Invertible Flow Non Equilibrium sampling - SUPPLEMENTARY DOCUMENT

Anonymous Authors[1]

S1. Proofs of Section 2

S1.1. Proof of Equation (5)

Let $f : \mathbb{R}^d \to \mathbb{R}_+$ be a measurable function and $k \in \{0, \ldots, K\}$. Denote $\boldsymbol{\rho}_k(f) = \int f(\mathrm{T}^k(x))\mathbb{1}_\mathrm{I}(x,k)\rho(x)\mathrm{d}x$. Using the change of variable $y = \mathrm{T}^k(x)$, and since by definition of the set I, $\mathbb{1}_\mathsf{O}(\mathrm{T}^{-k}(y))\mathbb{1}_\mathrm{I}(\mathrm{T}^{-k}(y),k) = \mathbb{1}_\mathsf{O}(y)\mathbb{1}_\mathrm{I}(y,-k)$, we obtain

$$\begin{aligned}\tilde{\boldsymbol{\rho}}_k(f) &= \int f(y)\rho(\mathrm{T}^{-k}(y))\mathbb{1}_\mathsf{O}(\mathrm{T}^{-k}(y))\mathbb{1}_\mathrm{I}(\mathrm{T}^{-k}(y),k)|\mathbf{J}_{\mathrm{T}^{-k}}(y)|\mathrm{d}y \\ &= \int f(y)\rho(\mathrm{T}^{-k}(y))\mathbb{1}_\mathsf{O}(y)\mathbb{1}_\mathrm{I}(y,-k)|\mathbf{J}_{\mathrm{T}^{-k}}(y)|\mathrm{d}y\ ,\end{aligned}$$

which concludes the proof.

S1.2. Proof of Theorem 1

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a measurable function. Since ρ_k is the pushforward measure of $x \mapsto \rho(x)\mathbb{1}_\mathrm{I}(x,k)$ by T^k,

$$\begin{aligned}\int f(x)\rho(x)\mathrm{d}x &= \int f(x)\frac{\rho(x)}{\rho_\mathrm{T}(x)}\rho_\mathrm{T}(x)\mathrm{d}x \\ &= \frac{1}{Z_\mathrm{T}}\sum_{k=0}^{K}\int f(x)\frac{\rho(x)}{\rho_\mathrm{T}(x)}\rho_k(x)\mathrm{d}x = \frac{1}{Z_\mathrm{T}}\sum_{k=0}^{K}\int f(\mathrm{T}^k(x))\frac{\rho(\mathrm{T}^k(x))}{\rho_\mathrm{T}(\mathrm{T}^k(x))}\mathbb{1}_\mathrm{I}(x,k)\rho(x)\mathrm{d}x \\ &= \sum_{k=0}^{K}\int f(\mathrm{T}^k(x))w_k(x)\rho(x)\mathrm{d}x\ .\end{aligned}$$

S1.3. Proof of Lemma 2

We need to show that for any $x \in \mathsf{O}$, $k \in \{0, \ldots, K\}$

$$\mathbb{1}_\mathrm{I}(x,k)\sum_{i=0}^{K}\rho_i(\mathrm{T}^k(x)) = \frac{\mathbb{1}_\mathrm{I}(x,k)}{|\mathbf{J}_{\mathrm{T}^k}(x)|}\sum_{j=-k}^{K-k}\rho_j(x)\ .$$

Using the identity $|\mathbf{J}_{\mathrm{T}^{i+k}}(x)| = |\mathbf{J}_{\mathrm{T}^i}(\mathrm{T}^k(x))||\mathbf{J}_{\mathrm{T}^k}(x)|$, we obtain

$$\begin{aligned}\mathbb{1}_\mathrm{I}(x,k)\sum_{i=0}^{K}\rho_i(\mathrm{T}^k(x)) &= \sum_{i=0}^{K}\mathbb{1}_\mathrm{I}(x,k)\rho(\mathrm{T}^i(\mathrm{T}^k(x)))\mathbf{J}_{\mathrm{T}^i}(\mathrm{T}^k(x))\mathbb{1}_\mathrm{I}(\mathrm{T}^k(x)),i) \\ &= \frac{1}{\mathbf{J}_{\mathrm{T}^k}(x)}\sum_{i=0}^{K}\mathbb{1}_\mathrm{I}(x,k)\rho(\mathrm{T}^{i+k}(x))\mathbf{J}_{\mathrm{T}^{i+k}}(x)\mathbb{1}_\mathrm{I}(\mathrm{T}^k(x)),i) \\ &= \frac{1}{\mathbf{J}_{\mathrm{T}^k}(x)}\sum_{j=-k}^{K-k}\rho(\mathrm{T}^j(x))\mathbf{J}_{\mathrm{T}^j}(x)\mathbb{1}_\mathrm{I}(\mathrm{T}^k(x),j-k)\mathbb{1}_\mathrm{I}(x,k)\end{aligned}$$

Note that is $(x,k) \in \mathrm{I}$, we have $(x,j) \in \mathrm{I}$ if and only if $(\mathrm{T}^k(x), j-k) \in \mathrm{I}$ by definition of I (3). Then, we obtain

$$\mathbb{1}_{\mathrm{I}}(\mathrm{T}^k(x)), j-k)\mathbb{1}_{\mathrm{I}}(x,k) = \mathbb{1}_{\mathrm{I}}(x,j)\mathbb{1}_{\mathrm{I}}(x,k)$$

This concludes the proof.

S2. Proofs of Section 3

S2.1. Notations

In this section, we use measure theoretic notations. We denote by $\boldsymbol{\pi}$ and $\boldsymbol{\rho}$ the target and proposal probability measures. These two probability measures are assumed to have p.d.f. w.r.t. the Lebesgue measure on $\mathbb{R}^d$ denoted by π and ρ in the main article. The central property exploited here is that

$$\boldsymbol{\pi}(\mathrm{d}x) = \boldsymbol{\rho}(\mathrm{d}x)\mathrm{L}(x)/Z \ , \tag{S1}$$

or equivalently, using Radon-Nikodym derivative

$$\frac{\mathrm{d}\boldsymbol{\pi}}{\mathrm{d}\boldsymbol{\rho}}(x) = \frac{\mathrm{L}(x)}{Z} \ . \tag{S2}$$

For $k \in \{0,\dots,K\}$, we denote by $\boldsymbol{\rho}_k(\mathrm{d}x)$ the pushforward of $\boldsymbol{\rho}(\mathrm{d}x)\mathbb{1}_I(x,k)$ by T^k, for any nonnegative measurable function f, and $k \in \mathbb{N}$,

$$\int f(x)\boldsymbol{\rho}_k(\mathrm{d}x) = \int f(\mathrm{T}^k(x))\mathbb{1}_I(x,k)\boldsymbol{\rho}(\mathrm{d}x) \ . \tag{S3}$$

If $\boldsymbol{\rho}$ has a density ρ with respect to the Lebesgue measure on $\mathbb{R}^d$, then $\boldsymbol{\rho}_k$ also has a density with respect to the Lebesgue measure which is given by (4). With these notations, for $k \in \{0,\dots,K\}$,

$$w_k(x) = \frac{1}{Z_{\mathrm{T}}}\frac{\mathrm{d}\boldsymbol{\rho}}{\mathrm{d}\boldsymbol{\rho}_T}(\mathrm{T}^k(x)) \ , \tag{S4}$$

$$\boldsymbol{\rho}_T(\mathrm{d}x) = \frac{1}{Z_{\mathrm{T}}}\sum_{k=0}^{K}\boldsymbol{\rho}_k(\mathrm{d}x) \ . \tag{S5}$$

For $i \in \{1,\dots,N\}$, we denote by $R_i(x^i, \mathrm{d}x^{1:N\setminus\{i\}})$ the condition proposal kernels. Recall that for all $i,j \in \{1,\dots,N\}$, we assume that (see (15))

$$\boldsymbol{\rho}(\mathrm{d}x^i)R_i(x^i;\mathrm{d}x^{1:N\setminus\{i\}}) = \boldsymbol{\rho}(\mathrm{d}x^j)R_j(x^j;\mathrm{d}x^{1:N\setminus\{j\}}) = \boldsymbol{\rho}_N(\mathrm{d}x^{1:N}) \ , \tag{S6}$$

where $\boldsymbol{\rho}_N$ is the joint distribution of the proposals. In words, it means that all the one-dimensional marginal of $\boldsymbol{\rho}_N(\mathrm{d}x^{1:N})$ is $\rho(\mathrm{d}x^i)$.

S2.2. Iterated Sampling Importance Resampling

We first consider a general version of the ISIR algorithm (see (Tjelmeland, 2004; Andrieu et al., 2010; Ruiz et al., 2020)) and we show in this section that it is a partially collapsed Gibbs sampler (van Dyk & Park, 2008) of the extended distribution, given for $i \in \{1,\dots,N\}$ by

$$\bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N}, i, \mathrm{d}y) = \frac{1}{N}\boldsymbol{\pi}(\mathrm{d}x^i)R_i(x^i,\mathrm{d}x^{1:N\setminus\{i\}})\delta_{x^i}(\mathrm{d}y) \ . \tag{S7}$$

For ease of presentation, we added the selected sample y in the joint distribution. It is straightforward to establish that the marginal distributions of (S7) are given by

$$\bar{\boldsymbol{\pi}}(\mathrm{d}y) = \boldsymbol{\pi}(\mathrm{d}y) \ , \tag{S8}$$

$$\bar{\boldsymbol{\pi}}(i) = 1/N \ , \quad i \in \{1,\dots,N\} \ , \tag{S9}$$

$$\bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N}) = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\pi}(\mathrm{d}x^i)R_i(x^i;\mathrm{d}x^{1:N\setminus\{i\}}) \ . \tag{S10}$$

We now compute the conditional distributions and check that

$$K_1(i, y; \mathrm{d}x^{1:N}) = \bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N} \mid i, y) = \delta_y(\mathrm{d}x^i) R_i(x^i, \mathrm{d}x^{1:N\setminus\{i\}}) \;. \tag{S11}$$

This corresponds exactly to the first step of ISIR, the refreshment of the set of proposals given the conditioning proposal. Indeed, for any nonnegative measurable functions $\{f_j\}_{j=1}^N$ and g,

$$\begin{aligned}
\frac{1}{N}\sum_{i'=1}^{N}\int \prod_{j=1}^{N} \mathbb{1}_{\{i\}}(i') f_j(x^j) g(y) \bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N}, i', \mathrm{d}y) &= \frac{1}{N}\int \prod_{j=1}^{N} f_j(x^j) g(x^i) \boldsymbol{\pi}(\mathrm{d}x^i) R_i(x^i; \mathrm{d}x^{1:N\setminus\{i\}}) \\
&= \frac{1}{N}\int \boldsymbol{\pi}(\mathrm{d}y) g(y) \int \delta_y(\mathrm{d}x^i) R_i(x^i; \mathrm{d}x^{1:N\setminus\{i\}}) \prod_{j=1}^{N} f_j(x^j) \;,
\end{aligned}$$

which validates (S11). We now establish that the conditional density of i satisfies

$$K_2(x_{1:n}; i) = \bar{\boldsymbol{\pi}}(i \mid x^{1:N}) = \frac{\mathrm{L}(x^i)}{\sum_{j=1}^{N} \mathrm{L}(x^j)} \;. \tag{S12}$$

This corresponds to the second step of the ISIR algorithm, in which a proposal index is selected conditional to the set of proposals. Indeed, for any nonnegative measurable functions $\{f_j\}_{j=1}^N$,

$$\begin{aligned}
\frac{1}{N}\int \boldsymbol{\pi}(\mathrm{d}x^i) R_i(x^i; \mathrm{d}x^{1:N\setminus\{i\}}) \prod_{j=1}^{N} f_j(x^j) &= \frac{1}{NZ}\int \mathrm{L}(x^i) \boldsymbol{\rho}(\mathrm{d}x^i) R_i(x^i; \mathrm{d}x^{1:N\setminus\{i\}}) \prod_{j=1}^{N} f_j(x^j) \\
&= \frac{1}{NZ}\int \mathrm{L}(x^i) \boldsymbol{\rho}_N(\mathrm{d}x^{1:N}) \prod_{j=1}^{N} f_j(x^j) \\
&= \frac{1}{NZ}\int \frac{\mathrm{L}(x^i)}{\sum_{j=1}^{N}\mathrm{L}(x^j)} \sum_{m=1}^{N} \mathrm{L}(x^m) \boldsymbol{\rho}(\mathrm{d}x^m) R_m(x^m; \mathrm{d}x^{1:N\setminus\{m\}}) \prod_{j=1}^{N} f_j(x^j)
\end{aligned}$$

where we have used (S6). We conclude by noting that $\boldsymbol{\pi}(\mathrm{d}x) = \mathrm{L}(x)\boldsymbol{\rho}(\mathrm{d}x)/Z$ and using (S10). We obviously have, by construction, that the conditional distribution of the auxiliary variable y satisfies

$$K_3(x^{1:N}, i; \mathrm{d}y) = \boldsymbol{\pi}(\mathrm{d}y \mid x^{1:N}, i) = \delta_{x^i}(\mathrm{d}y). \tag{S13}$$

This is the final step of the algorithm: the selection of the conditioning particle (this step is implicit in the general description of the algorithm in the main text).

The ISIR sampler is a partially collapsed Gibbs sampler. In the first step (S11), we use the first full conditional, where K_1 leaves $\bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N}, i, \mathrm{d}y)$ invariant. In a second step, we collapse the distribution with respect to y. Lastly, K_2 leaves the marginal $\bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:n}, i)$ invariant. Therefore,

$$\sum_{i_0=1}^{N}\int \bar{\boldsymbol{\pi}}(\mathrm{d}x_0^{1:N}, i_0, \mathrm{d}y_0) K_1(i_0, y_0; \mathrm{d}x_1^{1:N}) K_2(x_1^{1:N}; i_1) = \bar{\boldsymbol{\pi}}(\mathrm{d}x_1^{1:N}, i_1)$$

The validity of the PCG follows from the decomposition

$$\bar{\boldsymbol{\pi}}(\mathrm{d}x_1^{1:N}, i_1) K_3(x_1^{1:N}, i_1; \mathrm{d}y_1) = \bar{\boldsymbol{\pi}}(\mathrm{d}x_1^{1:N}, i_1, \mathrm{d}y_1) \;.$$

S2.3. Invariance for InFiNE sampler

Consider the joint proposal distribution, given for all $i \in \{1, \ldots, N\}$ and $k \in \{0, \ldots, K\}$ by

$$\bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N}, i, k, \mathrm{d}y) = \frac{1}{NZ} w_k(x^i) \mathrm{L}(\mathrm{T}^k(x^i)) \boldsymbol{\rho}(\mathrm{d}x^i) R_i(x^i; \mathrm{d}x^{1:N\setminus\{i\}}) \delta_{\mathrm{T}^k(x^i)}(\mathrm{d}y) \;. \tag{S14}$$

For ease of presentation, we introduce here an additional auxiliary variable, denoted by y, which corresponds to the active sample. We show below that the InFiNE algorithm is a partially collapsed Gibbs sampler; see (van Dyk & Park, 2008).

We first prove that for any $i \in \{1, \dots, N\}$ and $k \in \{0, \dots, K\}$, the marginal distribution of the variables (i, k, y) is given by

$$\bar{\boldsymbol{\pi}}(i, k, \mathrm{d}y) = \frac{1}{N Z_{\mathrm{T}}} \frac{\mathrm{d}\boldsymbol{\pi}}{\mathrm{d}\boldsymbol{\rho}_T}(y) \boldsymbol{\rho}_k(\mathrm{d}y) \ . \tag{S15}$$

Note indeed that, if g is a nonnegative measurable function

$$\begin{aligned}\sum_{i'=1}^{N} \sum_{k'=0}^{K} \int \mathbb{1}_{\{i\}}(i') \mathbb{1}_{\{k\}}(k') g(y) \bar{\boldsymbol{\pi}}(\mathrm{d}x_{1:N}, i', k', \mathrm{d}y) &= \frac{1}{NZ} \int w_k(x^i) \mathrm{L}(\mathrm{T}^k(x^i)) \boldsymbol{\rho}(\mathrm{d}x^i) R_i(x^i; \mathrm{d}x^{1:N \setminus \{i\}}) g(\mathrm{T}^k(x^i)) \\ &= \frac{1}{NZ} \int w_k(x^i) \mathrm{L}(\mathrm{T}^k(x^i)) \boldsymbol{\rho}(\mathrm{d}x^i) g(\mathrm{T}^k(x^i)) \ .\end{aligned}$$

Plugging (S4) inside the integral and using the fact that $\boldsymbol{\rho}_k$ is the pushforward of $\boldsymbol{\rho}$ by T^k, we obtain

$$\begin{aligned}\frac{1}{NZ} \int w_k(x^i) \mathrm{L}(\mathrm{T}^k(x^i)) \boldsymbol{\rho}(\mathrm{d}x^i) g(\mathrm{T}^k(x^i)) &= \frac{1}{NZ} \int \frac{1}{Z_{\mathrm{T}}} \frac{\mathrm{d}\boldsymbol{\rho}}{\mathrm{d}\boldsymbol{\rho}_T}(\mathrm{T}^k(x^i)) \mathrm{L}(\mathrm{T}^k(x^i)) \boldsymbol{\rho}(\mathrm{d}x^i) g(\mathrm{T}^k(x^i)) \\ &= \frac{1}{N Z_{\mathrm{T}}} \int \frac{\mathrm{d}\boldsymbol{\pi}}{\mathrm{d}\boldsymbol{\rho}_T}(\mathrm{T}^k(x^i)) \boldsymbol{\rho}(\mathrm{d}x^i) g(\mathrm{T}^k(x^i)) \\ &= \frac{1}{N Z_{\mathrm{T}}} \int \frac{\mathrm{d}\boldsymbol{\pi}}{\mathrm{d}\boldsymbol{\rho}_T}(y) \boldsymbol{\rho}_k(\mathrm{d}y) g(y) \ ,\end{aligned}$$

which shows (S15). Using (S5),

$$\bar{\boldsymbol{\pi}}(\mathrm{d}y) = \sum_{i=1}^{N} \sum_{k=0}^{K} \bar{\boldsymbol{\pi}}(i, k, \mathrm{d}y) = \sum_{k=0}^{K} \frac{1}{Z_{\mathrm{T}}} \frac{\mathrm{d}\boldsymbol{\pi}}{\mathrm{d}\boldsymbol{\rho}_T}(y) \boldsymbol{\rho}_k(\mathrm{d}y) = \frac{\mathrm{d}\boldsymbol{\pi}}{\mathrm{d}\boldsymbol{\rho}_T}(y) \boldsymbol{\rho}_T(\mathrm{d}y) = \boldsymbol{\pi}(\mathrm{d}y) \ . \tag{S16}$$

Next, we establish that, for $i \in \{1, \dots, N\}$,

$$\bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N}, i) = \frac{\widehat{Z}_{x^i}}{NZ} \boldsymbol{\rho}_N(\mathrm{d}x^{1:N}) \ , \tag{S17}$$

where, see (11),

$$\widehat{Z}_x = \sum_{k=0}^{K} \mathrm{L}(\mathrm{T}^k(x)) w_k(x) \ . \tag{S18}$$

For all nonnegative measurable functions $\{f_j\}_{j=1}^N$,

$$\begin{aligned}\sum_{i'=1}^{N} \sum_{k=0}^{K} \mathbb{1}_{\{i\}}(i') \int \prod_{j=1}^{N} f_j(x^j) \bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N}, i', k, \mathrm{d}y) &= \frac{1}{NZ} \sum_{k=0}^{K} \int w_k(x^i) \mathrm{L}(\mathrm{T}^k(x^i)) \boldsymbol{\rho}_N(\mathrm{d}x^{1:N}) \prod_{j=1}^{N} f_j(x^j) \\ &= \frac{1}{NZ} \int \widehat{Z}_{x^i} \boldsymbol{\rho}_N(\mathrm{d}x^{1:N}) \prod_{j=1}^{N} f_j(x^j) \ ,\end{aligned}$$

which establishes (S17). If we marginalize this distribution w.r.t the path index i, we get

$$\bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N}) = \frac{\widehat{Z}_{x^{1:N}}}{Z} \boldsymbol{\rho}_N(\mathrm{d}x^{1:N}) \ , \tag{S19}$$

where $\widehat{Z}_{x^{1:N}} = \sum_{i=1}^{N} \widehat{Z}_{x^i} / N$, see (12). We then compute the conditional distributions and establish first that for any $i \in \{1, \dots, N\}$ and $k \in \{0, \dots, K\}$,

$$K_1(i, k, y; \mathrm{d}x^{1:N}) = \bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N} \mid i, k, y) = \delta_{\mathrm{T}^{-k}(y)}(\mathrm{d}x^i) R_i(x^i; \mathrm{d}x^{1:N \setminus \{i\}}) \ . \tag{S20}$$

This corresponds to the first step of the InFiNE algorithm. We keep the i-th path and then draw $N-1$ new paths from the conditional kernels $R_i(x^i;\mathrm{d}x^{1:N\setminus\{i\}})$. Because the paths are deterministic, we do not need in practice to compute $\mathrm{T}^{-k}(y)$ (which is the initial point of the path which has been selected). For all nonnegative measurable functions $\{f_j\}_{j=1}^N$ and g,

$$\begin{aligned}
&\frac{1}{NZ}\int\prod_{j=1}^{N} f_j(x^j)g(y)\bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N},i,k,\mathrm{d}y)\\
&=\frac{1}{NZ}\int\prod_{j=1}^{N} f_j(x^j)g(\mathrm{T}^k(x^i))w_k(x^i)\mathrm{L}(\mathrm{T}^k(x^i))\boldsymbol{\rho}(\mathrm{d}x^i)R_i(x^i;\mathrm{d}x^{1:N\setminus\{i\}})\\
&=\frac{1}{NZ}\int\prod_{j=1}^{N} f_j(x^j)g(\mathrm{T}^k(x^i))\frac{1}{Z_\mathrm{T}}\frac{\mathrm{d}\boldsymbol{\rho}}{\mathrm{d}\boldsymbol{\rho}_T}(\mathrm{T}^k(x^i))\mathrm{L}(\mathrm{T}^k(x^i))\boldsymbol{\rho}(\mathrm{d}x^i)R_i(x^i;\mathrm{d}x^{1:N\setminus\{i\}})\\
&=\frac{1}{NZ_\mathrm{T}}\int f_i(x^i)g(\mathrm{T}^k(x^i))\frac{\mathrm{d}\boldsymbol{\pi}}{\mathrm{d}\boldsymbol{\rho}_T}(\mathrm{T}^k(x^i))\boldsymbol{\rho}(\mathrm{d}x^i)\int R_i(x^i;\mathrm{d}x^{1:N\setminus\{i\}})\prod_{j\neq i}f_j(x^j)\;.
\end{aligned}$$

Since $\boldsymbol{\rho}_k$ is the pushforward on $\boldsymbol{\rho}$ by T^k, the latter identity implies

$$\begin{aligned}
&\frac{1}{NZ}\int\prod_{j=1}^{N} f_j(x^j)g(y)\bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N},i,k,\mathrm{d}y)\\
&=\frac{1}{NZ_\mathrm{T}}\int f_i(\mathrm{T}^{-k}(y))g(y)\frac{\mathrm{d}\boldsymbol{\pi}}{\mathrm{d}\boldsymbol{\rho}_T}(y)\boldsymbol{\rho}_k(\mathrm{d}y)\int R_i(\mathrm{T}^{-k}(y);\mathrm{d}x^{1:N\setminus\{i\}})\prod_{j\neq i}f_j(x^j)\\
&=\frac{1}{NZ_\mathrm{T}}\int g(y)\frac{\mathrm{d}\boldsymbol{\pi}}{\mathrm{d}\boldsymbol{\rho}_T}(y)\boldsymbol{\rho}_k(\mathrm{d}y)\int \delta_{\mathrm{T}^{-k}(y)}(\mathrm{d}x^i)R_i(x^i;\mathrm{d}x^{1:N\setminus\{i\}})\prod_{j=1}^{N}f_j(x^j)
\end{aligned}$$

and the proof is concluded by (S15). Next we show that, for $i\in\{1,\ldots,N\}$,

$$K_2(x_{1:N};i)=\bar{\boldsymbol{\pi}}(i\mid x_{1:N})=\frac{\widehat{Z}_{x^i}}{\sum_{j=1}^N\widehat{Z}_{x^j}}\;. \tag{S21}$$

This is the third step of the InFiNE algorithm (the second step in our description amounts to computing the new paths whence the starting points of the trajectories have been updated). For nonnegative measurable functions $\{f_j\}_{j=1}^N$,

$$\begin{aligned}
\frac{1}{NZ}\sum_{k=0}^{K}w_k(x^i)\mathrm{L}(\mathrm{T}^k(x^i))\boldsymbol{\rho}(\mathrm{d}x^i)R_i(x^i;\mathrm{d}x^{1:N\setminus\{i\}})\prod_{\ell=1}^{N}f_\ell(x^\ell)&=\frac{1}{NZ}\int\widehat{Z}_{x^i}\boldsymbol{\rho}_N(\mathrm{d}x^{1:N})\prod_{\ell=1}^{N}f_\ell(x^\ell)\\
&=\frac{1}{NZ}\int\frac{\widehat{Z}_{x^i}}{\sum_{j=1}^N\widehat{Z}_{x^j}}\sum_{j=1}^{N}\widehat{Z}_{x^j}\boldsymbol{\rho}_N(\mathrm{d}x^{1:N})\prod_{\ell=1}^{N}f_\ell(x^\ell)\\
&=\int\frac{\widehat{Z}_{x^i}}{\sum_{j=1}^N\widehat{Z}_{x^j}}\frac{\widehat{Z}_{x^{1:N}}}{Z}\boldsymbol{\rho}_N(\mathrm{d}x^{1:N})\prod_{\ell=1}^{N}f_\ell(x^\ell)\\
&=\int\frac{\widehat{Z}_{x^i}}{\sum_{j=1}^N\widehat{Z}_{x^j}}\bar{\boldsymbol{\pi}}(\mathrm{d}x_{1:N})\prod_{\ell=1}^{N}f_\ell(x^\ell)\;,
\end{aligned}$$

where we used (S19) in the last identity. This establishes (S21). We finally prove that for $k\in\{0,\ldots,K\}$ and $i\in\{1,\ldots,N\}$,

$$K_3(i,x^{1:N};k)=\bar{\boldsymbol{\pi}}(k\mid i,x^{1:N})=\frac{w_k(\mathrm{T}^k(x^i))\mathrm{L}(\mathrm{T}^k(x^i))}{\widehat{Z}_{x^i}}\;. \tag{S22}$$

This is the fourth step of the InFiNE algorithm, which amounts to selecting a proposal along the selected path. Proceeding as above, for nonnegative measurable functions $\{f_j\}_{j=1}^N$,

$$\begin{aligned}
&\frac{1}{NZ}\int w_k(\mathrm{T}^k(x^i))\mathrm{L}(\mathrm{T}^k(x^i))\boldsymbol{\rho}(\mathrm{d}x^i)R_i(x^i;\mathrm{d}x^{1:N\setminus\{i\}})\prod_{j=1}^N f_j(x^j)\\
&=\frac{1}{NZ}\int \frac{w_k(\mathrm{T}^k(x^i))\mathrm{L}(\mathrm{T}^k(x^i))}{\widehat{Z}_{x^i}}\widehat{Z}_{x^i}\boldsymbol{\rho}_N(\mathrm{d}x^{1:N})\prod_{j=1}^N f_j(x^j)\\
&=\int \frac{w_k(\mathrm{T}^k(x^i))\mathrm{L}(\mathrm{T}^k(x^i))}{\widehat{Z}_{x^i}}\bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N},i)\prod_{j=1}^N f_j(x^j)\ ,
\end{aligned}$$

where we used (S17) in the last identity. This establishes (S22). It follows directly from the definition of (S14) that

$$K_4(x_{1:N},i,k;\mathrm{d}y)=\bar{\boldsymbol{\pi}}(\mathrm{d}y\mid x^{1:N},i,k)=\delta_{\mathrm{T}^k(x^i)}(\mathrm{d}y)\ . \tag{S23}$$

This characterizes the sample produced at each iteration of the InFiNE algorithm, which is used to generate the next starting point.

The InFiNE algorithm is a partially collapsed Gibbs. In the first step, (S20), we use the full conditional. In the second step, (S21) (selection of the path index), we marginalize with respect to k and y:

$$\sum_{i_0=1}^N\sum_{k_0=0}^K\int \bar{\boldsymbol{\pi}}(\mathrm{d}x_0^{1:N},i_0,k_0,\mathrm{d}y_0)K_1(i_0,k_0,y_0;\mathrm{d}x_1^{1:N})K_2(x_1^{1:N};i_1)=\bar{\boldsymbol{\pi}}(\mathrm{d}x_1^{1:N},i_1)\ .$$

The transition kernel K_3, defined in (S22) is the full conditional in the decomposition

$$\bar{\boldsymbol{\pi}}(\mathrm{d}x_1^{1:N},i_1)K_3(i_1,x_1^{1:N};k_1)=\bar{\boldsymbol{\pi}}(\mathrm{d}x_1^{1:N},i_1,k_1)\ .$$

The validity of the algorithm is guaranteed by noting that

$$\bar{\boldsymbol{\pi}}(\mathrm{d}x_1^{1:N},i_1,k_1)K_4(x_1^{1:N},i_1,k_1;\mathrm{d}y_1)=\bar{\boldsymbol{\pi}}(\mathrm{d}x_1^{1:N},i_1,k_1,\mathrm{d}y_1)\ .$$

S2.4. Ergodicity of iterated SIR

The ergodicity of iterated SIR has been studied in (Andrieu et al., 2018) in the case when the conditional kernels are independent: $R_i(x^i;\mathrm{d}x^{1:N\setminus\{i\}})=\prod_{j\neq i}\rho(\mathrm{d}x^j)$ under the assumption that the likelihood is bounded $\mathrm{L}_\infty=\sup_{x\in\mathbb{R}^d}\mathrm{L}(x)<\infty$. We extend the analysis to the case of dependent proposals. At iteration k, denote by $X_k^{1:N}$ the set of proposals, I_k the proposal index and the conditioning proposal, $Y_k=X_k^{I_k}$. The algorithm goes as follows:

1. Set $X_{k+1}^{I_k}=Y_{k+1}$ and refresh the set of proposals by drawing $X_{k+1}^{1:N\setminus\{I_k\}}\sim R_{I_k}(X_{k+1}^{I_k},\cdot)$.
2. Compute the unnormalized importance weights $\omega_{k+1}^i=\mathrm{L}(X_{k+1}^i)$, $i\in\{1,\ldots,N\}$.
3. Draw $I_{k+1}\in\{1,\ldots,N\}$ with probabilities proportional to $\{\omega_{k+1}^i\}_{i=1}^N$.
4. Set $Y_{k+1}=X_{k+1}^{I_{k+1}}$.

The key of the analysis is to collapse the representation as to only retain the conditioning index I_k and the conditioning proposal Y_k. It is easily seen that $\{(I_k,Y_k)\}_{k\geq 0}$ is a Markov chain with Markov kernel defined for any $y\in\mathbb{R}^d$ and $A\in\mathcal{B}(\mathbb{R}^d)$ by

$$P(i,y;j\times A)=\int \delta_y(\mathrm{d}x^i)R_i(x^i,\mathrm{d}x^{1:N\setminus\{i\}})\frac{\mathrm{L}(x^j)}{\sum_{\ell=1}^N\mathrm{L}(x^\ell)}\delta_{x^j}(A)\ . \tag{S24}$$

Consider the following assumptions:

H1. *The likelihood function* L *is both lower and upper bounded, i.e.*

$$\kappa = \inf_{x \in \mathbb{R}^d} \mathrm{L}(x) / \sup_{x \in \mathbb{R}^d} \mathrm{L}(x) > 0 \; . \tag{S25}$$

For $i \in \{1, \dots, N\}$ and $j \in \{1, \dots, N\} \setminus \{i\}$, we define for $x^i \in \mathbb{R}^d$ and $A \in \mathcal{B}(\mathbb{R}^d)$,

$$R_{i,j}(x^i, A) = \int R_i(x^i, \mathrm{d}x^{1:N \setminus \{i\}}) \mathbb{1}_A(x^j) \; . \tag{S26}$$

If $R_i(x^i, \mathrm{d}x^{1:N\setminus\{i\}}) = \prod_{\ell \neq i} \rho(\mathrm{d}x^\ell)$, then $R_{i,j}(x^i, A) = \rho(A)$. If the Markov kernel R_i satisfies (17), then $R_{i,j}(x, A) = M^{|j-i|}(x, A)$.

H2. *There exist* $C \in \mathcal{B}(\mathbb{R}^d)$ *and* $\varepsilon > 0$ *such that, for any* $i \neq j \in \{1, \dots, N\}$

1. $\sum_{j=1}^N R_{i,j}(x^i, C) > 0$ *for any* $x^i \in \mathbb{R}^d$.
2. *For any* $x^i \in C$ *and* $A \in \mathcal{B}(\mathbb{R}^d)$, $R_{i,j}(x^i, A) \geq \varepsilon \rho(A)$.

Theorem S1. *Assume **H**1 and **H**2. Then the conditional ISIR kernel* P *(see* (S24)*) is irreducible, positive recurrent and ergodic. If for all* $i \in \{1, \dots, N\}$, $R_i(x^i; \mathrm{d}x^{1:N\setminus\{i\}}) = \prod_{j \neq i} \rho(\mathrm{d}x^j)$*, then* P *is uniformly ergodic.*

Proof. For all $i \in \{1, \dots, N\}$ and $y \in C$ and $A \in \mathcal{B}(\mathbb{R}^d)$ we get

$$P(i, y; j \times A) = \int \delta_y(\mathrm{d}x^i) R_i(x^i; \mathrm{d}x^{1:N\setminus\{i\}}) \frac{\mathrm{L}(x^j)}{\sum_{\ell=1}^N \mathrm{L}(x^\ell)} \delta_{x^j}(A) \geq \frac{\kappa\varepsilon}{N} \rho(A) \; .$$

Hence the set $D = \{1 \dots, N\} \times C$ is small. Under **H**2, we get

$$P(i, y; D) \geq \frac{\kappa}{N} \sum_{j=1}^N R_{i,j}(y, C) > 0 \; ,$$

showing that D is accessible. Since D is accessible and small and $\bar{\boldsymbol{\pi}}(i \times \mathrm{d}y) = \frac{1}{N} \boldsymbol{\pi}(\mathrm{d}y)$ is invariant by P , then P is positive recurrent (see (Douc et al., 2018), Theorem 10.1.6). If the proposals are independent, the whole state space is small and hence the Markov kernel P is uniformly geometrically ergodic. □

The conditions for the InFiNE algorithm are similar.

S3. Additional details about the experiments

S3.1. Additional experiments

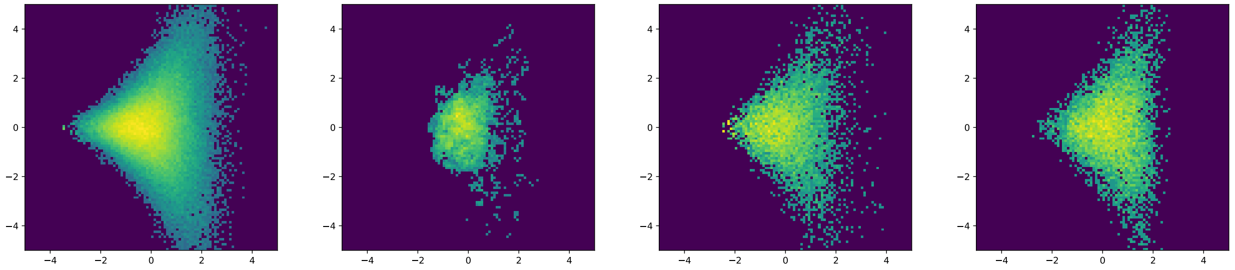
In this section, we consider the target Funnel distribution, following (Jia & Seljak, 2020). The dimension d is set to 16, and the target distribution is

$$\pi(x) = \mathrm{N}(x_1; 0, a^2) \prod_{i=2}^d \mathrm{N}(x_i; 0, \mathrm{e}^{2bx_1}) \; ,$$

with $a = 1$ and $b = 0.5$ and where $x = (x_1, \dots, x_d)$. The normalizing constant of π is thus $Z = 1$ here. InFiNE is used to estimate Z and obtain samples approximately distributed according to π. A reliable choice for the mass matrix and the step-size of InFiNE is obtained by running a warm-up chain of the adaptive HMC or NUTS algorithm given by the Pyro framework which provides estimates of those parameters (Bingham et al., 2019). Therefore, we set the mass matrix and the step size for InFiNE to those provided by the Pyro adaptive scheme. The length K of the trajectories of the InFiNE sampler is set to the number of leapfrog steps of the HMC algorithm, here $K = 10$.

We draw $n = 10^4$ samples and compare them to 10^6 samples from NUTS. We also compare these to $K \cdot 10^4 = 10^5$ samples drawn with ISIR. The prior distribution is chosen as a centered Gaussian with variance $\sigma^2 \mathbf{I}_d$ with $\sigma^2 = 4$. The results

Figure S1. Empirical histograms of samples from the Funnel distribution. From left to right, target distribution (very long run of NUTS), ISIR, HMC and InFiNE



of InFiNE and HMC are similar. Note however that InFiNE lends itself easily to parallel implementations: conformal Hamiltonian integration of the N paths, which is the main computational bottleneck, can be parallelized.

We also present the normalizing constant estimation of this distribution. We initialize the mass matrix and the step-size as discussed previously, and compare IS, AIS, and InFiNE schemes. The IS estimator is run with $2 \cdot 10^5$ samples. For the InFiNE estimator, the number of samples is $N = 2 \cdot 10^4$ and the trajectory length is $K = 10$. The AIS estimator is run with $2 \cdot 10^4$ samples, with the annealing scheme presented in (Grosse et al., 2015, Section 6.2) of length $K = 50$. Moreover, the parameters of the HMC transitions in AIS (mass matrix, step-size) are set to the estimated parameters of the HMC algorithm in Pyro.

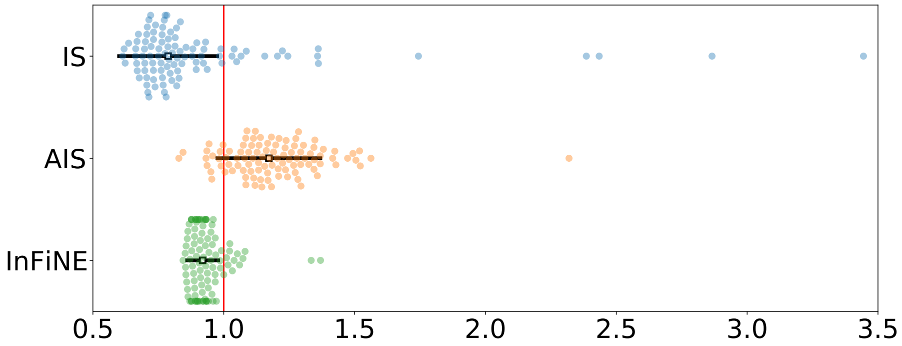


Figure S2. 200 independent estimations of the normalizing constant of π. The prior used is a centered Gaussian distribution with $4\mathbf{I}_d$ as covariance matrix. The true value is $Z = 1$ (red line). The figure displays the median (square) and the interquartile range (solid lines) in each case.

S3.2. VAE experiments

We detail in this section InFiNE VAE with N samples (similarly to the IWAE algorithm). Recall that for each sample, a trajectory of length K is produced. For simplicity, we use $N = 1$ in all our experiments to outline InFiNE VAE in several experimental settings. It is expected that extension to $N > 1$ will further improve the results. Recall that the lower bound $\mathcal{L}_{\mathrm{InFiNE}}$ is

$$\mathcal{L}_{\mathrm{InFiNE}}(\theta, \phi; y) = \int \rho_N(x^{1:N}) \log \widehat{Z}_{x^{1:N}} \mathrm{d}x^{1:N} = \int \prod_{i=1}^{N} q_\phi(x^i \mid y) \log \left(N^{-1} \sum_{i=1}^{N} \sum_{k=0}^{K} w_k(x^i) \frac{p_\theta(y, \mathrm{T}^k(x^i))}{q_\phi(\mathrm{T}^k(x^i) \mid y)} \right) \mathrm{d}x^{1:N} \; .$$

Assume here that q_ϕ is amenable to the reparameterization trick, that is, there exist some diffeomorphism $V_{\phi,y}$ and some fixed pdf g, such that sampling $x \sim q_\phi(\cdot \mid y)$ boils down to sampling $\epsilon \sim$ g and set $x = V_{\phi,y}(\epsilon)$. In the particular case where $N=1$, an estimator of the ELBO and of its gradient are given by

$$\widehat{\mathcal{L}}_{\mathrm{InFiNE}}(\theta,\phi;y)=\log\sum_{k=0}^{K} w_k(x)\frac{p_\theta(y,\mathrm{T}^k(x))}{q_\phi(\mathrm{T}^k(x)\mid y)}\ ,\quad \text{where } x\sim q_\phi(\cdot\mid y)\ ,$$

$$\nabla\widehat{\mathcal{L}}_{\mathrm{InFiNE}}(\theta,\phi;y)=\nabla\log\sum_{k=0}^{K} w_k(V_{\phi,y}(\epsilon))\frac{p_\theta(y,\mathrm{T}^k(V_{\phi,y}(\epsilon)))}{q_\phi(\mathrm{T}^k(V_{\phi,y}(\epsilon))\mid y)}\ ,\quad \text{where } \epsilon\sim \mathrm{g}\ .$$

This is the setting we consider in our experiments. More generally, inspired by the IWAE approach, we can write an estimator of the ELBO and of its gradient as

$$\widehat{\mathcal{L}}_{\mathrm{InFiNE}}(\theta,\phi;y)=\log\left(N^{-1}\sum_{i=1}^{N}\sum_{k=0}^{K} w_k(x^i)\frac{p_\theta(y,\mathrm{T}^k(x^i))}{q_\phi(\mathrm{T}^k(x^i)\mid y)}\right)\ ,\quad \text{where } x^{1:n}\overset{\text{iid}}{\sim} q_\phi(\cdot\mid y)\ ,$$

$$\nabla\widehat{\mathcal{L}}_{\mathrm{InFiNE}}(\theta,\phi;y)=\sum_{i=1}^{N}\varpi_i\nabla\log\left(\sum_{k=0}^{K} w_k(V_{\phi,y}(\epsilon^i))\frac{p_\theta(y,\mathrm{T}^k(V_{\phi,y}(\epsilon^i)))}{q_\phi(\mathrm{T}^k(V_{\phi,y}(\epsilon^i))\mid y)}\right)=\sum_{i=1}^{N}\varpi_i\nabla\log\widehat{Z}_{V_{\phi,y}(\epsilon^i)}\ ,\quad \text{where } \epsilon^{1:n}\overset{\text{iid}}{\sim}\mathrm{g}\ , \tag{S27}$$

where $\varpi_i=\widehat{Z}_{x^i}/(N\widehat{Z}_{x^{1:n}})$.

Algorithm 1 InFiNE VAE, trajectory length K, and N samples

```
**Input:** batch of samples $x$, latent dim $d$.
$(\mu, \log\sigma) \leftarrow EncoderNeuralNet_\phi(x)$.
Sample $N$ initial position and momentums: $q_i \sim \mathcal{N}(\mu,\ \mathrm{diag}(\sigma^2))$ and $p_i \sim \mathcal{N}(0, \mathbf{I}_d)$.
**for** $i = 1$ **to** $N$ **do**
    Compute $\mathrm{T}^k(q_i, p_i)$ *This implies forward / backward passes in the decoder to get* $\nabla \log p_\theta(q_i^k)$.
    Compute $\varpi_i$.
**end for**
Compute the $\mathrm{ELBO}_{\theta,\phi}$ gradient estimator (S27).
SGD update of parameters $(\theta, \phi)$ using the gradient estimatior.
```

Table 1 displays the Negative loglikelihood estimates using both IS and InFiNE on the FashionMNIST dataset (Xiao et al., 2017). The settings are the same than those used in the MNIST experiment. The conclusions are similar: the InFiNE estimate is almost always better than the IS estimate, by a large margin on small dimensions. The InFiNE VAEs are always better than standard VAEs, and better than IWAE with $N=30$ when the dimension of the latent space is small to moderate. When the dimension of the latent space increases ($d=50$), the performance differences become relatively small.

Table 1. NLL estimates for VAE models on FashionMNIST for different latent space dimensions.

	$d=4$		$d=8$		$d=16$		$d=50$	
model	IS	InFiNE	IS	InFiNE	IS	InFiNE	IS	InFiNE
VAE	240.61	240.19	235.78	235.73	235.02	234.96	234.82	234.83
IWAE, $N=5$	239.66	239.27	234.05	233.98	233.12	233.12	233.52	233.46
IWAE, $N=30$	239.25	238.47	233.63	233.49	233.01	232.71	232.88	232.76
InFiNE VAE, $K=3$	238.64	237.91	233.49	233.48	233.26	233.09	233.33	233.35
InFiNE VAE, $K=10$	238.89	238.46	233.51	233.45	233.24	233.15	233.28	233.26

S4. Connection with Nested sampling

We return here to the problem of computing the normalizing constant Z of the target density $\pi(x)=\rho(x)\mathrm{L}(x)/Z$ to point out a simplification induced by our method compared to the method proposed in (Rotskoff & Vanden-Eijnden, 2019). The method proposed in (Rotskoff & Vanden-Eijnden, 2019) uses the identity

$$Z=\int\int_0^\infty \mathbb{1}(\mathrm{L}(x)>\ell)\rho(x)\mathrm{d}\ell\mathrm{d}x=\int_0^\infty \mathbb{P}_{X\sim\rho}(\mathrm{L}(X)>\ell)\mathrm{d}\ell\ , \tag{S28}$$

which was instrumental in the construction of nested sampling (Skilling, 2006; Chopin & Robert, 2010). Using identical level sets as (Skilling, 2006), of the form $\mathsf{O} := \{x : \mathrm{L}(x) > \ell\}$ with $\ell > 0$ and their dissipative Langevin dynamics, (Rotskoff & Vanden-Eijnden, 2019, Equation 13) obtain a concise estimator of the volume of these level sets based on the length of the path $(\mathrm{T}^k(X^i))_{k\in\mathbb{N}}$ remaining inside O. (This estimator is constructed under a uniform prior assumption and continuous-time integrator, but the argument in (Rotskoff & Vanden-Eijnden, 2019) easily translates to discrete-time.)

Considering instead InFiNE, it provides an approximation of $\mathbb{P}_{X\sim\rho}(\mathrm{L}(X) > \ell)$ for a fixed ℓ, but a more efficient resolution is available, which bypasses repeated approximations induced by the quadrature version of both (Skilling, 2006; Rotskoff & Vanden-Eijnden, 2019). The crux of the improvement is that paths only need be simulated once, using only the stopping time associated with the lowest positive ℓ found in early simulations. Integration over the likelihood levels ℓ can then be accomplished with no further approximation. Using a single stopping time as indicated earlier, the following is an unbiased estimator of $\mathbb{P}_{X\sim\rho}(\mathrm{L}(X) > \ell)$ for all values of ℓ:

$$\widehat{\mathbb{P}}_{X\sim\rho}(\mathrm{L}(X) > \ell) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K}\mathbb{1}_{\{\mathrm{L}(\mathrm{T}^k(X^i))>\ell\}}w_k(X^i)\,, \qquad X^i \overset{\text{iid}}{\sim} \rho\,, \tag{S29}$$

where the weights $w_k(X^i)$, defined in (9), incorporate the stopping times. Integrating the above over $\ell \in \mathbb{R}^+$ as in (S28) leads to an estimator of the normalizing constant Z:

$$\begin{aligned}\widehat{Z}_{X^{1:N}} &= \frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K}\int_{\mathbb{R}^+}\mathbb{I}(\mathrm{L}(\mathrm{T}^k(X^i)) > \ell)w_k(X^i)\mathrm{d}\ell \\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K}\mathrm{L}(\mathrm{T}^k(X^i))w_k(x^i)\,,\end{aligned} \tag{S30}$$

where we used the slice sampling identity

$$\int_{\mathbb{R}^+}\mathbb{1}_{\{\mathrm{L}(\mathrm{T}^k(x))>\ell\}}\mathrm{d}\ell = \mathrm{L}(\mathrm{T}^k(x))\,.$$

In conclusion, the InFiNE estimator of Z coincides with the conformal Hamiltonian version of nested sampling with the additional benefit of removing the quadrature approximation. (Note that, as suggested Remark 1, we could resort to both forward and backward push-forward rather than starting at $k = 0$, which could only improve the precision of the estimator (S30).)

Bibliography

Andrieu, C., Doucet, A., and Holenstein, R. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342, 2010.

Andrieu, C., Lee, A., Vihola, M., et al. Uniform ergodicity of the iterated conditional SMC and geometric ergodicity of particle Gibbs samplers. *Bernoulli*, 24(2):842–872, 2018.

Bingham, E., Chen, J. P., Jankowiak, M., Obermeyer, F., Pradhan, N., Karaletsos, T., Singh, R., Szerlip, P., Horsfall, P., and Goodman, N. D. Pyro: Deep universal probabilistic programming. *The Journal of Machine Learning Research*, 20(1): 973–978, 2019.

Chopin, N. and Robert, C. P. Properties of nested sampling. *Biometrika*, 97(3):741–755, 2010.

Douc, R., Moulines, E., Priouret, P., and Soulier, P. *Markov chains*. Springer Series in Operations Research and Financial Engineering. Springer, Cham, 2018. ISBN 978-3-319-97703-4; 978-3-319-97704-1. doi: 10.1007/978-3-319-97704-1. URL `https://doi.org/10.1007/978-3-319-97704-1`.

Grosse, R. B., Ghahramani, Z., and Adams, R. P. Sandwiching the marginal likelihood using bidirectional Monte Carlo. *arXiv preprint arXiv:1511.02543*, 2015.

Jia, H. and Seljak, U. Normalizing constant estimation with Gaussianized bridge sampling. In *Symposium on Advances in Approximate Bayesian Inference*, pp. 1–14. PMLR, 2020.

Rotskoff, G. and Vanden-Eijnden, E. Dynamical computation of the density of states and Bayes factors using nonequilibrium importance sampling. *Physical Review Letters*, 122(15):150602, 2019.

Ruiz, F. J., Titsias, M. K., Cemgil, T., and Doucet, A. Unbiased gradient estimation for variational auto-encoders using coupled Markov chains. *arXiv preprint arXiv:2010.01845*, 2020.

Skilling, J. Nested sampling for general Bayesian computation. *Bayesian Analysis*, 1(4):833–859, 2006.

Tjelmeland, H. Using all Metropolis–Hastings proposals to estimate mean values. Technical report, 2004.

van Dyk, D. A. and Park, T. Partially collapsed Gibbs samplers. *Journal of the American Statistical Association*, 103(482): 790–796, 2008.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.