Introduction to Stochastic Normalizing Flow

Xuda Ye

Purdue University

December 6, 2025

In this note, we introduce the Stochastic Normalizing Flow (SNF) [1] for sampling a Boltzmann distribution $\pi(x) \propto e^{-U(x)}$, where $U(x)$ is a possibly nonconvex potential function defined on $\mathbb{R}^d$. We begin by reviewing the Boltzmann generator realized with Normalizing Flows (NF) [2]. The SNF can be viewed as a generalization of the NF achieved by inserting additional diffusion steps. The goal of these diffusion steps is to overcome the restrictions imposed by the diffeomorphic nature of the NF and to increase the representation capacity of the model. The numerical results in this note are directly cited from [1].

The primary motivation for this note is that [1], being a conference paper, does not fully detail the mathematical intricacies of the method. While a subsequent study [3] provides a rigorous probabilistic perspective, its abstract and general setting may obscure the underlying mathematical intuition. Consequently, this note aims to provide an intuitive and comprehensive derivation of the SNF framework, with particular emphasis on the loss function. We also establish a direct connection between SNF and classical NF by adopting a consistent notation system. Furthermore, by initiating the discussion with a simplified setup, we introduce the method from a pedagogical viewpoint.

1 Problem Statement

Efficiently sampling from a Boltzmann distribution of the form $\pi(x) \propto e^{-U(x)}$ is a central problem in computational statistical physics. Assume that $U(x)$ is a confining potential function on $\mathbb{R}^d$ with $\mathcal{Z} = \int_{\mathbb{R}^d} e^{-U(x)} \mathrm{d}x < +\infty$, such that the target distribution $\pi(x)$ can be written as

$$\pi(x) = \frac{1}{\mathcal{Z}} e^{-U(x)}, \quad x \in \mathbb{R}^d. \tag{1}$$

Classically, the Langevin dynamics serves as a standard approach to sample from $\pi(x)$:

$$\mathrm{d}x_t = -\nabla U(x_t)\mathrm{d}t + \sqrt{2}\mathrm{d}B_t, \tag{2}$$

where B_t denotes a standard Brownian motion. However, this method often suffers from the high computational cost associated with the discretization of the Langevin dynamics (2). The seminal work [2] proposes a revolutionary approach, known as the Boltzmann generator, to directly sample from $\pi(x)$. Specifically, assume there exists a simple reference distribution $\pi_0(z) \propto e^{-U_0(z)}$, where

$U_0(z)$ is a confining potential on $\mathbb{R}^d$. By defining the constant $\mathcal{Z}_0 = \int_{\mathbb{R}^d} e^{-U_0(z)} \mathrm{d}z < +\infty$, the reference distribution $\pi_0(z)$ can be written as

$$\pi_0(z) = \frac{1}{\mathcal{Z}_0} e^{-U_0(z)}, \quad z \in \mathbb{R}^d. \tag{3}$$

The reference $\pi_0(z)$ is typically chosen as the standard normal distribution $\mathcal{N}(0, I_d)$. The Boltzmann generator aims to construct a mapping $T : \mathbb{R}^d \to \mathbb{R}^d$ satisfying

$$T_{\#}\pi_0 = \pi. \tag{4}$$

Consequently, for any random variable $Z \sim \pi_0$, one need only apply the map T to obtain samples $T(Z)$ distributed according to π. This procedure requires only sampling from the reference distribution π_0 and evaluating the map T, thereby bypassing the need to simulate Langevin dynamics.

The SNF generalizes this framework by formulating $T : \mathbb{R}^d \to \mathcal{P}(\mathbb{R}^d)$ as a Markov transition kernel. This kernel assigns to each state $z \in \mathbb{R}^d$ a conditional probability measure $T(z, \cdot) \in \mathcal{P}(\mathbb{R}^d)$. In this context, the condition $T_{\#}\pi_0 = \pi$ in (4) is understood in the distributional sense: if Z is a random variable distributed according to π_0, and X is sampled conditionally such that $X|Z \sim T(Z, \cdot)$, then the marginal distribution of X satisfies $X \sim \pi$.

2 Normalizing Flow as Boltzmann Generators

2.1 Derivation of loss function

The mapping $T : \mathbb{R}^d \to \mathbb{R}^d$ transforms the reference distribution $\pi_0(z)$ into the target distribution $\pi(x)$. Let $JT : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ denote the Jacobian matrix of T. Under the change of variable $x = T(z)$, the density function of the push-forward distribution $T_{\#}\pi_0$ is given by

$$(T_{\#}\pi_0)(x) = \frac{\pi_0(z)}{|\det JT(z)|} = \frac{1}{\mathcal{Z}_0} \frac{\exp(-U_0(z))}{|\det JT(z)|}, \quad x = T(z). \tag{5}$$

The KL divergence between $T_{\#}\pi_0$ and π is then computed as

$$\begin{aligned}
D_{\mathrm{KL}}(T_{\#}\pi_0 \| \pi) &= \int_{\mathbb{R}^d} (T_{\#}\pi_0)(x) \log \frac{(T_{\#}\pi_0)(x)}{\pi(x)} \mathrm{d}x \\
&= \int_{\mathbb{R}^d} \pi_0(z) \log \frac{(T_{\#}\pi_0)(T(z))}{\pi(T(z))} \mathrm{d}z && (\text{Set } x = T(z)) \\
&= \int_{\mathbb{R}^d} \pi_0(z) \log \left(\frac{1}{\mathcal{Z}_0} \frac{\exp(-U_0(z))}{|\det JT(z)|} \Big/ \frac{1}{\mathcal{Z}} \exp\big(-U(T(z))\big) \right) \mathrm{d}z && (\text{Substitute } (5)) \\
&= \int_{\mathbb{R}^d} \pi_0(z) \Big(U(T(z)) - U_0(z) - \log|\det JT(z)| \Big) \mathrm{d}z + \log \frac{\mathcal{Z}}{\mathcal{Z}_0} \\
&= \mathbb{E}_{Z\sim\pi_0} \Big[U(T(Z)) - U_0(Z) - \log|\det JT(Z)| \Big] + \log \frac{\mathcal{Z}}{\mathcal{Z}_0} + \text{constant}.
\end{aligned}$$

Here, we explicitly distinguish the random variable Z sampled from π_0 from the integration variable z. Consequently, the loss function (with respect to the map T) is defined as

$$\mathcal{L}[T] = \mathbb{E}_{Z\sim\pi_0} \Big[U(T(Z)) - U_0(Z) - \log|\det JT(Z)| \Big]. \tag{6}$$

By minimizing $\mathcal{L}[T]$ with respect to T, we obtain the desired map satisfying condition (4).

We emphasize that the map satisfying condition (4) is generally **non-unique**. For instance, when the mapping T is constructed via an ODE flow, the loss function $D_{\mathrm{KL}}(T_\# \pi_0 \| \pi)$ constrains only the marginal distribution at the terminal state, without imposing restrictions on the intermediate dynamics of the flow. This phenomenon is analogous to the framework of Flow Matching [4], where distinct couplings between the reference and target distributions yield different ODE flows.

2.2 Coupling flow parameterization of T

Next, we detail the parameterization of the mapping $T : \mathbb{R}^d \to \mathbb{R}^d$. Employing the classical discrete coupling flow approach, we select a positive integer K and decompose T as

$$T = f_K \circ \cdots \circ f_1,$$

where $\{f_k\}_{k=1}^K$ are invertible maps on $\mathbb{R}^d$ (representing the layers of the coupling flow). These maps can be effectively implemented using RealNVP [5]. We can express the mapping chain $x = T(z)$ as

$$y_0 = z \xrightarrow{f_1} y_1 \xrightarrow{f_2} y_2 \xrightarrow{\cdots} y_{K-1} \xrightarrow{f_K} y_K = x, \tag{7}$$

where we have introduced the sequence of intermediate states $\{y_k\}_{k=0}^K$ for clarity. By the chain rule, the Jacobian matrix JT evaluated at $y_0 = z$ is given by

$$JT(y_0) = \prod_{k=1}^K Jf_k(y_{k-1}) \Longrightarrow \det JT(y_0) = \prod_{k=1}^K \det Jf_k(y_{k-1}).$$

In the construction of the coupling flow, we require that each layer f_k on $\mathbb{R}^d$ has a positive Jacobian determinant. This allows us to write

$$\log\det JT(y_0) = \sum_{k=1}^K \log\det Jf_k(y_{k-1}). \tag{8}$$

Substituting (8) into the loss function (6), we obtain the loss function in terms of $\{f_k\}_{k=1}^K$:

$$\boxed{\mathcal{L}\big[\{f_k\}_{k=1}^K\big] = \mathbb{E}_{Y_0\sim\pi_0}\Big[U(Y_K) - U_0(Y_0) - \sum_{k=1}^K \log\det Jf_k(Y_{k-1})\Big],} \tag{9}$$

where the random variables $\{Y_k\}_{k=0}^K$ are defined recursively by $Y_0 \sim \pi_0$ and $Y_k = f_k(Y_{k-1})$ for $k = 1, \cdots, K$. We visualize the chain of random variables corresponding to (7) as follows:

$$Y_0 \sim \pi_0 \xrightarrow{f_1} Y_1 \xrightarrow{f_2} Y_2 \xrightarrow{\cdots} Y_{K-1} \xrightarrow{f_K} Y_K \sim \pi. \tag{10}$$

Upon minimizing the loss function (9), any input $Y_0 \sim \pi_0$ is expected to yield an output Y_K distributed according to π, thereby enabling efficient sampling from the target distribution $\pi(x)$.

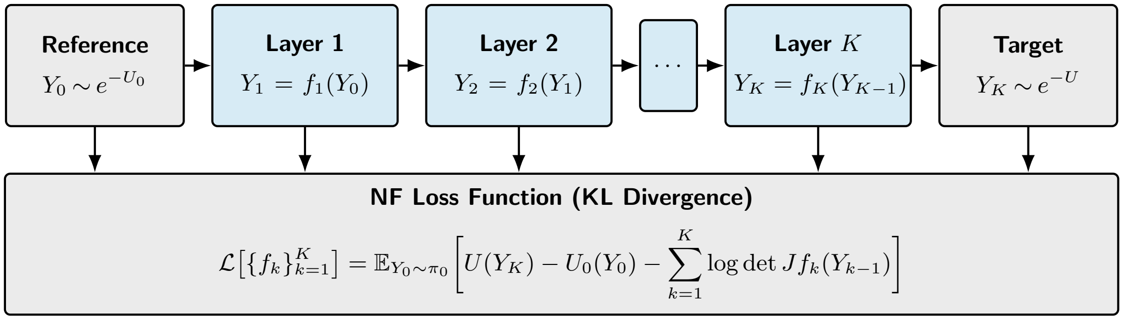
Reference
$Y_0 \sim e^{-U_0}$
Layer 1
$Y_1 = f_1(Y_0)$
Layer 2
$Y_2 = f_2(Y_1)$
$\cdots$
Layer K
$Y_K = f_K(Y_{K-1})$
Target
$Y_K \sim e^{-U}$
NF Loss Function (KL Divergence)
$\mathcal{L}\big[\{f_k\}_{k=1}^K\big] = \mathbb{E}_{Y_0 \sim \pi_0}\Big[U(Y_K) - U_0(Y_0) - \sum_{k=1}^{K} \log\det Jf_k(Y_{k-1})\Big]$

Figure 1: Normalizing Flow as Boltzmann Generators.

3 Stochastic Normalizing Flow

Normalizing Flows (NF) have proven to be a successful method for sampling from a target distribution $\pi(x)$. However, their performance is sometimes hindered by the inherent restrictions of diffeomorphisms, which strictly preserve the topology of the support. Drawing upon the success of Score-Based Diffusion models [6], it is widely acknowledged that incorporating stochastic diffusion into the generative dynamics can significantly expand the representational capacity of the model. Therefore, a natural motivation for the Stochastic Normalizing Flow (SNF) is to introduce artificial diffusion steps to augment the expressivity of standard NFs.

3.1 Revisiting the chain structure

In the standard NF framework, the sequence of random variables $\{Y_k\}_{k=0}^K$ in (10) is propagated via K invertible maps $\{f_k\}_{k=1}^K$. We now introduce K additional Markov transition steps $\{g_k\}_{k=1}^K$, where each g_k is characterized by a conditional probability kernel $g_k(x, \cdot)$. These Markovian kernels $\{g_k\}_{k=1}^K$ are fixed prior to training, such that only the invertible mappings $\{f_k\}_{k=1}^K$ remain parameterized. The resulting chain is constructed as

$$Y_0 \sim \pi_0 \xrightarrow{g_1 \circ f_1} Y_1 \xrightarrow{g_2 \circ f_2} Y_2 \xrightarrow{\cdots} Y_{K-1} \xrightarrow{g_K \circ f_K} Y_K \sim \pi. \tag{11}$$

More precisely, the random variables $\{Y_k\}_{k=0}^K$ are generated recursively by sampling $Y_0 \sim \pi_0$ and $Y_k \sim g_k\big(f_k(Y_{k-1}), \cdot\big)$ for $k = 1, \cdots, K$. In this framework, the joint probability density of the trajectory $(Y_0, Y_1, \cdots, Y_K)$ is given by

$$p(y_0, y_1, \cdots, y_K) = \pi_0(y_0) \prod_{k=1}^{K} g_k\big(f_k(y_{k-1}), y_k\big). \tag{12}$$

Consequently, the marginal distribution of the final state Y_K is expressed as

$$p_K(y_K) = \int_{\mathbb{R}^{dK}} \pi_0(y_0) \prod_{k=1}^{K} g_k\big(f_k(y_{k-1}), y_k\big) \mathrm{d}y_0 \cdots \mathrm{d}y_{K-1}. \tag{13}$$

Intuitively, one might select the KL divergence between the marginal distribution p_K and the target distribution π as the loss function. However, the expression for p_K in (13) involves a highdimensional integral, rendering the explicit evaluation of the KL divergence computationally intractable. Instead, it is advantageous to analyze the distribution of the entire trajectory $\{Y_k\}_{k=0}^K$, whose joint density (12) explicitly incorporates all intermediate states. To this end, we introduce a sequence of guiding distributions $\{\pi_k\}_{k=0}^K$ interpolating between the reference distribution π_0 and the target distribution $\pi_K = \pi$. Furthermore, we require that each Markov kernel g_k preserves π_k as its invariant distribution, i.e.,

$$\int_{\mathbb{R}^d} \pi_k(x) g_k(x, y) \mathrm{d}x = \pi_k(y). \tag{14}$$

Consequently, we impose a stronger objective on the random variable chain defined in (11):

$$Y_0 \sim \pi_0 \xrightarrow{g_1 \circ f_1} Y_1 \sim \pi_1 \xrightarrow{g_2 \circ f_2} Y_2 \sim \pi_2 \xrightarrow{\cdots} Y_{K-1} \sim \pi_{K-1} \xrightarrow{g_K \circ f_K} Y_K \sim \pi_K = \pi, \tag{15}$$

which necessitates control over the marginal distributions of the intermediate states $\{Y_k\}_{k=0}^K$. A challenge arises here: while the joint distribution of the generated path $(Y_0, Y_1, \cdots, Y_K)$ is explicitly given by (12), a corresponding joint distribution for the target sequence $(\pi_0, \pi_1, \cdots, \pi_K)$ is undefined, as the optimal coupling between these marginals is unknown.

Nevertheless, the chain depicted in (15) encapsulates a precise mathematical objective:

Given a sequence of guiding distributions $\{\pi_k\}_{k=0}^K$ and the corresponding invariant Markov kernels $\{g_k\}_{k=1}^K$, we aim to train the invertible maps $\{f_k\}_{k=1}^K$ such that for the Markov chain $\{Y_k\}_{k=0}^K$ generated by $Y_0 \sim \pi_0$ and $Y_k \sim g_k\big(f_k(Y_{k-1}), \cdot\big)$, there hold $f_k(Y_{k-1}) \sim \pi_k$ for all $k = 1, \cdots, K$.

We observe that since g_k preserves π_k as the invariant distribution as specified in (14), the condition $f_k(Y_{k-1}) \sim \pi_k$ immediately implies $Y_k \sim \pi_k$, which aligns with the chain requirements in (15).

A natural choice for the guiding distributions $\{\pi_k\}_{k=0}^K$ is given by

$$\pi_k(x) \propto \exp\left(-\frac{k}{K} U(x) - \frac{K-k}{K} U_0(x) \right), \quad x \in \mathbb{R}^d, \tag{16}$$

which functions as a linear interpolation of the potential function. The specific choice of the Markov kernels $\{g_k\}_{k=1}^K$ will be detailed in subsequent sections.

3.2 Loss function over forward and backward chains

A key innovation of the SNF framework [3] is the introduction of a backward chain of random variables $\{Y_k'\}_{k=0}^K$, which facilitates the formulation of the loss function. We begin by recalling the forward chain structure defined in (15):

Forward chain $\{Y_k\}_{k=0}^K$

$$Y_0 \sim \pi_0 \xrightarrow{g_1 \circ f_1} Y_1 \sim \pi_1 \xrightarrow{g_2 \circ f_2} Y_2 \sim \pi_2 \xrightarrow{\cdots} Y_{K-1} \sim \pi_{K-1} \xrightarrow{g_K \circ f_K} Y_K \sim \pi_K.$$

This chain is governed by the invertible maps $\{f_k\}_{k=1}^K$ and the Markov kernels $\{g_k\}_{k=1}^K$.

Subsequently, we introduce a set of backward Markov kernels $\{h_k\}_{k=1}^K$ to define the backward chain $\{Y_k'\}_{k=0}^K$:

Backward chain $\{Y_k'\}_{k=0}^K$

$$Y_K' \sim \pi_K \xrightarrow{f_K^{-1}\circ h_K} Y_{K-1}' \sim \pi_{K-1} \xrightarrow{f_{K-1}^{-1}\circ h_{K-1}} Y_{K-2}' \sim \pi_{K-2} \xrightarrow{\cdots} Y_1' \sim \pi_1 \xrightarrow{f_1^{-1}\circ h_1} Y_0' \sim \pi_0.$$

More precisely, the generative processes for the random variables $\{Y_k\}_{k=0}^K$ and $\{Y_k'\}_{k=0}^K$ are defined as follows:

- **Forward:** Initialize $Y_0 \sim \pi_0$. For $k = 1, \cdots, K$, sample $Y_k \sim g_k\big(f_k(Y_{k-1}), \cdot\big)$.
- **Backward:** Initialize $Y_K' \sim \pi_K$. For $k = K, \cdots, 1$, sample an intermediate variable from $h_k(Y_k', \cdot)$ and apply f_k^{-1} to obtain Y_{k-1}'. Equivalently, this implies $f_k(Y_{k-1}') \sim h_k(Y_k', \cdot)$.

Accordingly, the joint probability density of the forward trajectory $(Y_0, Y_1, \cdots, Y_K)$, previously introduced in (12), is denoted by p^f:

$$p^f(y_0, y_1, \cdots, y_K) = \pi_0(y_0) \prod_{k=1}^K g_k\big(f_k(y_{k-1}), y_k\big). \tag{17}$$

Similarly, we formulate the joint probability density p^b for the backward chain $\{Y_k'\}_{k=0}^K$ as:

$$p^b(y_0, y_1, \cdots, y_K) = \pi(y_K) \prod_{k=1}^K h_k\big(y_k, f_k(y_{k-1})\big) \cdot \det Jf_k(y_{k-1}). \tag{18}$$

Note that the Jacobian determinants appear in (18) due to the change of variables induced by applying the inverse maps f_k^{-1} to the samples generated by the kernels $\{h_k\}_{k=1}^K$.

The guiding philosophy in designing the loss function is to match the forward and backward probability densities, p^f and p^b. This objective implies a crucial relationship between the two Markov chains $\{Y_k\}_{k=0}^K$ and $\{Y_k'\}_{k=0}^K$, specifically regarding **reversibility**. For the deterministic transport steps, the reverse of f_k is naturally given by its inverse map f_k^{-1}. For the Markov transition steps, we require that the forward kernel g_k and the backward kernel h_k satisfy the following reversibility condition with respect to the guiding distribution π_k:

$$\pi_k(x) g_k(x, y) = \pi_k(y) h_k(y, x), \quad x, y \in \mathbb{R}^d. \tag{19}$$

We observe that the invariant distribution property in (14) is a direct consequence of (19), obtained by integrating both sides of the equation with respect to x. Consequently, when the mappings $\{f_k\}_{k=1}^K$ are trained such that the forward probability p^f and the backward probability p^b align perfectly, the marginal distribution of the forward chain's output Y_K coincides with that of the backward chain's initialization Y_K', which is precisely the target distribution π.

To this end, we define the loss function as the KL divergence between the forward probability density p^f and the backward probability density p^b. The density ratio of p^f and p^b is given by

$$\frac{p^f(y_0, y_1, \cdots, y_K)}{p^b(y_0, y_1, \cdots, y_K)} = \frac{\pi_0(y_0) \displaystyle\prod_{k=1}^K g_k\big(f_k(y_{k-1}), y_k\big)}{\pi(y_K) \displaystyle\prod_{k=1}^K h_k\big(y_k, f_k(y_{k-1})\big) \cdot \det Jf_k(y_{k-1})} = \frac{\pi_0(y_0) \displaystyle\prod_{k=1}^K \pi_k\big(f_k(y_{k-1})\big)}{\pi(y_K) \displaystyle\prod_{k=1}^K \pi_k(y_k) \cdot \det Jf_k(y_{k-1})},$$

where we have applied the reversibility condition stated in (19). Consequently, the log-density ratio is derived as

$$\log \frac{p^f(y_0, y_1, \cdots, y_K)}{p^b(y_0, y_1, \cdots, y_K)} = U(y_K) - U_0(y_0) + \sum_{k=1}^{K} \left(U_k(y_k) - U_k(f_k(y_{k-1}))\right) - \sum_{k=1}^{K} \log\det Jf_k(y_{k-1}) + \text{constant}. \tag{20}$$

Therefore, the KL divergence between p^f and p^b is expressed as

$$D_{\mathrm{KL}}(p^f \| p^b) = \int_{\mathbb{R}^{d(K+1)}} p^f(y_0, y_1, \cdots, y_K) \mathrm{d}y_0 \mathrm{d}y_1 \cdots \mathrm{d}y_K \times$$
$$\left(U(y_K) - U_0(y_0) + \sum_{k=1}^{K} \left(U_k(y_k) - U_k(f_k(y_{k-1}))\right) - \sum_{k=1}^{K} \log\det Jf_k(y_{k-1}) \right) + \text{constant}.$$

Remarkably, this final expression for $D_{\mathrm{KL}}(p^f \| p^b)$ does not explicitly depend on the Markov transition kernels $\{g_k\}_{k=1}^K$ and $\{h_k\}_{k=1}^K$, implying that the only constraint on these kernels is the reversibility condition (19). Finally, the loss function is formally constructed as

$$\boxed{\mathcal{L}[\{f_k\}_{k=1}^K] = \mathbb{E}_{p^f}\left[U(Y_K) - U_0(Y_0) + \sum_{k=1}^{K} \left(U_k(Y_k) - U_k(f_k(Y_{k-1}))\right) - \sum_{k=1}^{K} \log\det Jf_k(Y_{k-1}) \right].} \tag{21}$$

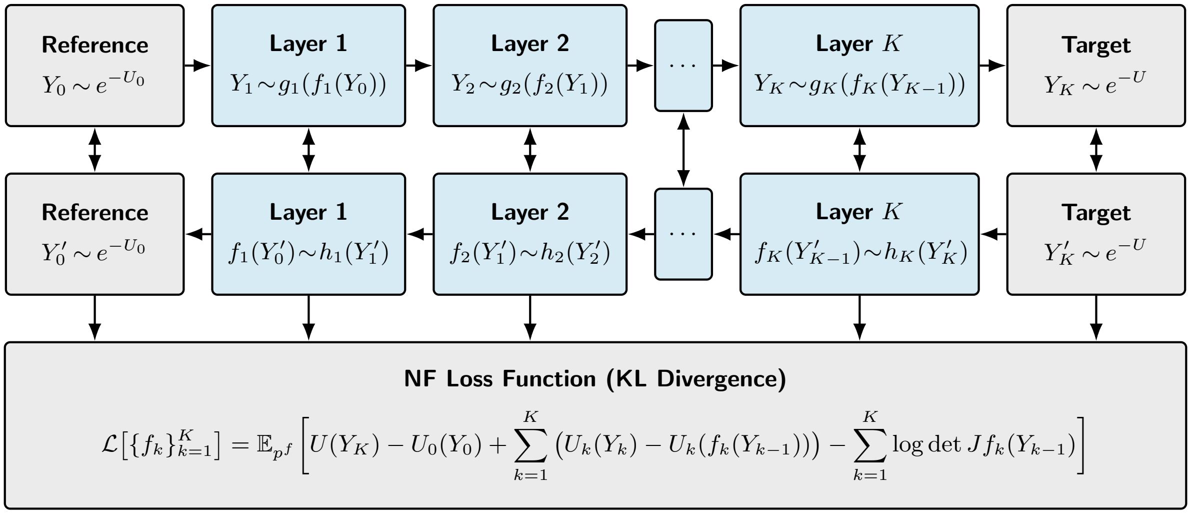
Reference
$Y_0 \sim e^{-U_0}$
Layer 1
$Y_1 \sim g_1(f_1(Y_0))$
Layer 2
$Y_2 \sim g_2(f_2(Y_1))$
$\cdots$
Layer K
$Y_K \sim g_K(f_K(Y_{K-1}))$
Target
$Y_K \sim e^{-U}$
Reference
$Y'_0 \sim e^{-U_0}$
Layer 1
$f_1(Y'_0) \sim h_1(Y'_1)$
Layer 2
$f_2(Y'_1) \sim h_2(Y'_2)$
$\cdots$
Layer K
$f_K(Y'_{K-1}) \sim h_K(Y'_K)$
Target
$Y'_K \sim e^{-U}$
NF Loss Function (KL Divergence)
$\mathcal{L}[\{f_k\}_{k=1}^K] = \mathbb{E}_{p^f}\left[U(Y_K) - U_0(Y_0) + \sum_{k=1}^{K} \left(U_k(Y_k) - U_k(f_k(Y_{k-1}))\right) - \sum_{k=1}^{K} \log\det Jf_k(Y_{k-1}) \right]$

Figure 2: Stochastic Normalizing Flow.

We note that the loss function (21) exhibits a close connection to the standard NF case (9). When the forward and backward kernels are chosen as Dirac distributions, i.e., $g_k(x, y) = h_k(x, y) = \delta(x - y)$, the reversibility condition (19) is automatically satisfied, and the SNF loss function (21) reduces to (9). In this sense, the SNF framework serves as a direct generalization of NF. However, regarding computational complexity, SNF incurs an additional cost due to the evaluation of $U_k(Y_k) - U_k(f_k(Y_{k-1}))$ for the intermediate random variables $\{Y_k\}_{k=1}^K$. Therefore, for complex potential functions, training an SNF model may be more computationally expensive.

3.3 Choice of Markov kernels via detailed balance

We discuss the selection of the forward and backward transition kernels, denoted by g_k and h_k, respectively. Following the approach suggested in [3], we set $g_k = h_k$, thereby utilizing the same transition kernel for both the forward and backward chains. Consequently, the reversibility condition simplifies to

$$\pi_k(x)g_k(x,y) = \pi_k(y)g_k(y,x), \tag{22}$$

which is commonly referred to as the detailed balance condition in the study of Markov Chain Monte Carlo (MCMC). A standard construction of $g_k(x,y)$ is provided by the Metropolis–Hastings algorithm. We first select a symmetric proposal kernel $q_k(x,y)$ satisfying $q_k(x,y) = q_k(y,x)$, and subsequently define $g_k(x,y)$ as

$$g_k(x,y) = q_k(x,y)\alpha_k(x,y) + \delta(x-y)(1-\alpha_k(x,y)), \tag{23}$$

where $\delta(x-y)$ denotes the Dirac distribution, and the acceptance probability $\alpha_k(x,y)$ is given by

$$\alpha_k(x,y) = \min\left\{1, \frac{\pi_k(y)}{\pi_k(x)}\right\}. \tag{24}$$

Provided that the proposal kernel $q_k(x,y)$ is diffusive (e.g., a standard heat kernel), the resulting transition kernel $g_k(x,y)$ will exhibit diffusion effects.

4 Numerical Experiment

In this section, we present a numerical experiment originally detailed in [1] to demonstrate the superior expressive capacity of SNF compared to standard NF. The objective is to construct a generative map that transforms a unimodal standard Gaussian reference distribution $\pi_0 = \mathcal{N}(0, I)$ into a complex target distribution $\pi(x) \propto e^{-U(x)}$ characterized by a double-well potential $U(x)$. This target distribution features two distinct modes separated by a energy barrier.

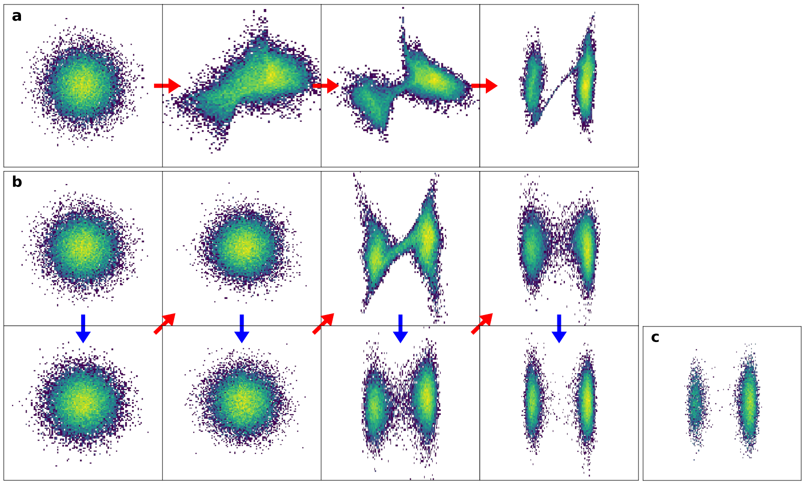

Figure 3: Comparison of generated samples on a double-well potential. **a.** NF output. **b.** SNF output. **c.** Ground truth target distribution.

The results, illustrated in the figure above, highlight a fundamental topological limitation of NFs. As shown in panel **a**, the output of the NF exhibits an artificial “bridge” connecting the two modes. This phenomenon arises because NFs are constructed as diffeomorphisms—continuous, differentiable, and invertible maps. A diffeomorphism must strictly preserve the topology of the support; therefore, it cannot transform a connected domain (the single-mode Gaussian) into a disconnected or effectively separated domain (the double-well target) without stretching probability mass across the void.

In contrast, the SNF (panel **b**) successfully reproduces the separated bimodal structure of the target distribution (panel **c**). By incorporating stochastic Markov transition kernels $\{g_k\}_{k=1}^K$ between the invertible layers, the SNF relaxes the strict topological constraints of the diffeomorphism. The stochastic diffusion steps allow probability mass to “jump” across the energy barrier, enabling the model to accurately capture multimodal distributions without introducing spurious connections between components.

5 Summary

In this note, we introduced the Stochastic Normalizing Flow (SNF) as a generalization of the Boltzmann generator, combining invertible flows with stochastic transitions.

The primary advantage of SNF over NF is topological flexibility. NFs are limited by diffeomorphisms and often create artificial “bridges” between separated modes. SNF overcomes this via stochastic jumps, allowing accurate modeling of multimodal targets. Additionally, the reversibility condition on the path level enhances expressivity.

Conversely, SNF is computationally more demanding. It requires evaluating the potential energy at every intermediate step, whereas NF evaluates it only once. Furthermore, the stochastic nature renders the exact marginal likelihood intractable, preventing the exact density estimation available in standard flows.

References

[1] Hao Wu, Jonas Köhler, and Frank Noé. Stochastic normalizing flows. *Advances in neural information processing systems*, 33:5933–5944, 2020.

[2] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

[3] Paul Hagemann, Johannes Hertrich, and Gabriele Steidl. Stochastic normalizing flows for inverse problems: A markov chains viewpoint. *SIAM/ASA Journal on Uncertainty Quantification*, 10(3):1162–1190, 2022.

[4] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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

[6] Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in neural information processing systems*, 34:1415–1428, 2021.