Normalizing flow sampling with Langevin dynamics in the latent space

Florentin Coeurdoux, Nicolas Dobigeon, Pierre Chainais

Machine Learning
https://doi.org/10.1007/s10994-024-06623-x

Normalizing flow sampling with Langevin dynamics in the latent space

Florentin Coeurdoux[1] · **Nicolas Dobigeon**[1] · **Pierre Chainais**[2]

Received: 20 May 2023 / Revised: 17 June 2024 / Accepted: 30 August 2024

Abstract
Normalizing flows (NF) use a continuous generator to map a simple latent (e.g. Gaussian) distribution, towards an empirical target distribution associated with a training data set. Once trained by minimizing a variational objective, the learnt map provides an approximate generative model of the target distribution. Since standard NF implement differentiable maps, they may suffer from pathological behaviors when targeting complex distributions. For instance, such problems may appear for distributions on multi-component topologies or characterized by multiple modes with high probability regions separated by very unlikely areas. A typical symptom is the explosion of the Jacobian norm of the transformation in very low probability areas. This paper proposes to overcome this issue thanks to a new Markov chain Monte Carlo algorithm to sample from the target distribution in the latent domain before transporting it back to the target domain. The approach relies on a Metropolis adjusted Langevin algorithm whose dynamics explicitly exploits the Jacobian of the transformation. Contrary to alternative approaches, the proposed strategy preserves the tractability of the likelihood and it does not require a specific training. Notably, it can be straightforwardly used with any pre-trained NF network, regardless of the architecture. Experiments conducted on synthetic and high-dimensional real data sets illustrate the efficiency of the method.

Keywords Normalizing flows · Generative models · Monte Carlo sampling · Metropolis adjusted Langevin algorithm

Editor: Herna L. Viktor.

✉ Florentin Coeurdoux
florentin.coeurdoux@irit.fr

Nicolas Dobigeon
nicolas.dobigeon@irit.fr

Pierre Chainais
pierre.chainais@centralelille.fr

[1] IRIT, CNRS, Toulouse INP, University of Toulouse, Toulouse, France

[2] CRIStAL, CNRS, Centrale Lille, University of Lille, Lille, France

Published online: 25 September 2024

1 Introduction

Normalizing flows (NF) are known to be very efficient generative models to approximate probability distributions in an unsupervised setting. For example, *Glow* (Kingma & Dhariwal, 2018) is able to generate very realistic human faces, competing with state-of-the-art algorithms of variational inference (Papamakarios et al., 2021). Despite some early theoretical results about their stability (Nalisnick et al., 2018) or their approximation and asymptotic properties (Behrmann et al., 2019), their training remains challenging in the most general cases. Their capacity is limited by intrinsic architectural constraints, resulting in a variational mismatch between the target distribution and the actually learnt distribution. In particular Cornish et al. (2020) pointed out the capital issue of target distributions with disconnected support featuring several components. Since NF provide a continuous differentiable change of variable, they are not able to deal with such distributions when using a monomodal (e.g., Gaussian) latent distribution. Even targeting multimodal distributions featuring high probability regions separated by very unlikely areas remains problematic. The trained NF is a continuous differentiable transformation so that the transport of latent samples to the target space may overcharge low probability areas with (undesired) samples. These out-of-distribution samples will correspond to smooth transitions between different modes, which leads to out-of-distribution samples, as discussed by Cornish et al. (2020).

Figure 1 illustrates this behavior on a archetypal example considering a bimodal two-dimensional two-moon target measure and a latent Gaussian measure. It is worth noting that, for this toy example, the target measure is not only empirically described by data points but also admits an explicitly known distribution. The NF is first trained on a large set of data points drawn from the true target distribution. Figure 1a shows the latent distribution p_Z actually learnt by the NF and computed after applying the (inverse) pushforward operator to the explicitly known target distribution p_X, i.e., $p_Z = f_\sharp^{-1} p_X$. It appears that the NF splits the expected Gaussian latent space into two sub-regions separated by an area of minimal likelihood. This area corresponds, in the target domain, to the low probability area between the two modes of the target distribution, which is somewhat expected.

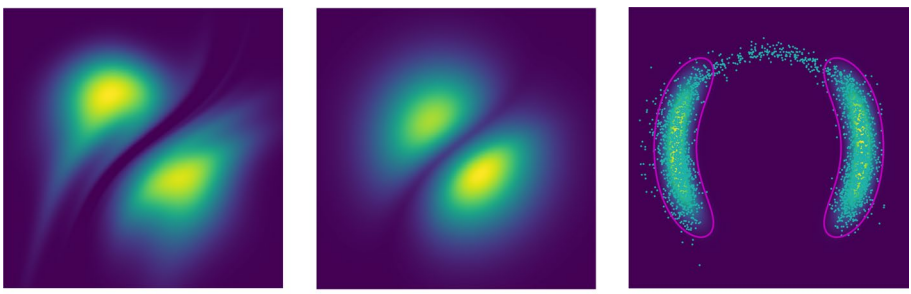


Fig. 1 **a** True latent measure p_Z given the explicit target measure p_X (chosen as a double-moon in this toy example); **b** Tempered distribution $\tilde{q}_Z = q_Z |J_f|^{-1}$ learnt by the NF when the instrumental latent measure q_Z is Gaussian; **c** Outputs $\left\{x^{(n)}\right\}_{n=1}^N$ drawn from the learnt target measure q_X with a naive sampling procedure, i.e., $x^{(n)} = f(z^{(n)})$ and $z \sim q_Z$. The reader is also invited to refer to Fig. 2 for an explicit description of the relationships between these measures

Figure 1c shows the result of a naive sampling from the Gaussian model latent distribution when the generated Gaussian samples are translated into the target domain thanks to the mapping learnt by the NF. The purple line represents the 97.5% level set. It appears that many samples generated by this naive sampling procedure are out-of-distribution in the low probability area between the two moons, see the top of the plot. They correspond to samples drawn from the latent Gaussian distribution in the low-likelihood area (depicted in dark blue in Fig. 1a) located between the two modes (represented in yellow and light green). Note that this illustrative example and in particular Fig. 1b will be more deeply discussed in the contributive Sects. 3 to 5 in the light of the findings reported along the paper.

The observations made above illustrate a behavior that is structural. NF are diffeomorphisms that preserve the topological structure of the support of the latent distribution. If the information about the structure of the target distribution is ignored, many out-of-distribution samples will be generated. This effect is reinforced by the fact that the NF is trained on a finite data set so that in practice there exist close to empty areas in low probability regions. In other words, since the latent distribution is usually a simple Gaussian unimodal distribution, there is a topological mismatch with the often much more complex target distribution (Cornish et al., 2020), in particular when it is multimodal.

A first contribution of this paper is a theoretical study of the impact of a topological mismatch between the latent distribution on the Jacobian of the NF transformation. We prove that the norm of the Jacobian of a sequence of differentiable mappings between a unimodal distribution and a distribution with disconnected support diverges to infinity (see Proposition 1). This observation suggests that one should consider the information brought by the Jacobian when sampling from the target distribution with a NF.

Capitalizing on this theoretical study, the second contribution of this paper is a new dedicated Markov chain Monte Carlo algorithm to sample efficiently according to the distribution targeted by a NF. The proposed sampling method builds on a Langevin dynamics formulated in the target domain and translated into the latent space, which is made possible thanks to the invertibility of the NF. Interestingly the resulting Langevin diffusion is defined on the Riemann manifold whose geometry is driven by the Jacobian of the NF. As a result, the proposed Markov chain Monte Carlo method is shown to avoid low probability regions and to produce significantly less out-of-distribution samples, even when the target distribution is multimodal. It is worth noting that the proposed method does not require a specific training procedure but can be implemented to sample from any pre-trained NF with any architecture.

The paper is organized as follows. Section 2 reports on related works. Section 3 recalls the main useful notions about normalizing flows. Section 4 studies the theoretical implications of a topological mismatch between the latent distribution and the target distribution. Section 5 introduces the proposed sampling method based on a Langevin dynamics in the latent space. In Sect. 6, numerical experiments illustrate the advantages of the proposed approach by reporting performance results both for 2D toy distributions and in high dimensions on the usual Cifar-10, CelebA and LSUN data sets.

2 Related works

Geometry in neural networks Geometry in neural networks as a tool to understand local generalization was first discussed by Bengio et al. (2013). As a key feature, the Jacobian matrix controls how smoothly a function interpolates a surface from some input data. As an extension, Rifai et al. (2011) showed that the norm of the Jacobian acts as a regularizer of the

deterministic autoencoder. Later Arvanitidis et al. (2018) were the first to establish the link between push forward generative models and surface modeling. In particular, they showed that the latent space could reveal a distorted view of the input space that can be characterized by a stochastic Riemannian metric governed by the local Jacobian.

Distribution with disconnected support As highlighted by Cornish et al. (2020), when using ancestral sampling, the structure of the latent distribution should fit the unknown structure of the target distribution. To tackle this issue, several solutions have been proposed. These strategies include augmenting the space on which the model operates (Huang et al., 2020), continuously indexing the flow layers (Cornish et al., 2020), and including stochastic (Wu et al., 2020) or surjective layers (Nielsen et al., 2020). However, these approaches sacrifice the bijectivity of the flow transformation. In most cases, this sacrifice has dramatic impacts: the model is no longer tractable, memory savings during training are no longer possible (Gomez et al., 2017), and the model is no longer a perfect encoder-decoder pair. Other works have promoted the use of multimodal latent distributions (Izmailov et al., 2020; Ardizzone et al., 2020; Hagemann & Neumayer, 2021). Nevertheless, rather than capturing the inherent multimodal nature of the target distribution, their primary motivation is to perform a classification task or to solve inverse problems with flow-based models. Papamakarios et al. (2017) has shown that choosing a mixture of Gaussians as a latent distribution could lead to an improvement of the fidelity to multimodal distributions. Alternatively, Pires and Figueiredo (2020) have studied the learning of a mixture of generators. Using a mutual information term, they encourage each generator to focus on a different submanifold so that the mixture covers the whole support. More recently, Stimper et al. (2022) predicted latent importance weights and proposed a sub-sampling method to avoid the generation of the most irrelevant samples. However, all these methods require to implement elaborated learning strategies which handle several sensitive hyperparameters or impose specific neural architectures. On the contrary, as emphasized earlier, the proposed approach does not require a specific training strategy, is computationally efficient, and can be implemented to any pre-trained NF.

Sampling with normalizing flows Recently NF have been used to facilitate the sampling from explicitly known distributions with non-trivial geometries. To solve the problem, samplers that combine Monte Carlo methods with NF have been proposed. On the one hand, flows have been used as reparametrization maps that improve the geometry of the target distribution before running local conventional samplers such as Hamiltonian Monte Carlo (HMC) (Hoffman et al., 2019; Noé et al., 2019). On the other hand, the push-forward of the NF base distribution through the map has also been used as an independent proposal in importance sampling (Müller et al., 2019) and Metropolis-Hastings steps (Gabrié et al., 2022; Samsonov et al., 2022). In this context, NF are trained using the reverse Kullback-Leiber divergence so that the push-forward distribution approximates the target distribution. These approaches are particularly appealing when a closed-form expression of the target distribution is available explicitly. In contrast, this paper does not assume an explicit knowledge of the target distribution. The proposed approach aims at improving the sampling from a distribution learnt by a given NF trained beforehand.

3 Normalizing flows: preliminaries and problem statement

3.1 Learning a change of variables

NF define a flexible class of deep generative models that seeks to learn a change of variable between a reference Gaussian measure q_Z and a target measure p_X through an invertible

transformation $f : \mathcal{Z} \to \mathcal{X}$ with $f \in \mathcal{F}$ where $\mathcal{F}$ defines the class of NF. Figure 2 summarizes the usual training of NF that minimizes a discrepancy measure between the target measure p_X and the push-forwarded measure q_X defined as

$$q_X = f_\sharp q_Z \tag{1}$$

where $f_\sharp$ stands for the associated push-forward operator. This discrepancy measure is generally chosen as the Kullback-Leibler (KL) divergence $D_{\mathrm{KL}}(p_X \| q_X)$. Explicitly writing the change of variables

$$q_X(x) = q_Z(f^{-1}(x))\left|J_{f^{-1}}(x)\right| \tag{2}$$

where $J_{f^{-1}}$ is the Jacobian matrix of f^{-1}, the training is thus formulated as the minimization problem

$$\min_{f \in \mathcal{F}} \quad \mathbb{E}_{p_X}[-\log q_Z(f^{-1}(x)) + \log |J_{f^{-1}}(x)|] \tag{3}$$

Note that the term $\log p_X(x)$ does not appear in the objective function since this term does not depend on f. In this work, the class $\mathcal{F}$ of admissible transformations is chosen as the structures composed of coupling layers ((Papamakarios et al., 2021; Dinh et al., 2016; Kingma & Dhariwal, 2018)) ensuring the Jacobian matrix of f to be lower triangular with positive diagonal entries. Because of this triangular structure, the Jacobian J_f and the inverse of the map f^{-1} are available explicitly. In particular the Jacobian determinant $\left|J_f(z)\right|$ evaluated at $z \in \mathcal{Z}$ measures the dilation, the change of volume of a small neighborhood around z induced by f, i.e., the ratio between the volumes of the corresponding neighborhoods of x and z.

In practice, the target measure p_X is available only though observed samples $\{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$. Adopting a sample-average approximation, the objective function in (3) is replaced by its Monte Carlo estimate. For this fixed set of samples per data batch, the NF training is formulated as

$$\hat{f} \in \min_{f \in \mathcal{F}} \frac{1}{N} \sum_{n=1}^{N} \left[-\log q_Z(f^{-1}(x^{(n)})) + \log |J_{f^{-1}}(x^{(n)})|\right]. \tag{4}$$

It is important to note that the obtained solution $\hat{f}$ is only an approximation of the exact transport map for two main reasons. First, the feasible set $\mathcal{F}$ (the class of admissible NF)

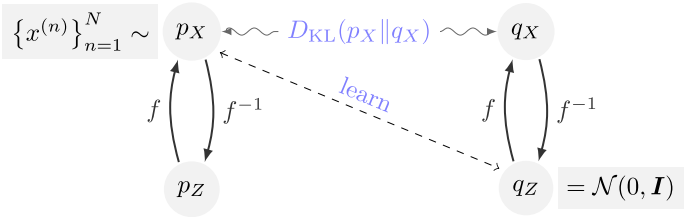


Fig. 2 NF learns a mapping f from data points $\{x^{(n)}\}_{n=1}^N$ assumed to be drawn from p_X towards the latent Gaussian measure q_Z. The training consists in minimizing the KL divergence between p_X and $q_X = f_\sharp q_Z$. Once trained, the learnt map permits to go from q_Z to q_X, which is an approximation of the true target distribution p_X (Color figure online)

is restricted to continuous, differentiable, and bijective functions. There is no guarantee that at least one transformation from this set will achieve $D_{\mathrm{KL}}(p_X \| q_X) = 0$. Second, even if such a transformation exists in $\mathcal{F}$, the solution $\hat{f}$ obtained by (4) only asymptotically matches the minimizer of (3) as $N \to \infty$.

The main issues inherent to the NF training and identified above would still hold for more refined training procedures (Coeurdoux et al., 2022), i.e., that would go beyond to the crude minimization problem (3). However, the work reported in this paper does not address the training of the NF. Instead, one will focus on the task which consists in generating samples from the learnt target measure. Thus one will assume that a NF has been already trained to learn a given change of variable. To make the sequel of this paper smoother to read, no distinction will be made between the sought transformation and its estimate, that will be denoted f in what follows.

3.2 A Gaussian latent space?

As noticed by Marzouk et al. (2016), learning the transformation f by variational inference can be reformulated with respect to (w.r.t.) the corresponding inverse map f^{-1}. Since the KL divergence is invariant to changes of variables, minimizing $D_{\mathrm{KL}}(p_X \| q_X)$ is equivalent to minimizing $D_{\mathrm{KL}}(p_Z \| q_Z)$ with $p_Z = f^{-1}_{\sharp} p_X$. The training procedure is thus formulated in the latent space instead of the target space. In other words, the NF aims at fitting the target measure p_Z expressed in the latent space to the latent Gaussian measure q_Z. However, due to inescapable shortcomings similar to those highlighted above, the target measure p_Z in the latent space is only an approximation of the latent Gaussian measure q_Z. This mismatch can be easily observed in Fig. 1a where the depicted actual measure p_Z is clearly not Gaussian. This issue may be particularly critical when there is a topological mismatch between the respective supports of the target and latent distributions. This will be discussed in more details in Sect. 4.

3.3 Beyond conventional NF sampling

Once the NF has been trained, the standard method to sample from the learnt target distribution is straightforward. It consists in drawing a sample z_k from the latent Gaussian distribution q_Z and then applying the learnt transformation f to obtain a sample $x^{(n)} = f\left(z^{(n)}\right)$. This method will be referred to as "naive sampling" in the sequel of this paper.

Unfortunately, as discussed in Sect. 3.2 (see also Fig. 1), the latent distribution q_Z is expected to be different from the actual target distribution p_Z expressed in the latent space. As suggested in the next section, this mismatch will be even more critical when it results from topological differences between the latent and target spaces. As a consequence the naive NF sampling is doomed to be suboptimal and to produce out-of-distribution samples, as illustrated in Fig. 1c. In contrast, the approach proposed in Sect. 5 aims at devising an alternative sampling strategy that explicitly overcomes these shortcomings.

4 Implications of a topological mismatch

The push-forward operator $f_\sharp$ learnt by an NF transports the mass allocated by q_Z in $\mathcal{Z}$ to $\mathcal{X}$, thereby defining q_X by specifying where each elementary mass is transported. This imposes a global constraint on the operator f if the model distribution q_X is expected to match a given target measure p_X perfectly. Let $\text{supp}(q_Z) = \{z \in \mathcal{Z} : q_Z(z) > 0\}$ denote the support of q_Z. Then the push-forward operator $f_\sharp$ can yield $q_X = p_X$ only if

$$\text{supp}(p_X) = \overline{f\left(\text{supp}(q_Z)\right)} \tag{5}$$

where $\overline{B}$ is the closure of set B. The constraint (5) is especially onerous for NF because of their bijectivity. The operators f and f^{-1} are continuous, and f is a homeomorphism. Consequently, for these models, q_Z and p_X are isomorphic, i.e., homeomorphic as topological spaces (Runde et al. 2005, Def. 3.3.10). This means that $\text{supp}(q_Z)$ and $\text{supp}(p_X)$ must share exactly the same topological properties, in particular the number of connected components. This constraint may be unlikely satisfied when learning complex real-world distributions, leading to an insurmountable topological mismatch. In such cases, this finding has serious consequences on the operator f learnt and implemented by a NF. Indeed, the following proposition states that if the respective supports of the latent distribution q_Z and the target distribution p_X are not homeomorphic, then the norm of the Jacobian $|J_f|$ of f may become arbitrary large. Here $\xrightarrow{\mathcal{D}}$ denotes weak convergence.

Proposition 1 *Let q_Z and p_X denote distributions defined on $\mathbb{R}^d$. Assume that $\text{supp}(q_Z) \neq \text{supp}(p_X)$. For any sequence of measurable, differentiable Lipschitz functions $f_t : \mathbb{R}^d \to \mathbb{R}^d$, if $f_{t\sharp}q_Z \xrightarrow{\mathcal{D}} p_X$ when $t \to +\infty$, then*

$$\lim_{t\to\infty} \sup_{z\in\mathcal{Z}} \left(\left\|J_{f_t}(z)\right\|\right) = +\infty.$$

The proof is reported in Appendix A.

It is worth noting that training a generative model is generally conducted by minimizing a statistical divergence. For most used divergence measures, (e.g., KL and Jensen-Shannon divergences, Wasserstein distance), this minimization implies a weak convergence of the approximated distribution q_X towards the target distribution p_X (Arjovsky et al., 2017). As a consequence, Proposition 1 states that in practice, when training a NF to approximate p_X with an iterative (e.g., stochastic gradient descent) algorithm, the learnt mapping f_t along the iterations (denoted here by t) is characterized by a Jacobian supremum which tends to explode in some regions as the algorithm approaches convergence. This result is in line with the experimental findings early discussed and visually illustrated by Fig. 1. Indeed, Fig. 1b depicts the heatmap of the log-likelihood

$$\log q_X(f(z)) = \log q_Z(z) - \log \left|J_f(z)\right| \tag{6}$$

given by (2) after training an NF. The impact of the term governed by the determinant of the Jacobian is clear. It highlights a boundary separating two distinct areas, each associated with a mode in the target distribution p_X. This result still holds when q_Z and q_X are defined on $\mathbb{R}^{d_Z}$ and $\mathbb{R}^{d_X}$, respectively, with $d_Z \neq d_X$. This shortcoming is thus also unavoidable when learning injective flow models (Kumar et al., 2017) and other push-forward models such as GANs (Goodfellow et al., 2020).

In practice, models are trained on a data set of finite size. In other words, the underlying target measure p_X is available only through the empirical measure $\frac{1}{N}\sum_{n=1}^{N}\delta_{x^{(n)}}$. During the training defined by (4), areas of low probability possibly characterizing a multi-modal target measure are likely interpreted as areas of null probability observed in the empirical measure. This directly results in the topological mismatch discussed above. Thus, even when targeting a distribution p_X defined over a connected support with regions of infinitesimal support, the learnt mapping is expected to be characterized by a Jacobian with exploding norm in these regions, see Fig 1.

This suggests that these regions correspond to the frontiers between cells defining a partition of the latent space. Specifically, when targeting a multi-modal distribution, the learned model implicitly partitions the Gaussian latent space into disjoint subsets associated with different modes. The boundaries of these subsets correspond to regions with a high Jacobian norm, which must be avoided during sampling to prevent out-of-distribution samples. The Gaussian multi-bubble conjecture was formulated when looking for a way to partition the Gaussian space with the least-weighted perimeter. This conjecture was proven recently by Milman and Neeman (2022). Recently, Issenhuth et al. (2022) leveraged on this finding to assess the optimality of the precision of GANs. They show that the precision of the generator vanishes when the number of components of the target distribution tends towards infinity.

5 NF sampling in the latent space

5.1 Local exploration of the latent space

As explained in Sect. 3.3, naive NF sampling boils down to drawing a Gaussian variable before transformation by the learnt mapping f. This strategy is expected to produce out-of-distribution samples, due to the topological mismatch between q_X and p_X discussed in Sect. 4. The proposed alternative elaborates directly on the learnt target distribution q_X.

The starting point of our rational consists in expressing a Langevin diffusion in the target space. This Markov chain Monte Carlo (MCMC) algorithm would target the distribution q_X using only the derivative of its likelihood $\nabla_x \log q_X(x)$. After initializing the chain by drawing from an arbitrary distribution $x_0 \sim \pi_0(x)$, the updating rule writes

$$x_{k+1} \leftarrow x_k + \frac{\epsilon^2}{2}\nabla_x \log q_X(x_k) + \epsilon\xi \tag{7}$$

where $\xi \sim \mathcal{N}(0, I)$ and $\epsilon > 0$ is a stepsize. When $\epsilon \to 0$ and the number of samples $K \to \infty$, the distribution of the samples generated by the iterative procedure (7) converges to q_X under some regularity conditions. In practice, the error is negligible when ϵ is sufficiently small and K is sufficiently large. This algorithm referred to as the unadjusted Langevin Algorithm (ULA) always accepts the generated sample proposed by (7), neglecting the errors induced by the discretization scheme of the continuous diffusion. To correct this bias, Metropolis-adjusted Langevin Algorithm (MALA) applies a Metropolis-Hastings step to accept or reject a sample proposed by ULA (Grenander & Miller, 1994).

Again, sampling according to q_X thanks to the diffusion (7) is likely to be inefficient due to the expected complexity of the target distribution possibly defined over a subspace of $\mathbb{R}^d$. In particular, this strategy suffers from the lack of prior knowledge about the location of the mass. Conversely, the proposed approach explores the latent space by leveraging on

the closed-form change of variable (2) operated by the trained NF. After technical derivations reported in Appendix C.2, the counterpart of the diffusion (7) expressed in the latent space writes

$$z' = z_k + \frac{\epsilon^2}{2} G^{-1}(z_k) \nabla_z \log \tilde{q}_Z(z_k) + \epsilon \sqrt{G^{-1}(z_k)} \xi \tag{8}$$

where

$$\tilde{q}_Z(z) = q_Z(z) \left| J_f(z) \right|^{-1} \tag{9}$$

and

$$G^{-1}(z) = \left[J_f^{-1}(z) \right]^2. \tag{10}$$

Note that the distribution $\tilde{q}_Z$ in (9) originates from the change of variable that defines q_X in (2) and has been already implicitly introduced by (6) in Sect. 4. Interestingly, the matrix $G(\cdot)$ is a positive definite matrix (see Appendix B). Thus the diffusion (8) characterizes a Riemannian manifold Langevin dynamics where $G(\cdot)$ is the Riemannian metric associated with the latent space (Girolami & Calderhead, 2011; Xifara et al., 2014). More precisely, it defines the conventional proposal move of the Riemannian manifold adjusted Langevin algorithm (RMMALA) which targets the distribution $\tilde{q}_Z$ defined by (9). This distribution is explicitly defined through the Jacobian $J_f(\cdot)$ of the transformation whose behavior has been discussed in depth in Sect. 4. It can be interpreted as the Gaussian latent distribution q_Z tempered by the (determinant of the) Jacobian of the transformation. It has also been evidenced by depicting the heatmap of (6) in Fig. 1b, which shows that it appears as a better approximation of p_Z than q_Z. Since it governs the drift of the diffusion through the gradient of its logarithm, the diffusion is expected to escape from the areas where the determinant of the Jacobian explodes, see Sect. 4.

The proposal kernel $g(z'|z)$ associated with the diffusion (8) is a Gaussian distribution whose probability density function (pdf) can be conveniently rewritten as (see Property 5 in Appendix C.2)

$$g\left(z' \mid z_k\right) \propto |J_{f(z_k)}| \exp\left[-\frac{1}{2\epsilon^2} \left\| J_f(z_k)(z' - z_k) + \frac{\epsilon^2}{2} \tilde{s}_Z(z_k) \right\|^2 \right]. \tag{11}$$

where $\tilde{s}_Z(\cdot)$ denotes the so-called latent score

$$\tilde{s}_Z(z) = J^{-1}(z) \nabla_z \log \tilde{q}_Z(z). \tag{12}$$

The sample proposed according to (8) is then accepted with probability

$$\alpha_{\text{RMMALA}}(z_k, z') = \min\left(1, \frac{\tilde{q}_Z(z') g\left(z_k \mid z'\right)}{\tilde{q}_Z(z_k) g\left(z' \mid z_k\right)} \right). \tag{13}$$

It is worth noting that the formulation (11) of the proposal kernel leads to a significantly faster implementation than its canonical formulation. Indeed, it does not require to compute the metric $G^{-1}(\cdot)$ defined by (10), which depends on the inverse of the Jacobian matrix twice. Moreover, the evaluation of the latent score (12) can be achieved in an efficient manner, bypassing the need for evaluating the inverse of the Jacobian matrix, as elaborated in

Appendix C.3.2. Finally, only the Jacobian associated with the forward transformation is required to compute (11). This approach enables a streamlined calculation of the acceptance ratio (13), ensuring an overall computational efficiency.

Besides, the proposal scheme (8) requires to generate high dimensional Gaussian variables with covariance matrix $\epsilon^2 G^{-1}(\cdot)$ (Vono et al., 2022). To lighten the corresponding computational burden, we take advantage of a 1st order expansion of f^{-1} to approximate (8) by the diffusion (see Appendix C.3.1)

$$z' = f^{-1}\left(f(z_k) + \epsilon\xi\right) + \frac{\epsilon^2}{2} J_f^{-1}(z_k)\tilde{s}_Z(z). \tag{14}$$

According to (14), this alternative proposal scheme requires to generate high dimensional Gaussian variables with a covariance matrix which is now identity, i.e., most cheaper. Moreover, it is worth noting that *i)* the latent score $\tilde{s}_Z(\cdot)$ can be evaluated efficiently (see above) and *ii)* using $J_f^{-1}(z) = J_{f^{-1}}(f(z))$ (see Property 2 in Appendix C.1), sampling z' according to (14) only requires to evaluate the Jacobian associated with the backward transformation. Proofs and implementation details are reported in Appendix C. The algorithmic procedure to sample according to this kernel denoted $\mathcal{K}_{\text{RMMALA}}(\cdot)$ is summarized in Algorithm 1.

Algorithm 1 Sampling kernel $\mathcal{K}_{\text{RMMALA}}(\cdot)$.

Input: trained NF $f(\cdot)$, time step ϵ, current state z_k of the sampler.
`/* Draw the candidate */`
Draw $\xi \sim \mathcal{N}(0, 1)$
Set $z' = f^{-1}\left(f(z_k) + \epsilon \cdot \xi\right) + \frac{\epsilon^2}{2} J_f^{-1}(z_k)\tilde{s}_Z(z_k)$ (see Eq. (14))
`/* Accept/reject procedure */`
Draw $u \sim \mathcal{U}(0, 1)$
if $u < \alpha_{\text{RMMALA}}(z_k, z')$ (see Eq. (13)) **then**
  Set $z_{k+1} = z'$
else
  Set $z_{k+1} = z_k$
Output: New state $z_{k+1} = \mathcal{K}_{\text{RMMALA}}(z_k)$ of the sampler.

5.2 Independent Metropolis-Hastings sampling

Handling distributions that exhibit several modes or defined on a complex multi-component topology is another major issue raised by the problem addressed here. In practice, conventional sampling schemes such as those based on Langevin dynamics fail to explore the full distribution when modes are isolated since they may get stuck around one of these modes. Thus, the samples proposed according to (8) in areas with high values of $\|J_f(\cdot)\|$

are expected to be rejected. These areas have been identified in Sect. 4 as the low probability regions between modes when targeting a multimodal distribution. To alleviate this problem, one strategy consists in resorting to another kernel to propose moves from one high probability region to another, without requiring to cross the low probability regions. Following this strategy, this paper proposes to combine the diffusion (8) with an independent Metropolis-Hastings (I-MH) with the distribution q_Z as a proposal. The corresponding acceptance ratio writes

$$\begin{aligned} \alpha_{\text{I-MH}}(z_k, z') &= \min\left(1, \frac{\tilde{q}_Z(z')q_Z(z_k)}{\tilde{q}_Z(z_k)q_Z(z')}\right) \\ &= \min\left(1, \frac{|J_f(z_k)|}{|J_f(z')|}\right). \end{aligned} \tag{15}$$

It is worth noting that this probability of accepting the proposed move only depends on the ratio between the Jacobians evaluated at the current and the candidate states. In particular, candidates located in regions of the latent space characterized by exploding Jacobians in case of a topological mismatch (see Sect. 4) are expected to be rejected with high probability. Conversely, this kernel will favor moves towards other high probability regions not necessarily connected to the regions of the current state. The algorithmic procedure is sketched in Algorithm 2.

Algorithm 2 Sampling kernel $\mathcal{K}_{\text{I-MH}}(\cdot)$.

Input: trained NF $f(\cdot)$, current state z_k of the sampler.
`/* Draw candidate */`
Draw $z' \sim \mathcal{N}(0, 1)$
`/* Accept/reject procedure */`
Draw $u \sim \mathcal{U}(0, 1)$
if $u < \alpha_{\text{I-MH}}(z_k, z')$ (see Eq. (15)) **then**
  Set $z_{k+1} = z'$
else
  Set $z_{k+1} = z_k$
Output: New state $z_{k+1} = \mathcal{K}_{\text{I-MH}}(z_k)$ of the sampler.

Finally, the overall proposed sampler, referred to as NF-SAILS for NF SAmpling In the Latent Space and summarized in Algorithm 3, combines the transition kernels $\mathcal{K}_{\text{RMMALA}}$ and $\mathcal{K}_{\text{I-MH}}$, which permits to efficiently explore the latent space both locally and globally. At each iteration k of the sampler, the RMMALA kernel $\mathcal{K}_{\text{RMMALA}}$ associated with the acceptance ratio (13) is selected with probability p and the I-MH kernel $\mathcal{K}_{\text{I-MH}}$ associated with acceptance ratio (15) is selected with the probability $1 - p$. Again, one would like to emphasize that the proposed strategy does not depend on the NF architecture and can be adopted to sample from any pretrained NF model.

Algorithm 3 NF-SAILS: NF SAmpling In the Latent Space

Input: trained NF $f(\cdot)$, stepsize ϵ, probability p

/* Initialization */

Draw $z_0 \sim \pi_0(z)$

for $k = 0$ **to** K **do**

  /* Choose the kernel */

  Draw $u \sim \mathcal{U}(0, 1)$

  if $u < p$ **then**

    /* LOCAL EXPLORATION (see Algo. 1) */

    $z_{k+1} = \mathcal{K}_{\text{RMMALA}}(z_k)$

  else

    /* GLOBAL EXPLORATION (see Algo. 2) */

    $z_{k+1} = \mathcal{K}_{\text{I-MH}}(z_k)$

end

Output: Collection of samples $\{z_k\}_{k=1}^{K}$.

6 Experiments

This section reports performance results to illustrate the efficiency of NF-SAILS thanks to experiments based on several models and synthetic data sets. It is compared to state-of-the-art generative models known for their abilities to handle multimodal distributions. These results will show that the proposed sampling strategy achieves good performance, without requiring to adapt the NF training procedure or resorting to non-Gaussian latent distributions. We will also confirm the relevance of the method when working on popular image data sets, namely Cifar-10 (Krizhevsky et al., 2010), CelebA (Liu et al., 2015) and LSUN (Yu et al., 2015).

To illustrate the versatility of proposed approach w.r.t. the NF architecture, two types of coupling layers are used to build the trained NF. For the experiments conducted on the synthetic data sets, the NF architecture is RealNVP (Dinh et al., 2016). Conversely, a Glow model is used for experiments conducted on the image data sets (Kingma & Dhariwal, 2018). However, it is worth noting that the proposed method can apply on top of any generative model fitting multimodal distributions. Additional details regarding the training procedure are reported in Appendix D.1.

6.1 Figures-of-merit

To evaluate the performance of the NF, several figures-of-merit have been considered. When addressing bidimensional problems, we perform a Kolmogorov-Smirnov test to assess the quality of the generated samples w.r.t. the underlying true target distribution (Justel et al., 1997). The goodness-of-fit is also monitored by evaluating the mean log-likelihood of the generated samples and the entropy estimator between samples, which approximates the Kullback–Leibler divergence between empirical samples (Kraskov et al., 2004).

For applications to higher dimensional problems, such as image generation, the performances of the compared algorithms are evaluated using the Fréchet inception distance (FID) (Heusel et al., 2017) using a classifier pre-trained specifically on each data set. Besides, for completeness, we report the bits per dimension (bpd) (Papamakarios et al., 2017), i.e., the log-likelihoods in the logit space, since this is the objective optimized by the trained models.

6.2 Results obtained on synthetic data set

As a first illustration of the performance of NF-SAILS, we consider to learn a mixture of k bidimensional Gaussian distributions, with $k \in \{2, 3, 4, 6, 9\}$. The NF model $f(\cdot)$ is a RealNVP (Dinh et al., 2016) composed of $M = 4$ flows, each composed of two three-layer neural networks ($d \to 16 \to 16 \to d$) using hyperbolic tangent activation function. We use the Adam optimizer with learning rate 10^{-4} and a batch size of 500 samples.

Table 1 reports the considered metrics when comparing the proposed NF-SAILS sampling method to a naive sampling (see Sect. 3.3) or to state-of-the-art sampling techniques from the literature, namely Wasserstein GAN with gradient penalty (WGAN-GP) (Gulrajani et al., 2017) and denoising diffusion probabilistic models (DDPM) (Ho et al., 2020). These results show that NF-SAILS consistently competes favorably against the compared methods, in particular as the degree of multimodality of the distribution increases. Note that WGAN-GP exploits a GAN architecture. Thus, contrary to the proposed NF-based sampling method, it is unable to provide an explicit evaluation of the likelihood, which explains the N/A values in the table.

Figure 3 illustrates this result for $k = 6$ and shows that our method considerably reduces the number of out-of-distribution generated samples. Additional results are reported in Appendix D.2.

Figure 4 depicts the samples generated when using a single kernel of the proposed NF-SAILS algorithm independently, i.e., when a single I-MH kernel $\mathcal{K}_{\text{I-MH}}$ (left panel) or a single RMMALA $\mathcal{K}_{\text{RMMALA}}$ (middle and right panels) is used. It also shows the impact of the stepsize on the local exploration performed by the RMMALA kernel. For various tuning of the parameters, the effective sample size (ESS) and the rate of rejection (p_{reject}) are reported in the associated table. Using the single I-MH kernel ($p = 0$) leads to a good exploration and good effective sample size (ESS); however it is not very efficient due to high number of rejection ($p_{\text{reject}} = 0.5$). On the other hand, using only the RMMALA kernel ($p = 1$) leads to a higher efficiency ($p_{\text{reject}} = 0.1$) and lower ESS but fails to explore all the modes. Besides, regarding the stepsize ϵ, the smaller the less efficient the sampling, as shown in the middle and right panels. In the experiments described in this paper, this stepsize has been adjusted following the heuristic of the order of magnitude of $d^{1/3}$, where d is the dimension of the problem, as advocated in Pillai et al. (2012). Combining the two kernels in NF-SAILS (with $p = 0.7$ and $\epsilon = 0.2$, see last line of the table) seems to be the most efficient strategy to explore all the modes of the targeted distribution, with the best ESS-p_{reject} trade-off.

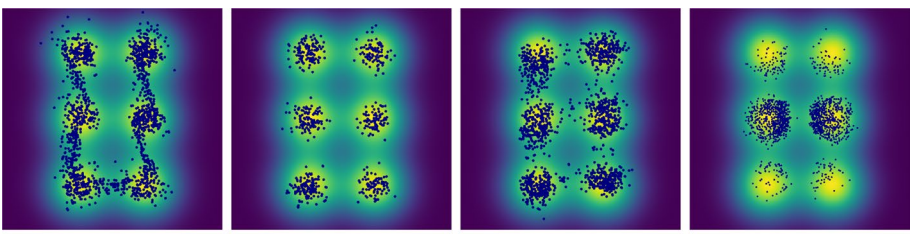
Table 1 Goodness-of-fit of the generated samples w.r.t. the number k of Gaussians

	$\uparrow \log p_X$	$\downarrow$ KL	$\downarrow$ KS
$k = 2$			
Naive sampling	−4.08 ±0.19	0.258 ±0.08	0.178 ±0.17
NF-SAILS	**−1.46** ±0.08	**0.053** ±0.02	**0.052** ±0.01
WGAN-GP	N/A	0.311 ±0.07	0.287 ±0.06
DDPM	−2.98 ±0.18	0.121 ±0.04	0.066 ±0.04
$k = 3$			
Naive sampling	−3.54 ±0.15	0.886 ±0.09	0.242 ±0.1
NF-SAILS	**−1.83** ±0.14	**0.056** ±0.03	**0.034** ±0.04
WGAN-GP	N/A	0.981 ±0.07	0.237 ±0.07
DDPM	−2.97 ±0.17	0.364 ±0.04	0.124 ±0.04
$k = 4$			
Naive sampling	−3.08 ±0.16	0.961 ±0.08	0.289 ±0.08
NF-SAILS	**−1.07** ±0.14	**0.044** ±0.01	**0.041** ±0.01
WGAN-GP	N/A	1.012 ±0.09	0.317 ±0.08
DDPM	−1.81 ±0.16	0.427 ±0.04	0.127 ±0.03
$k = 6$			
Naive sampling	−2.06 ±0.15	1.219 ±0.08	0.205 ±0.06
NF-SAILS	**−1.09** ±0.13	**0.039** ±0.01	0.309 ±0.01
WGAN-GP	N/A	1.392 ±0.09	0.212 ±0.04
DDPM	−1.99 ±0.14	1.004 ±0.06	**0.179** ±0.03
$k = 9$			
Naive sampling	−2.297 ±0.13	1.764 ±0.1	0.215 ±0.06
NF-SAILS	**−0.801** ±0.12	**0.151** ±0.01	**0.052** ±0.01
WGAN-GP	N/A	1.939 ±0.15	0.340 ±0.07
DDPM	−1.258 ±0.13	0.906 ±0.07	0.205 ±0.05

Reported scores (means and standard deviations) result from the average over 50 Monte Carlo runs

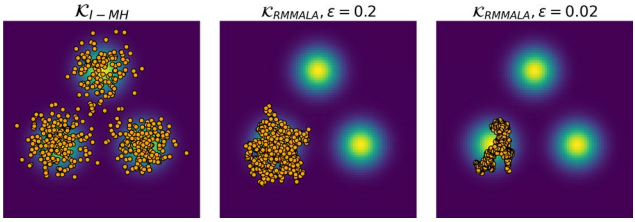
6.3 Results obtained on real image data sets

Moreover, we further study the performance of NF-SAILS on three different real image data sets, namely Cifar-10 (Krizhevsky et al., 2010), CelebA (Liu et al., 2015) and LSUN (Yu et al., 2015). Following the same protocol as implemented by Kingma and



(a) Naive sampling (b) NF-SAILS (c) WGAN-GP (d) DDPM

Fig. 3 Mixture of $k = 6$ Gaussian distributions (green), and 1000 generated samples (blue). The proposed NF-SAILS method in Fig. 3b does not generate samples in-between modes



Hyperparameters	ESS	p_{reject}
$p = 0.0$	3.1	50%
$p = 1.0,\ \epsilon = 0.2$	4.4	10%
$p = 1.0,\ \epsilon = 0.7$	3.8	30%
$p = 1.0,\ \epsilon = 0.02$	10.3	3%
$p = 0.7,\ \epsilon = 0.2$	3.5	20%

Fig. 4 Mixture of $k = 3$ Gaussian distributions (green): impact of the kernels and the hyperparameters p and ϵ. Left: $p = 0$, i.e., using the single $\mathcal{K}_{\text{I–MH}}$ kernel. Middle and right panels: $p = 1$, i.e., using the single $\mathcal{K}_{\text{RMMALA}}$ kernel for two values of the stepsize ϵ. The table reports the ESS and the rate of rejection for various combinations of the hyperparametervalues. The last line of the table corresponds the implementation of NF-SAILS adopted for this toy example (Color figure online)

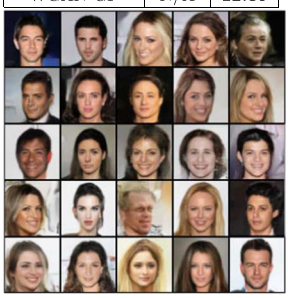
Dhariwal (2018), we use a Glow architecture where each neural network are composed of three convolutional layers. The two hidden layers have ReLU activation functions and 512 channels. The first and last convolutions are 3×3, while the center convolution is 1×1, since its input and output have a large number of channels, in contrast with the first and last convolutions. Details regarding the implementation are reported in Appendix D.3.

We compare the FID score as well as the average negative log-likelihood (bpd), keeping all training conditions constant and averaging the results over 10 Monte Carlo runs. The results are depicted in Fig. 5 reports the results when compared to those obtained by naive sampling or WGAN-GP (Gulrajani et al., 2017). As shown by the different panels of this figure, the proposed NF-SAILS method considerably improves the quality of the generated images, both quantitatively (in term of FID) and semantically. Our methodology compares favourably w.r.t. to WGAN-GP for the two data sets CelebA and LSUN.

	↓ bpd	↓ FID
Naive sampling	3.35	44.6
NF-SAILS	**3.08**	43.11
WGAN-GP	N/A	**18.8**

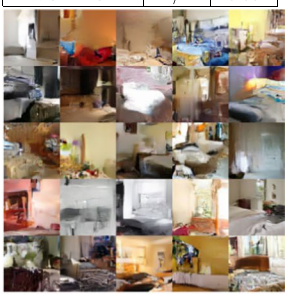
(a) Cifar10

	↓ bpd	↓ FID
Naive sampling	1.03	15.82
NF-SAILS	**0.96**	14.12
WGAN-GP	N/A	**12.89**



(b) CelebA

	↓ bpd	↓ FID
Naive sampling	2.38	8.91
NF-SAILS	**2.11**	**7.86**
WGAN-GP	N/A	9.56



(c) LSUN (bedroom)

Fig. 5 Tables report quantitative and perceptual metrics computed from the samples generated by the compared methods. The figures show some samples generated from Glow using the proposed NF-SAILS method

7 Conclusion

This paper discusses the sampling from the target distribution learnt by a normalizing flow. Architectural constraints prevent normalizing flows to properly learn disconnect support measures due to the topological mismatch between the latent and target spaces. Moreover, we theoretically prove that Jacobian norm of the transformation become arbitrarily large to closely represent such target measures. The conducted analysis exhibits the existence of pathological areas in the latent space corresponding to points with exploding Jacobian norms. Using a naive sampling strategy leads to out of distribution samples located in these areas. To overcome this issue, we propose a new sampling procedure based on a Langevin diffusion directly formulated in the latent space. This sampling is interpreted as a Riemanian manifold Metropolis adjusted Langevin algorithm, whose metrics is driven by the Jacobian of the learnt transformation. This local exploration of the latent space is complemented by an independent Metropolis-Hastings kernel which allows moves from one high probability region to another while avoiding crossing pathological areas. One particular advantage of the proposed is that it can be applied to any pre-trained NF model. Indeed it does not require a particular training strategy of the NF or to adapt the distribution assumed in the latent space. The performances of the proposed sampling strategy show to compare favorably to state-of-the art, with very few out-of-distribution samples.

Appendix A: Proof of Proposition 1

The proof of Proposition 1 in Sect. 4 combines existing results from topology and real analysis. The complete background can be found in (Dudley, 2002) and (Cornish et al., 2020). The proof is mainly based on the following results.

Theorem 1 (Cornish et al., 2020) *Let q_Z and q_X define probability measures on $\mathbb{R}^d$, with* $\text{supp}(q_Z) \neq \text{supp}(q_X)$. *For any sequence of measurable, differentiable Lipschitzian functions $f_n : \mathbb{R}^d \to \mathbb{R}^d$, if the sequence weakly converges as $f_{n\#}q_Z \xrightarrow{\mathcal{D}} q_X$, then*

$$\lim_{n\to\infty} \text{Lip} f_n = \infty. \tag{A1}$$

Moreover, Behrmann et al. (2019) showed the relation between the Lipschitz constant and the Jacobian of a transformation, as stated below.

Lemma 1 (Rademacher's theorem) *If $f : \mathbb{R}^m \to \mathbb{R}^n$ is Lipschitzian, then f is continuous and differentiable at almost all points of $\mathbb{R}^m$ and*

$$\text{Lip} f = \sup_{z\in\mathcal{Z}} \|J_f(z)\|_{\text{op}} \tag{A2}$$

Both Theorem 1 and Lemma 1 rely on the same starting hypothesis, i.e., f is required to be continuous, differentiable and Lipschitzian. Combining these two results yields Proposition 1 following a development of the proof of the results by Cornish et al. (2020) and Behrmann et al. (2019).

Appendix B: Properties of the Jacobian of coupling layer-based NF

B.1 Structure of the Jacobian matrix and computation of its determinant

RealNVP model defines a NF by implementing a sequence of M invertible bijective transformation functions, herein referred to as coupling layers (Dudley, 2002). In other words, the mapping f writes as $f = f^{(M)} \circ f^{(M-1)} \circ f^{(2)} \circ f^{(1)}$. Each bijection $f^{(m)} : u \mapsto v$ associated to the mth layer splits the input $u \in \mathbb{R}^D$ into two parts of sizes d and $d - D$ ($d \leq D$), respectively, such that the output $v \in \mathbb{R}^D$ writes

$$\begin{cases} v_{1:d} &= u_{1:d} \\ v_{d+1:D} &= u_{d+1:D} \odot \exp\left(h^{(m)}\left(u_{1:d}\right)\right) + t^{(m)}\left(u_{1:d}\right) \end{cases} \tag{B3}$$

where $h_m(\cdot) : \mathbb{R}^d \to \mathbb{R}^{D-d}$ and $t_m(\cdot) : \mathbb{R}^d \to \mathbb{R}^{D-d}$ are scale and translation functions implemented as deep networks and $\odot$ stands for the Hadamard product. The Jacobian of the above transformation is a lower triangular matrix

$$J^{(m)}(u) = \begin{bmatrix} I_d & \mathbf{0}_{d\times(D-d)} \\ A^{(m)}(u) & E^{(m)}(u) \end{bmatrix} \tag{B4}$$

where I_d and $\mathbf{0}_{d\times(D-d)}$ are the identity and zero matrices with indexed sizes, respectively, and

$$\begin{cases} A^{(m)}(u) &= u_{d+1:D} \odot \frac{\partial \exp h^{(m)}(u_{1:d})}{\partial u_{1:d}} + \frac{\partial t^{(m)}(u_{1:d})}{\partial u_{1:d}} \\ E^{(m)}(u) &= \operatorname{diag}\left(\exp\left(h^{(m)}\left(u_{1:d}\right)\right)\right). \end{cases} \tag{B5}$$

Thanks to the chain rule, it follows that the Jacobian of the overall NF is

$$J_f(z) = \prod_{j=1}^{J} J^{(m)}(u^{(m)}) \tag{B6}$$

with $u^{(m)} = f^{(m-1)}(u^{(m-1)})$ and $z = u^{(0)}$.

Moreover, because of the structure of each layer, the determinant of the Jacobian $J^{(m)}(u)$ associated with the mth layer is

$$|J^{(m)}(u)| = \prod_{k=1}^{d} \exp\left(h^{(m)}\left(u_k\right)\right). \tag{B7}$$

The determinant of the Jacobian $J_f(\cdot)$ characterizing the overall NF can be easily computed from (B6) and (B7).

B.2 Positive definiteness of the Jacobian

Property 1 *The product of two lower triangular matrices with strictly positive diagonal elements is a positive definite lower triangular matrix.*

Proof Let $A = \left[a_{ij}\right]$ and $B = \left[b_{ij}\right]$ be two $n \times n$ lower triangular matrices with positive diagonal entries, i.e.,

$$\forall i, j \text{ such that } i < j, \text{ then } a_{ij} = b_{ij} = 0. \tag{B8}$$

$$\forall i \; a_{ii} > 0 \text{ and } b_{ii} > 0 \tag{B9}$$

Let $C = \left[c_{ij}\right]$ denote the product matrix $C = AB$ with $c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$. The upper elements c_{ij} $(i < j)$ of C can be computed as

$$c_{ij} = \sum_{k=1}^{i} a_{ik} b_{kj} + \sum_{k=i+1}^{n} a_{ik} b_{kj}. \tag{B10}$$

In the right hand side of (B10), if $k \leq i$ then $b_{kj=0}$. Moreover if $k > i$ then $a_{ik} = 0$. As a consequence, $c_{ij} = 0$ and C is triangular.

Moreover, the eigenvalues of a triangular matrix is its diagonal elements. It follows that C is positive definite. □

Thanks to the structure of coupling layer-based NF discussed in Appendix B.1, we have the two following corollaries.

Corollary 1 *The Jacobian matrix $J_f(\cdot)$ and its inverse $J_f^{-1}(\cdot)$ of coupling layer-based NF are positive definite.*

Corollary 2 *The matrix $G(\cdot)$ and its inverse $G^{-1}(\cdot)$ are positive definite.*

Appendix C: Diffusion in the latent space

C.1 Preliminaries

The Langevin diffusion is a particular instance of the Itô process defined in the following Lemma of which a proof is given in (Øksendal & Øksendal, 2003).

Lemma 2 (Itô's lemma) *Let X_t denote an Iô drift-diffusion process defined by the stochastic differential equation*

$$dX_t = \mu_t dt + \sigma_t dB_t. \tag{C11}$$

If $f : \mathbb{R}^2 \to \mathbb{R}$ is a differentiable scalar function, then

$$df\left(t, X_t\right) = \left(\frac{\partial f}{\partial t} + \mu_t \frac{\partial f}{\partial x} + \frac{\sigma_t^2}{2} \frac{\partial^2 f}{\partial x^2}\right) dt + \sigma_t \frac{\partial f}{\partial x} dB_t. \tag{C12}$$

It yields that $f\left(t, X_t\right)$ is an It drift-diffusion process itself.

The following property shows that for any bijective transformation, the Jacobian of the inverse transformation is equal to the inverse of the Jacobian of the transformation. This result will be useful later.

Property 2 *Let $f : \mathcal{Z} \to \mathcal{X}$ denote a bijective transformation and $J_f(\cdot)$ its Jacobian, then*

$$J_{f^{-1}}(f(z)) = J_f^{-1}(z).$$

Proof Let h and g denote two multivariate functions. The chain rule writes

$$J_{h \circ g}(\cdot) = J_h(g(\cdot)) J_g(\cdot) \tag{C13}$$

thus

$$J_h(g(\cdot)) = J_{h \circ g}(\cdot) J_g^{-1}(\cdot). \tag{C14}$$

Moreover, for any multivariate bijective function f, we have

$$J_{f \circ f^{-1}}(\cdot) = J_{f^{-1} \circ f}(\cdot) = I_d. \tag{C15}$$

Combining (C14) and (C15) with $h = f^{-1}$ and $g = f$ yields

$$J_{f^{-1}}(f(z)) = J_{f^{-1} \circ f}(z) J_f^{-1}(z) = I_d J_f^{-1}(z) = J_f^{-1}(z). \tag{C16}$$

□

The following property demonstrates that the gradient of the score of q_X can be expressed over the latent space $\mathcal{Z}$ using $\tilde{q}_Z$ defined in (9).

Property 3 *Let $f : \mathcal{Z} \to \mathcal{X}$ be a bijective transformation which maps a latent measure q_Z towards a target measure q_X. Then the score of $q_X(x)$ is given by*

$$\nabla_x \log q_X(x) = J_f^{-1}(z) \cdot \nabla_z \log \tilde{q}_Z(z)$$

where $\tilde{q}_Z(z) = q_Z(z) \left| J_f(z) \right|^{-1}$.

Proof From the definition of $q_X(x)$ in equation (2), the score of $q_X(x)$ writes

$$\nabla_x \log q_X(x) = \nabla_x \left[\log q_Z(f^{-1}(x)) + \log |J_{f^{-1}}(x)| \right] \tag{C17}$$

and, from Property 2,

$$\nabla_x \log q_X(x) = \nabla_{f(z)} \left[\log q_Z(z) + \log \left| J_f(z) \right|^{-1} \right] \tag{C18}$$

$$= \nabla_{f(z)} \log \tilde{q}_Z(z) \tag{C19}$$

The chain rule now leads to

$$\nabla_x \log q_X(z) = \nabla_x f^{-1}(x) \cdot \nabla_z \log \tilde{q}_Z(z) \tag{C20}$$

$$= J_{f^{-1}}(f(z)) \cdot \nabla_z \log \tilde{q}_Z(z) \tag{C21}$$

which, using Property 2, can be finally rewritten as

$$\nabla_x \log q_X(x) = J_f^{-1}(z) \cdot \nabla_z \log \tilde{q}_Z(z). \tag{C22}$$

□

C.2 Derivation of the proposal distribution

The following property shows that the Lanvegin diffusion which targets the distribution q_X can be rewritten as a diffusion over the latent space $\mathcal{Z}$.

Property 4 *We consider the overdamped Langevin Itô diffusion*

$$dX_t = \nabla_x \log q_X(X_t)dt + \sigma_t dB_t \tag{C23}$$

driven by the time derivative of a standard Brownian motion B_t. In the limit $t \to \infty$, this probability distribution X_t approaches a stationary distribution q_X. Let $f : \mathcal{Z} \to \mathcal{X}$ be a bijective transformation which maps a latent measure q_Z towards the target measure q_X. A counterpart Langevin diffusion expressed over the latent space $\mathcal{Z}$ writes

$$dZ_t = G^{-1}(Z_t)\nabla_z \log \tilde{q}_Z(Z_t)dt + \sigma_t \sqrt{G^{-1}(Z_t)} dB_t \tag{C24}$$

Proof The Langevin diffusion is a particular instance of the Itô process where the drift μ_t in (C11) is given by the gradient of the log-density $\nabla_x \log q_X(X_t)$, i.e.,

$$dX_t = \nabla_x \log q_X(X_t)dt + \sigma_t dB_t \tag{C25}$$

We are interested in the diffusion process of $f^{-1}(X_t)$ when $f(\cdot)$ is a NF which is continuous, differentiable and bijective such that $f(Z_t) = X_t$ and $f^{-1}(X_t) = Z_t$. The Îto's Lemma 2 states

$$df^{-1}(X_t) = \left(J_{f^{-1}}(X_t)\nabla_x \log q_X(X_t) + \frac{\sigma_t^2}{2}\text{tr}(H_{f^{-1}}(X_t)) \right)dt + \sigma_t J_{f^{-1}}(X_t)dB_t. \tag{C26}$$

Neglecting the second-order terms yields

$$df^{-1}(X_t) = J_{f^{-1}}(X_t)\nabla_x \log q_X(X_t)dt + \sigma_t J_{f^{-1}}(X_t)dB_t. \tag{C27}$$

Using Property 2, Eq. (C27) can be rewritten as

$$dZ_t = J_f^{-1}(Z_t)\nabla_x \log q_X(X_t)dt + \sigma_t J_f^{-1}(Z_t)dB_t. \tag{C28}$$

Finally, by denoting $G^{-1}(z) = \left[J_f^{-1}(z)\right]^2$ and using Property 3, the diffusion in the latent space writes

$$dZ_t = G^{-1}(Z_t)\nabla_z \log \tilde{q}_Z(Z_t)dt + \sigma_t \sqrt{G^{-1}(Z_t)} dB_t \tag{C29}$$

□

The discretization of the stochastic differential equation (C29) using the Euler-Maruyama scheme can be written as in (8). This discretized counterpart of the diffusion corresponds to the proposal move of a Riemann manifold Metropolis-Adjusted Langevin

algorithm which targets $\tilde{q}_Z$. The following property shows that the associated proposal kernel can be rewritten as (11).

Property 5 *The discrete Langevin diffusion given by*

$$z' = z + \frac{\epsilon^2}{2} \cdot G^{-1}(z)\nabla_z \log \tilde{q}_Z(z) + \epsilon \cdot \sqrt{G^{-1}(z)}\xi \tag{C30}$$

with $\xi \sim \mathcal{N}(0, I)$ *is defined by the transition kernel*

$$q\big(z' \mid z\big) \propto |J_f(z)| \exp\left[-\frac{1}{2\epsilon^2}\left\|J_f(z)(z'-z) + \frac{\epsilon^2}{2}J_f^{-1}(z)\nabla_z \log \tilde{q}(z)\right\|^2\right]. \tag{C31}$$

Proof From the Gaussian nature of ξ, the conditional distribution of z' is a Gaussian distribution whose mean is governed by the drift and covariance matrix is parametrized by the (inverse of) the Jacobian, namely

$$z' \mid z \sim \mathcal{N}(\mu, \Sigma) \tag{C32}$$

with

$$\mu = z + \frac{\epsilon^2}{2} \cdot G^{-1}(z)\nabla_z \log \tilde{q}_Z(z) \tag{C33}$$

$$\Sigma = \epsilon^2 J_f^{-1}(z) J_f^{-\top}. \tag{C34}$$

The corresponding pdf writes

$$q\big(z' \mid z\big) = \left(\frac{1}{2\pi}\right)^{\frac{d}{2}} \frac{1}{|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(z'-\mu)^\top \Sigma^{-1}(z'-\mu)\right). \tag{C35}$$

First, let notice that $\Sigma^{-1} = \epsilon^{-2} J_f^\top(z) J_f(z)$. Then we have

$$(z'-\mu)^\top \Sigma^{-1}(z'-\mu) = \epsilon^{-2}\left[z' - z - \frac{\epsilon^2}{2} \cdot \left[J_f^{-1}(z)\right]^2 \nabla_z \log \tilde{q}_Z(z)\right]^\top J_f^\top(z) \tag{C36}$$

$$\begin{aligned} &\times J_f(z)\left[z' - z - \frac{\epsilon^2}{2} \cdot \left[J_f^{-1}(z)\right]^2 \nabla_z \log \tilde{q}_Z(z)\right] \\ &= \epsilon^{-2}\left\|J_f(z)(z'-z) - \frac{\epsilon^2}{2}J_f^{-1}(z)\nabla_z \log \tilde{q}(z)\right\|^2 \end{aligned} \tag{C37}$$

Finally, using $|\Sigma|^{1/2} = \epsilon |J_f^{-1}(z)| = \epsilon |J_f(z)|^{-1}$ yields

$$q\big(z' \mid z\big) \propto |J_f(z)| \exp\left[-\frac{1}{2\epsilon^2}\left\|J_f(z)(z'-z) - \frac{\epsilon^2}{2}J_f^{-1}(z)\nabla_z \log \tilde{q}(z)\right\|^2\right]. \tag{C38}$$

□

It is worth noting that the pdf of this transition kernel should be computed when evaluating the acceptance ratio (13). When using the canonical writing (C35), evaluating this

pdf would require to compute $G^{-1}(z)$ in (C33) and $J_f^{-1}(z)J_f^{-\top}(z)$ in (C34). Instead, evaluating this pdf with the specific form (C38) only requires to compute $J_f(z)$ since the latent score $\tilde{s}_Z(z) = J_f^{-1}(z)\nabla_z \log \tilde{q}(z)$ can be computed efficiently, as discussed later in Appendix C.3.2.

C.3 Efficient implementation

C.3.1 Approximation of the proposal move

Generating high dimension Gaussian variables according to (8) is expected to be very costly because of the covariance matrix, even if the corresponding Cholesky factor $\epsilon J_f^{-1}(\cdot)$ is lower triangular and explicit (see Appendix B.1). Alternatively, to lighten the computation, we take advantage of the 1st order expansion

$$f^{-1}\big(f(z_k) + \epsilon\xi\big) \simeq f^{-1} \circ f(z_k) + \epsilon J_{f^{-1}}(f(z_k))\xi. \tag{C39}$$

Using Property 2, this amounts to approximate (8) by the diffusion

$$z' = f^{-1}\big(f(z_k) + \epsilon\xi\big) + \frac{\epsilon^2}{2} G^{-1}(z_k)\nabla_z \log \tilde{q}(z) \tag{C40}$$

$$= f^{-1}\big(f(z_k) + \epsilon\xi\big) + \frac{\epsilon^2}{2} J_f^{-1}(z_k)\tilde{s}_Z(z). \tag{C41}$$

According to (C41), this alternative proposal scheme only requires to generate high dimensional Gaussian variables whose covariance matrix is now identity.

C.3.2 Fast computation of the latent score

The latent score $\tilde{s}_Z(z)$ is a critical quantity in the proposed method, as it contributes to the drift term in the proposal move (14) and to the proposal kernel (11). Property 3 shows that the latent score is equal to the score of q_X expressed in the target domain, i.e.,

$$\nabla_x \log q_X(x) = J_f^{-1}(z) \cdot \nabla_z \log \tilde{q}_Z(z)$$

The adopted implementation bypasses the costly evaluation and storage of the inverse Jacobian by directly computing the latent score as $\nabla_x \log q_X(x)$. The evaluation of the score of q_X can be conveniently performed thanks to the auto-differentiation modules provided by numerous deep learning frameworks.

Appendix D: Experiments

D.1 Training

In all experiments we trained our models to maximize either the log-likelihood using the ADAM optimiser with default hyperparameters and no weight decay. We used a held-out validation set and trained each model until its validation score stopped improving, except for the synthetic data experiments where we train for a fixed number of 1000 epochs.

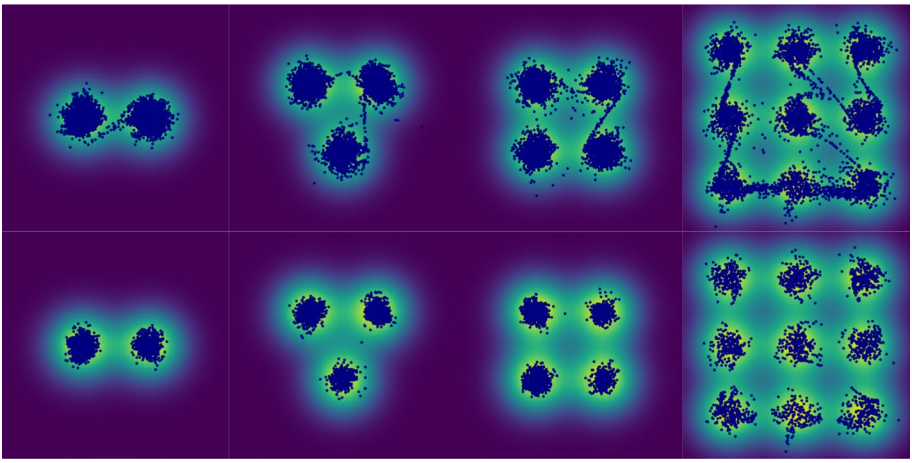


Fig. 6 Mixture of k Gaussian distributions (green), and 1000 samples (blue) generated by the naive sampling (top) and the proposed NF-SAILS method (bottom) with, from left to right, $k = 2$, $k = 3$, $k = 4$ and $k = 9$ (Color figure online)

D.2 Complementary results for the synthetic experiments

Figure 6 shows the difference of sampling quality between naive sampling and the proposed NF-SAILS method for RealNVP model trained on k-mixtures of Gaussians for $k \in \{2, 3, 4, 9\}$. See also Table 1 in Sect. 6.2 of the main document.

D.3 Implementation details for the image experiments

The hyperparameters used in the experiments conducted on images (see Section 6.3 of the main document) are reported in Table 2.

Table 2 Architectures of the Glow model implemented for the experiments conducted on the image data sets

Dataset	Minibatch size	Levels (L)	Depth per level (K)	Coupling
CIFAR-10	512	3	32	Affine
LSUN, 64×64	128	4	48	Affine
LSUN, 96×96	320	5	64	Affine
LSUN, 128×128	160	5	64	Affine
CelebA, 96×96	320	5	64	Affine
CelebA, 128×128	160	6	32	Affine

Author Contributions FC: conceptualization, formal analysis, investigation, methodology, software, validation, visualization, writing (original draft). ND: conceptualization, formal analysis, methodology, project administration, supervision, writing (original draft). PC: conceptualization, formal analysis, methodology, project administration, supervision, writing (original draft). (The categories follow the CRediT taxonomy: http://credit.niso.org/)

Funding Open access funding provided by Institut National Polytechnique de Toulouse. This work was supported by the Artificial Natural Intelligence Toulouse Institute (ANITI, ANR-19-PI3A-0004), the AI Sherlock Chair (ANR-20-CHIA-0031-01), the ULNE national future investment program (ANR-16-IDEX-0004) and the Hauts-de-France Region.

Data availibility The code for generating the synthetic data set is available (see below). The CIFAR-10, LSUN and CelebA data sets are available at https://www.cs.toronto.edu/~kriz/cifar.html, https://trends.openbayes.com/dataset/lsun and https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html, respectively.

Code availability The code (including data generation) will be available at https://github.com/FlorentinCDX/NF-SAILS.

Declarations

Conflict of interest None.

Consent to participate Not applicable.

Consent for publication Not applicable.

Ethics approval Not applicable.

References

Ardizzone, L., Mackowiak, R., Rother, C., & Köthe, U. (2020). Training normalizing flows with the information bottleneck for competitive generative classification. In *Advances in neural information processing systems (NeurIPS)*.

Arjovsky, M., Chintala, S., & Bottou, L. (2017) Wasserstein generative adversarial networks. In: Precup, D., Teh, Y. W. (Eds.) In *Proceedings of international conference on machine learning (ICML)*. PMLR, Proceedings of Machine Learning Research.

Arvanitidis, G., Hansen, L. K., & Hauberg, S. (2018). Latent space oddity: On the curvature of deep generative models. In *Proceedings of IEEE International conference on learning representation (ICLR)*.

Behrmann, J., Vicol, P., Wang, K. C., Grosse, R. B., & Jacobsen, J. H. (2019). On the invertibility of invertible neural networks. https://openreview.net/forum?id=BJlVeyHFwH.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 35*(8), 1798–1828.

Coeurdoux, F., Dobigeon, N., & Chainais, P. (2022) Sliced-Wasserstein normalizing flows: Beyond maximum likelihood training. In *Proceedings of European symposium on artificial neural networks, computational intelligence and machine learning (ESANN)*, Bruges, Belgium.

Cornish, R., Caterini, A., Deligiannidis, G., & Doucet, A. (2020) Relaxing bijectivity constraints with continuously indexed normalising flows. In *Proceedings of international conference machine learning (ICML)*, PMLR, pp. 2133–2143.

Dinh, L., Sohl-Dickstein, J., & Bengio, S. (2016). Density estimation using real NVP. arXiv preprint arXiv:1605.08803

Dudley, R. M. (2002). *Real analysis and probability*. Cambridge University Press.

Gabrié, M., Rotskoff, G. M., & Vanden-Eijnden, E. (2022). Adaptive monte carlo augmented with normalizing flows. *Proceedings of the National Academy of Sciences (PNAS), 119*(10):e2109420119.

Girolami, M., & Calderhead, B. (2011). Riemann manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of the Royal Statistical Society Series B: Statistical Methodology, 73*(2), 123–214.

Gomez, A. N., Ren, M., Urtasun, R., & Grosse, R. B (2017) The reversible residual network: Backpropagation without storing activations. In *Advances in neural information processing systems (NeurIPS)*.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Bing, Xu., David, Warde-Farley., Sherjil, Ozair, Aaron, Courville, & Yoshua, Bengio. (2020). Generative adversarial networks. *Communications of the ACM, 63*(11), 139–144.

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

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, Vincent., & Courville, A. C. (2017). Improved training of Wasserstein GANs. In *Advances in neural information processing systems (NeurIPS)*.

Hagemann, P., & Neumayer, S. (2021) Stabilizing invertible neural networks using mixture models. *Inverse Problems, 37*(8), 085002.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, Bernhard., & Hochreiter, S. (2017). GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in neural information processing systems (NeurIPS)*.

Ho, J., Jain, A., & Abbeel, P. (2020) Denoising diffusion probabilistic models. In *Advances in neural information processing systems (NeurIPS)*, pp. 6840–6851.

Hoffman, M., Sountsov, P., Dillon, J. V., Langmore, I., Tran, D., & Vasudevan, S. (2019). NeuTra-lizing bad geometry in Hamiltonian Monte Carlo using neural transport. In *Proceedings of 1st symposium advances in approximate bayesian inference*.

Huang, C. W., Dinh, L., & Courville, A. (2020). Augmented normalizing flows: Bridging the gap between generative flows and latent variable models. arXiv preprint arXiv:2002.07101.

Issenhuth, T., Tanielian, U., Mary, J., & Picard, D. (2022) On the optimal precision for GANs. arXiv preprint arXiv:2207.10541

Izmailov, P., Kirichenko, P., Finzi, M., & Wilson, A. G. (2020) Semi-supervised learning with normalizing flows. In *Proceedings international conference machine learning (ICML)*, PMLR.

Justel, A., Peña, D., & Zamar, R. (1997). A multivariate Kolmogorov–Smirnov test of goodness of fit. *Statistics & Probability Letters, 35*(3), 251–259.

Kingma, D. P., & Dhariwal, P. (2018) Glow: Generative flow with invertible 1×1 convolutions. In *Advances in neural information processing systems (NeurIPS)*.

Kraskov, A., Stögbauer, H., & Grassberger, P. (2004) Estimating mutual information. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics, 69*(6):066138.

Krizhevsky, A., Nair, V., & Hinton, G. (2010) Cifar-10 (Canadian Institute for Advanced Research).

Kumar, A., Sattigeri, P., & Fletcher, T. (2017) Semi-supervised learning with GANs: Manifold invariance with improved inference. In *Advances in neural information processing systems (NeurIPS)*.

Liu, Z., Luo, P., Wang, X., & Tang, Xiaoou. (2015) Deep learning face attributes in the wild. In *Proceedings of IEEE international conference on computer vision (ICCV)*, pp. 3730–3738.

Marzouk, Y., Moselhy, T., Parno, M., & Alessio, Spantini. (2016). Sampling via measure transport: An introduction. *Handbook of Uncertainty Quantification, 1*, 2.

Milman, E., & Neeman, J. (2022) The Gaussian double-bubble and multi-bubble conjectures. *Annals of Mathematics*, 195.

Müller, T., McWilliams, B., Rousselle, F., Markus, Gross, & Jan, Novák. (2019). Neural importance sampling. *ACM Transactions on Graphics (ToG), 38*(5), 1–19.

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

Nielsen, D., Jaini, P., Hoogeboom, E., Winther, O., & Welling, M. (2020). SurVAE flows: Surjections to bridge the gap between VAEs and flows. In *Advances in neural information processing systems (NeurIPS)*.

Noé, F., Olsson, S., Köhler, J., & Wu, H. (2019) Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science 365*(6457), eaaw1147.

Øksendal, B., & Øksendal, B. (2003). *Stochastic differential equations*. Springer.

Papamakarios, G., Pavlakou, T., & Murray, I. (2017) Masked autoregressive flow for density estimation. In *Advances in neural information processing systems (NeurIPS)*.

Papamakarios, G., Nalisnick, E. T., Rezende, D. J., Shakir, Mohamed, & Balaji, Lakshminarayanan. (2021). Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research, 22*(57), 1–64.

Pillai, N. S., Stuart, A. M., & Thiéry, A. H. (2012). Optimal scaling and diffusion limits for the Langevin algorithm in high dimensions. *The Annals of Applied Probability, 22*(6), 2320–2356.

Pires, G. G., & Figueiredo, M. A. (2020) Variational mixture of normalizing flows. In *Proceedings of European symposium on artificial neural networks, computational intelligence and machine learning (ESANN)*.

Rifai, S., Mesnil, G., Vincent, P., Muller, X., Bengio, Y., Dauphin, Y., & Glorot, X. (2011) Higher order contractive auto-encoder. In *Proceedings of European conference on machine learning and principles and practice of knowledge discovery in databases (ECML-PKDD)*, Springer.

Runde, V., Ribet, K., & Axler, S. (2005). *A taste of topology*. Springer.

Samsonov, S., Lagutin, E., Gabrié, M., Durmus, A., Naumov, A., & Moulines, E. (2022). Local-global MCMC kernels: The best of both worlds. In *Advances in neural information processing systems (NeurIPS)*, pp. 5178–5193.

Stimper, V., Schölkopf, B., & Hernández-Lobato, J. M. (2022) Resampling base distributions of normalizing flows. In *Proceedings of international conference on artificial intelligence and statistics (AISTATS)*, PMLR, pp. 4915–4936.

Vono, M., Dobigeon, N., & Chainais, P. (2022). High-dimensional Gaussian sampling: A review and a unifying approach based on a stochastic proximal point algorithm. *SIAM Review, 64*(1), 3–56.

Wu, H., Köhler, J., & Noé, F. (2020) Stochastic normalizing flows. In *Advances in neural information processing systems (NeurIPS)*.

Xifara, T., Sherlock, C., Livingstone, S., Simon, Byrne, & Mark, Girolami. (2014). Langevin diffusions and the Metropolis-adjusted Langevin algorithm. *Statistics & Probability Letters, 91*, 14–19.

Yu, F., Zhang, Y., Song, S., Seff, A., & Xiao, J. (2015) LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365