three proxy metrics, together with the accuracy of the underlying classifiers. The classifiers trained using artificial images (see lower part of Table 8) are as good as GANs, and unlike the latter, NFs can sample more images while continuously increasing those scores, as they do not suffer from a lack of diversity.

We further investigate the use of GLEM for image augmentation and classification , by finetuning very large transformers models on ImageNet [11] dataset, reaching enhanced generalization performances. To do so, we enrich training data by mapping the underlying images from the ambient to the latent spaces, and by disrupting the latent coordinates along principal modes using mixup operations. Afterwards, the resulting disrupted latent representations are mapped back to the ambient space using the NF generation function. This process implements linear (resp. nonlinear) data augmentation in the latent (resp. ambient) space. Table 9 shows classification results when finetuning pretrained transformers from timm [55] library. We selected the most performant transformer with less than 100M parameters (an EVA02 [16,15]) that was originally ranked number 9 on top 1 accuracy, and fine-tuned it on the NF-based augmented images. Figure 5 shows some interpolations, and Figure 6 shows some augmented images; additional details and experiments can be found in the supplementary material.

Fig. 4: Interpolations obtained using GLEM (top) and IB-INN [3] (bottom, taken from the paper) respectively reaching a NF-A of 91.88% and 91.28% on CIFAR10.

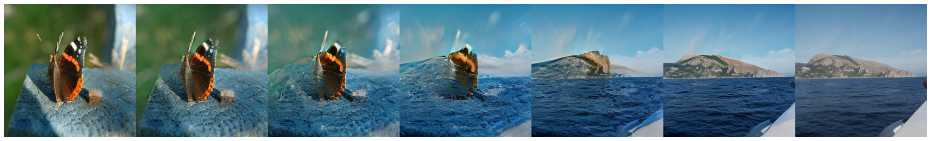


Fig. 5: Pairwise interpolation using GLEM on ImageNet dataset.

Table 8: Comparison of GLEM against closely related works. Blue rows show all the NF methods that classify using a GMM in the latent space (bold scores show the best results among them). NF + MLP are normalizing flows that classify with an MLP using a classification loss in the last layer. Red scores show the overall best results in the table. The lower part of the table shows the accuracy of CNNs trained on artificial images, for different generative models including IB-INN* (retrained following the setting in [3]).GLEM100× corresponds to the result from table 4, trained with 100 times more data. NA stands for not available.

Method	Accuracy		Model
	CIFAR10	CIFAR100	
GLEM (ours)	**92.63**	**70.11**	NF + GMM
IB-INN [3]	91.28	66.22	NF + GMM
IB-INN + KLJ [48]	88.6	NA	NF + GMM
FLOWGMM [26]	88.44		NF + GMM
ULCGM [18]	84.0		NF + GMM
Monotone Flow [1]	93.4	NA	NF + MLP
i-DenseNets [42]	92.40		NF + MLP
Residual NF [9]	91.78		NF + MLP
Invertible ResNets [6]	93.22	**75.42**	NF + MLP
Implicit NF [39]	92.71	70.94	NF + MLP
JEM++ [61]	**94.1**	74.5	EBM + GMM
SHOT-VAE [17]	93.89	74.70	VAE
GLEM (ours)+CNN	80.03	53.07	NF + CNN
GLEM100× (ours)+CNN	**87.14**	**63.52**	NF + CNN
IB-INN*+CNN	42.87	19.20	NF + CNN
SNGAN [46]	82.2	45.0	GAN + CNN

Table 9: Comparison of NF based augmentations with the best models from timm. It can be noted the GLEM based augmentation (blue) achieves the highest rank, in term of top1 accuracy, and the second highest in term of top5 accuracy. It also outperforms much larger transformers with 1B parameters. IS stands for image size in pixels. Besides timm, OmniVec [49] and ViT-G/14 soup [56] achieve higher results of 92.4% and 91.78%.

model name	top1-acc	top5-acc	params in M	IS
EVA02 base [15,51] GLEM augmentations (ours)	**91.229**	98.854	87.12	448^2
EVA02 LARGE [15,51]	91.129	98.713	305.08	448^2
EVA GIANT [16]	90.969	98.672	1,014.45	560^2
EVA02 base [15,51]	90.896	98.802	87.12	448^2
CAFormer [62]	90.781	**98.860**	98.75	384^2
Beit [14,5]	90.687	98.753	305.67	512^2
VOLO [63]	90.614	98.698	296.09	512^2

Fig. 6: 3 augmented images using GLEM (right), compared with the original image from ImageNet (left).

6 Conclusion

In this paper, we propose a new method that conditions NFs on multimodal gaussians with learned hyperparameters and controllable latent distributions. Our method achieves state of the art results, and outperforms closely related work both in classification accuracy as well as proxy metrics (including image diversity and label reliability). The strength of our method also resides in its ability to learn continuums of untangled gaussians using a loss that mixes a maximum likelihood criterion and a KLD term. As shown through the paper, the former enhances the generative properties of the resulting NFs while the latter improves their discrimination power. Extensive experiments — including model analysis, ablation study and comparisons — show the positive impact of our model both on classification accuracy and other proxy metrics.

Acknowledgments. This work was granted access to the HPC resources of IDRIS under the allocation 2023-AD011013954 made by GENCI.

The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ahn, B., Kim, C., Hong, Y., Kim, H.J.: Invertible monotone operators for normalizing flows. Advances in Neural Information Processing Systems **35**, 16836–16848 (2022)
2. Ardizzone, L., Kruse, J., Wirkert, S., Rahner, D., Pellegrini, E.W., Klessen, R.S., Maier-Hein, L., Rother, C., Köthe, U.: Analyzing inverse problems with invertible neural networks. arXiv preprint arXiv:1808.04730 (2018)
3. Ardizzone, L., Mackowiak, R., Rother, C., Köthe, U.: Training normalizing flows with the information bottleneck for competitive generative classification. Advances in Neural Information Processing Systems **33**, 7828–7840 (2020)
4. Atanov, A., Volokhova, A., Ashukha, A., Sosnovik, I., Vetrov, D.: Semi-conditional normalizing flows for semi-supervised learning. arXiv preprint arXiv:1905.00505 (2019)
5. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: Bert pre-training of image transformers. arXiv preprint arXiv:2106.08254 (2021)
6. Behrmann, J., Grathwohl, W., Chen, R.T., Duvenaud, D., Jacobsen, J.H.: Invertible residual networks. In: International conference on machine learning. pp. 573–582. PMLR (2019)
7. Bevins, H., Handley, W.: Piecewise normalizing flows. arXiv preprint arXiv:2305.02930 (2023)
8. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Advances in Neural Information Processing Systems. vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020), https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf
9. Chen, R.T., Behrmann, J., Duvenaud, D.K., Jacobsen, J.H.: Residual flows for invertible generative modeling. Advances in Neural Information Processing Systems **32** (2019)
10. Ciobanu, S.: Mixtures of normalizing flows. In: Proceedings of ISCA 34th International Conference on. vol. 79, pp. 82–90 (2021)
11. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
12. Dinh, L., Krueger, D., Bengio, Y.: Nice: Non-linear independent components estimation. arXiv preprint arXiv:1410.8516 (2014)
13. Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp. arXiv preprint arXiv:1605.08803 (2016)
14. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
15. Fang, Y., Sun, Q., Wang, X., Huang, T., Wang, X., Cao, Y.: Eva-02: A visual representation for neon genesis. arXiv preprint arXiv:2303.11331 (2023)
16. Fang, Y., Wang, W., Xie, B., Sun, Q., Wu, L., Wang, X., Huang, T., Wang, X., Cao, Y.: Eva: Exploring the limits of masked visual representation learning at scale.

In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19358–19369 (2023)
17. Feng, H.Z., Kong, K., Chen, M., Zhang, T., Zhu, M., Chen, W.: Shot-vae: semi-supervised deep generative models with label-aware elbo approximations. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 7413–7421 (2021)
18. Fetaya, E., Jacobsen, J.H., Grathwohl, W., Zemel, R.: Understanding the limitations of conditional generative models. arXiv preprint arXiv:1906.01171 (2019)
19. Gani, H., Naseer, M., Yaqub, M.: 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), https://bmvc2022.mpi-inf.mpg.de/0731.pdf
20. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
21. Hagemann, P., Neumayer, S.: Stabilizing invertible neural networks using mixture models. Inverse Problems **37**(8), 085002 (2021)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
23. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
24. Huang, C.W., Dinh, L., Courville, A.: Augmented normalizing flows: Bridging the gap between generative flows and latent variable models. arXiv preprint arXiv:2002.07101 (2020)
25. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Bach, F., Blei, D. (eds.) Proceedings of the 32nd International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 37, pp. 448–456. PMLR, Lille, France (07–09 Jul 2015), https://proceedings.mlr.press/v37/ioffe15.html
26. Izmailov, P., Kirichenko, P., Finzi, M., Wilson, A.G.: Semi-supervised learning with normalizing flows. In: International Conference on Machine Learning. pp. 4615–4630. PMLR (2020)
27. Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D., Wilson, A.G.: Averaging weights leads to wider optima and better generalization. arXiv preprint arXiv:1803.05407 (2018)
28. Jia, Y., Zhang, Y., Weiss, R., Wang, Q., Shen, J., Ren, F., Nguyen, P., Pang, R., Lopez Moreno, I., Wu, Y., et al.: Transfer learning from speaker verification to multispeaker text-to-speech synthesis. Advances in neural information processing systems **31** (2018)
29. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
30. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
31. Kingma, D.P., Dhariwal, P.: Glow: Generative flow with invertible 1x1 convolutions. Advances in neural information processing systems **31** (2018)
32. Kirichenko, P., Farajtabar, M., Rao, D., Lakshminarayanan, B., Levine, N., Li, A., Hu, H., Wilson, A.G., Pascanu, R.: Task-agnostic continual learning with hybrid probabilistic models. arXiv preprint arXiv:2106.12772 (2021)

33. Klein, S., Golling, T.: Decorrelation with conditional normalizing flows. arXiv preprint arXiv:2211.02486 (2022)
34. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images.(2009) (2009)
35. Kynkäänniemi, T., Karras, T., Laine, S., Lehtinen, J., Aila, T.: Improved precision and recall metric for assessing generative models. Advances in Neural Information Processing Systems **32** (2019)
36. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Computation **1**(4), 541–551 (1989). https://doi.org/10.1162/neco.1989.1.4.541
37. Lecun, Y., Chopra, S., Hadsell, R., Ranzato, M., Huang, F.: A tutorial on energy-based learning. MIT Press (2006)
38. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016)
39. Lu, C., Chen, J., Li, C., Wang, Q., Zhu, J.: Implicit normalizing flows. arXiv preprint arXiv:2103.09527 (2021)
40. Mackowiak, R., Ardizzone, L., Kothe, U., Rother, C.: Generative classifiers as a basis for trustworthy image classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2971–2981 (2021)
41. Naeem, M.F., Oh, S.J., Uh, Y., Choi, Y., Yoo, J.: Reliable fidelity and diversity metrics for generative models. In: International Conference on Machine Learning. pp. 7176–7185. PMLR (2020)
42. Perugachi-Diaz, Y., Tomczak, J., Bhulai, S.: Invertible densenets with concatenated lipswish. Advances in Neural Information Processing Systems **34**, 17246–17257 (2021)
43. Pomponi, J., Scardapane, S., Uncini, A.: Pseudo-rehearsal for continual learning with normalizing flows. arXiv preprint arXiv:2007.02443 (2020)
44. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
45. Sajjadi, M.S., Bachem, O., Lucic, M., Bousquet, O., Gelly, S.: Assessing generative models via precision and recall. Advances in neural information processing systems **31** (2018)
46. Shmelkov, K., Schmid, C., Alahari, K.: How good is my gan? In: Proceedings of the European conference on computer vision (ECCV). pp. 213–229 (2018)
47. Smith, L.N., Topin, N.: Super-convergence: Very fast training of neural networks using large learning rates. In: Artificial intelligence and machine learning for multi-domain operations applications. vol. 11006, pp. 369–386. SPIE (2019)
48. Song, K., Solozabal, R., Li, H., Takáč, M., Ren, L., Karray, F.: Robustly train normalizing flows via kl divergence regularization. Proceedings of the AAAI Conference on Artificial Intelligence **38**(13), 15047–15055 (Mar 2024). https://doi.org/10.1609/aaai.v38i13.29426, https://ojs.aaai.org/index.php/AAAI/article/view/29426
49. Srivastava, S., Sharma, G.: Omnivec: Learning robust representations with cross modal sharing. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision. pp. 1236–1248 (2024)
50. Stimper, V., Scholkopf, B., Hernandez-Lobato, J.M.: Resampling Base Distributions of Normalizing Flows
51. Sun, Q., Fang, Y., Wu, L., Wang, X., Cao, Y.: Eva-clip: Improved training techniques for clip at scale. arXiv preprint arXiv:2303.15389 (2023)

52. Tishby, N., Pereira, F.C., Bialek, W.: The information bottleneck method. arXiv preprint physics/0004057 (2000)
53. Villani, C., Society, A.M.: Topics in Optimal Transportation. Graduate studies in mathematics, American Mathematical Society (2003), https://books.google.fr/books?id=MyPjjgEACAAJ
54. Wang, T., Mirzazadeh, F., Zhang, X., Chen, J.: Gc-flow: A graph-based flow network for effective clustering. arXiv preprint arXiv:2305.17284 (2023)
55. Wightman, R.: Pytorch image models. https://github.com/rwightman/pytorch-image-models (2019). https://doi.org/10.5281/zenodo.4414861
56. Wortsman, M., Ilharco, G., Gadre, S.Y., Roelofs, R., Gontijo-Lopes, R., Morcos, A.S., Namkoong, H., Farhadi, A., Carmon, Y., Kornblith, S., et al.: Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In: International conference on machine learning. pp. 23965–23998. PMLR (2022)
57. Xiao, C., Liu, L.: Generative flows with matrix exponential. In: International Conference on Machine Learning. pp. 10452–10461. PMLR (2020)
58. Xiao, Z., Yan, Q., Amit, Y.: A method to model conditional distributions with normalizing flows. arXiv preprint arXiv:1911.02052 (2019)
59. Xu, C., Cheng, X., Xie, Y.: Invertible neural networks for graph prediction. IEEE Journal on Selected Areas in Information Theory **3**(3), 454–467 (2022)
60. Xu, C., Cheng, X., Xie, Y.: Normalizing flow neural networks by jko scheme. Advances in Neural Information Processing Systems **36** (2024)
61. Yang, X., Ji, S.: JEM++: Improved Techniques for Training JEM. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6474–6483. IEEE. https://doi.org/10.1109/ICCV48922.2021.00643, https://ieeexplore.ieee.org/document/9710663/
62. Yu, W., Si, C., Zhou, P., Luo, M., Zhou, Y., Feng, J., Yan, S., Wang, X.: Metaformer baselines for vision. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)
63. Yuan, L., Hou, Q., Jiang, Z., Feng, J., Yan, S.: Volo: Vision outlooker for visual recognition. IEEE transactions on pattern analysis and machine intelligence **45**(5), 6575–6586 (2022)