Invertible Flow Non Equilibrium sampling

Achille Thin, Yazid Janati, Sylvain Le Corff, Charles Ollion, Arnaud Doucet, Alain Durmus, Eric Moulines, Christian Robert

Invertible Flow Non Equilibrium sampling

Achille Thin[†], Yazid Janati[‡], Sylvain Le Corff[‡], Charles Ollion[†], Arnaud Doucet[⊤], Alain Durmus[⋆], Éric Moulines[†], and Christian Robert[≀]

[†]CMAP, École Polytechnique, Institut Polytechnique de Paris, Palaiseau.

[‡]Samovar, Télécom SudParis, département CITI, TIPIC, Institut Polytechnique de Paris, Palaiseau.

[⊤]Department of Statistics, University of Oxford.

[⋆]CMLA, École Normale Supérieure Paris-Saclay.

[≀]Ceremade, Université Paris-Dauphine & Department of Statistics, University of Warwick.

Abstract

Simultaneously sampling from a complex distribution with intractable normalizing constant and approximating expectations under this distribution is a notoriously challenging problem. We introduce a novel scheme, Invertible Flow Non Equilibrium Sampling (InFiNE), which departs from classical Sequential Monte Carlo (SMC) and Markov chain Monte Carlo (MCMC) approaches. InFiNE constructs unbiased estimators of expectations and in particular of normalizing constants by combining the orbits of a deterministic transform started from random initializations. When this transform is chosen as an appropriate integrator of a conformal Hamiltonian system, these orbits are optimization paths. InFiNE is also naturally suited to design new MCMC sampling schemes by selecting samples on the optimization paths. Additionally, InFiNE can be used to construct an Evidence Lower Bound (ELBO) leading to a new class of Variational AutoEncoders (VAE).

1 Introduction

Simulation from a challenging distribution $\pi(x) \propto \rho(x)\mathrm{L}(x)$ and approximation of its intractable normalizing constant $Z = \int \rho(x)\mathrm{L}(x)\mathrm{d}x$ remains a significant issue for generative models and Bayesian inference. In a Bayesian setting, ρ is a prior distribution and L is the likelihood. In Generative Adversarial Networks (GAN) Turner et al. (2019); Che et al. (2020), ρ is the generator and L is derived from the discriminator. This problem has attracted wealth of contributions; see for example Chen et al. (2000). Simulation approaches rarely rely on output from the target, since it either produces unreliable substitutes, as in the discredited harmonic mean estimator of Newton and Raftery (1994) or difficulties of implementation as in path sampling Gelman and Meng (1998) and nested sampling Skilling (2006); Chopin and Robert (2010). Many approaches are based on Importance Sampling (IS) techniques, the most popular being Annealed Importance Sampling (AIS) Neal (2001); Wu et al. (2016); Ding and Freedman (2019) and Sequential Monte Carlo (SMC) Del Moral et al. (2006). Many contributions about the estimation of normalizing constants have been devoted to use as an importance distribution the push-forward $\mathrm{T}_{\#}\,\rho$ of a base probability ρ by an invertible map T; see among others Jarzynski (2002); Meng and Schilling (2002); Neal (2005); Cuendet (2006); Procacci et al. (2006). More recently, it has been proposed to select the parameters of such a map so as to minimize the 'mode seeking' Kullback–Leibler (KL) divergence between $\mathrm{T}_{\#}\,\rho$ and π; see

e.g. El Moselhy and Marzouk (2012); Müller et al. (2019); Papamakarios et al. (2019); Prangle (2019); Wirnsberger et al. (2020). In high-dimension, these approaches can provide an importance distribution $\mathrm{T}_{\#}\,\rho$ which "covers" only a part of the support of ρ and therefore lead to ill-behaved importance weights. Finally, other proposals have focused solely on the normalizing constant approximation, as in Chib (1995) or the antagonistic solutions of Geyer (1993); Gutmann and Hyvärinen (2012). When these estimates are unbiased, they can be used to obtain ELBO to design Variational Auto-Encoders (VAE) Mnih and Rezende (2017).

Rotskoff and Vanden-Eijnden (2019) have introduced a new *Non-Equilibrium IS* (NEIS) method. It is inspired by Hamiltonian Monte Carlo (HMC) techniques in the sense that proposals are sampled from an Hamiltonian flow. However, contrary to "classical" HMC, a friction term is added, hence does not leave the Hamiltonian invariant. The NEIS estimator of the normalizing constant cannot be computed exactly as the theory relies on the integration of the conformal Hamiltonian flow. In practical implementations, a discretization is required and induces approximation errors.[1]

We propose in this work a new (discrete-time) Invertible Flow Non Equilibrium IS estimator for Z, named InFiNE, that circumvents the issues of the original estimator of Rotskoff and Vanden-Eijnden (2019). InFiNE method relies on iterated calls to a map $\mathrm{T} : \mathbb{R}^d \to \mathbb{R}^d$. When T is a discrete-time approximation of a conformal Hamiltonian integrator Franca et al. (2019), InFiNE constructs an estimate of the normalizing constant with *optimization paths* from random starting points. Moreover, contrary to NEIS, the InFiNE estimator is unbiased under assumptions that are mild and easy to verify. Finally, InFiNE lends itself well to massive parallelization. As illustrated in our numerical experiments, InFiNE improves the efficiency of state-of-the-art methods in a various set of experiments. In Section 4, we present different domains of applications for InFiNE that demonstrate its generality and the reach of its efficiency.

Our contributions can be summarized as follows:

- **(i)** We introduce a novel IS estimator, InFiNE, which builds and relies on optimization paths to estimate efficiently normalizing constants. In our numerical experiments, InFiNE is shown to be competitive with state-of-the-art methods.
- **(ii)** We show how InFiNE can be used to develop a novel class of Variational Auto-Encoders (VAE).
- **(iii)** We present new MCMC samplers that build upon InFiNE. This leads to massively parallel sampling methods obtained by selecting points on optimization paths started at random positions.

2 Invertible Flow Non Equilibrium Importance Sampling

In the spirit of the above, we thus consider a pdf ρ on $\mathbb{R}^d$, along with a C^1-diffeomorphism $\mathrm{T} : \mathbb{R}^d \to \mathbb{R}^d$. Write, for $k \in \mathbb{N}^*$, $\mathrm{T}^k = \mathrm{T} \circ \mathrm{T}^{k-1}$, $\mathrm{T}^0 = \mathrm{Id}_d$ and similarly $\mathrm{T}^{-k} = \mathrm{T}^{-1} \circ \mathrm{T}^{-(k-1)}$. Assume T is measure-preserving for ρ, meaning that when X has distribution ρ, for all $k \in \mathbb{Z}$, $\mathrm{T}^k(X)$ has also distribution ρ. Then, for an arbitrary nonnegative sequence $(\varpi_k)_{k\in\mathbb{Z}}$ such that $\sum_{k\in\mathbb{Z}} \varpi_k = 1$,

$$N^{-1} \sum_{k\in\mathbb{Z}} \varpi_k \sum_{i=1}^{N} f(\mathrm{T}^k(X^i)) \;, \qquad (X^i)_{1\le i\le N} \overset{\mathrm{iid}}{\sim} \rho$$

is an unbiased estimate of $\int f(x)\rho(x)\mathrm{d}x$. It further enjoys a smaller variance than the Monte Carlo estimator $N^{-1}\sum_{i=1}^{N} f(X^i)$.

[1] As done in the code provided by Rotskoff and Vanden-Eijnden (2019), while the impact of the discretization on the bias is not addressed in the paper.

InFiNE generalizes this construction to an arbitrary invertible flow T, tailored to move the samples $X^{1:N}$ towards regions with important contribution to the computation of $\int f(x)\rho(x)\mathrm{d}x$. All proofs associated with this section are postponed to Appendix A of the supplementary material.

2.1 Integration using non-equilibrium paths

Let O be the support of $f\rho$. In the applications below, our transformation T is defined on O. Thus, in the case where $\mathsf{O} \neq \mathbb{R}^d$, this motivates the introduction of an estimator based on sequences supported in O. Although we focus on applications where $\mathsf{O} = \mathbb{R}^d$ below, important extensions of our work discussed at the end of this section require $\mathsf{O} \neq \mathbb{R}^d$. Define the following exit times $\tau^+ : \mathbb{R}^d \to \mathbb{N}$ and $\tau^- : \mathbb{R}^d \to \mathbb{N}_-$, given, for all $x \in \mathbb{R}^d$, by

$$\tau^+(x) = \inf\{k \geq 1 \,:\, \mathrm{T}^k(x) \notin \mathsf{O}\} \,, \tag{1}$$

$$\tau^-(x) = \sup\{k \leq -1 \,:\, \mathrm{T}^k(x) \notin \mathsf{O}\} \,, \tag{2}$$

with the convention $\inf \emptyset = +\infty$ and $\sup \emptyset = -\infty$, and

$$\mathrm{I} = \{(x,k) \in \mathsf{O} \times \mathbb{Z} \,:\, k \in [\tau^-(x)+1 : \tau^+(x)-1]\} \,. \tag{3}$$

For any $k \in \mathbb{Z}$, define $\rho_k : \mathbb{R}^d \to \mathbb{R}_+$ by

$$\rho_k(x) = \rho(\mathrm{T}^{-k}(x))\mathbf{J}_{\mathrm{T}^{-k}}(x)\mathbb{1}_{\mathrm{I}}(x,-k) \,, \tag{4}$$

where $\mathbf{J}_\Phi(x) \in \mathbb{R}^+$ denotes the Jacobian of $\Phi : \mathbb{R}^d \to \mathbb{R}^d$ evaluated at x. The density ρ_k is the push-forward measure of $\mathbb{1}_{\mathrm{I}}(x,k)\rho(x)$ by T^k, *i.e.* for any $k \in \mathbb{Z}$ and $f : \mathbb{R}^d \to \mathbb{R}$,

$$\int f(y)\rho_k(y)\mathrm{d}y = \int f(\mathrm{T}^k(x))\mathbb{1}_{\mathrm{I}}(x,k)\rho(x)\mathrm{d}x \,. \tag{5}$$

When $(x,k) \in \mathrm{I}$ for any $x \in \mathsf{O}$ and any $1 \leq k \leq K$, a crucial identity is

$$\begin{aligned}\int f(y)\rho(y)\mathrm{d}y &= \int f(\mathrm{T}^k(x))\rho(\mathrm{T}^k(x))|\mathbf{J}_{\mathrm{T}^k}(x)|\mathrm{d}x \\ &= \int f(\mathrm{T}^k(x))\frac{\rho(\mathrm{T}^k(x))}{\rho_k(\mathrm{T}^k(x))}\rho(x)\mathrm{d}x \,.\end{aligned}$$

If $X^{1:N} \overset{\text{iid}}{\sim} \rho$, this suggests to improve the basic Monte Carlo estimator by the still unbiased estimator

$$\frac{1}{(K+1)N}\sum_{i=1}^{N}\sum_{k=0}^{K} f(\mathrm{T}^k(X^i))\frac{\rho(\mathrm{T}^k(X^i))}{\rho_k(\mathrm{T}^k(X^i))} \,, \tag{6}$$

obtained by averaging over flows T^k, towards turning the dominating measure into a T invariant one as in Kong et al. (2003).

InFiNE estimators exploit the above identity by computing the average of the $K+1$ measures ρ_k, $0 \leq k \leq K$, in the general case when $(x,k) \notin I$ for some values of (x,k). More precisely, in line with multiple importance sampling *à la* Owen and Zhou (2000), we introduce the pdf

$$\rho_{\mathrm{T}}(x) = Z_{\mathrm{T}}^{-1}\sum\nolimits_{k=0}^{K}\rho_k(x) \,, \tag{7}$$

where Z_{T} is the normalizing constant. This is a *non-equilibrium* distribution, since ρ_{T} is not invariant by T in general. Using ρ_{T} as an importance distribution to obtain an unbiased estimator of $\int f(x)\rho(x)\mathrm{d}x$ is feasible since it shares the same support as ρ, hence

$$\int f(x)\rho(x)\mathrm{d}x = \int \left(f(x)\frac{\rho(x)}{\rho_{\mathrm{T}}(x)} \right) \rho_{\mathrm{T}}(x)\mathrm{d}x \ .$$

From (5), the right hand side can be computed using the following key result whose proof is postponed to the supplementary material.

Theorem 1. *For any $f : \mathbb{R}^d \to \mathbb{R}$, we have*

$$\int f(x)\rho(x)\mathrm{d}x = \int \sum_{k=0}^{K} f(\mathrm{T}^k(x)) w_k(x)\rho(x)\mathrm{d}x \ , \tag{8}$$

where, with the convention $0/0 = 0$,

$$w_k(x) = \rho(\mathrm{T}^k(x))\mathbb{1}_{\mathrm{I}}(x,k)/[Z_{\mathrm{T}}\rho_{\mathrm{T}}(\mathrm{T}^k(x))] \ . \tag{9}$$

Note that $Z_{\mathrm{T}}\rho_{\mathrm{T}}(\mathrm{T}^k(x))$ simplifies and the normalizing constant Z_{T} does not appear in the right-hand side of (9). A naive implementation would require $O(K^2)$ complexity per sample, however a linear $O(K)$ estimator can be derived thanks to the following result.

Lemma 2. *For any $x \in \mathbb{R}^d$ and $k \in \{0, \ldots, K\}$,*

$$w_k(x) = \rho_{-k}(x) \Big/ \sum_{j=-k}^{K-k} \rho_j(x) \quad . \tag{10}$$

By Lemma 2, the weights w_k are also upper bounded uniformly in x: for any $x \in \mathbb{R}^d$, $w_k(x) \leq 1$. From (25) and Lemma 2, the InFiNE estimator of $\int f(x)\rho(x)\mathrm{d}x$ is defined in Algorithm 1. Contrary to

Algorithm 1 InFiNE method

(1) Sample $X^i \overset{\text{iid}}{\sim} \rho$ for $i \in [N]$.

(2) For $i \in [N]$, compute the path $(\mathrm{T}^j(X^i))_{j=0}^K$ and weights $(w_j(X^i))_{j=0}^K$.

(3) $I_N^{\mathrm{InFiNE}}(f) = \frac{1}{N}\sum_{i=1}^N \sum_{k=0}^K w_k(X^i) f(\mathrm{T}^k(X^i))$.

self normalized IS versions, we stress that $I_N^{\mathrm{InFiNE}}(f)$ remains unbiased despite the ratio appearing in the expression (10) of the weights.

Theorem 3. *$I_N^{\mathrm{InFiNE}}(f)$ is an unbiased estimator of $\int f(x)\rho(x)\mathrm{d}x$.*

Remark 1. We have chosen here to focus on multiple importance sampling to forward in time pushforwards $\{\rho_k\}_{k=0}^K$. The same construction holds if we consider both backward and forward pushforwards $\{\rho_k\}_{k=-K}^K$. If we take formally $K = \infty$ in (7), then ρ_{T} becomes invariant with respect to T. In this case, this becomes the discrete-time counterpart of the algorithm proposed in Rotskoff and Vanden-Eijnden (2019). In this particular case, we can write for $k \in \mathbb{Z}, x \in \mathbb{R}^d$,

$$w_k(x) = \rho_{-k}(x) \Big/ \sum_{j=-\infty}^{+\infty} \rho_j(x) \quad ,$$

in which case the weights are exactly self-normalized, and

$$I_N^{\mathrm{InFiNE}}(f) = N^{-1} \sum_{i=1}^{N} \sum_{k=-\infty}^{+\infty} w_k(X^i) f(\mathrm{T}^k(X^i)) \; .$$

However, choosing $K = +\infty$ requires additional assumptions on the stopping times and the measures ρ_k.

Remark 2. We can extend InFiNE to non homogeneous flows, replacing the family $\{\mathrm{T}^k \colon k \in \mathbb{Z}\}$ with a collection of mappings $\{\mathsf{T}_k \colon k \in \mathbb{Z}\}$. This would allow us to consider further flexible classes of transformations such as normalizing flows Papamakarios et al. (2019). However, we focus in the following on a single operator that targets the optima of $f\rho$, and leave this extension to future work.

Remark 3. In the case where ρ is an uniform distribution on the set O, $I_N^{\mathrm{InFiNE}}(f)$ offers some similarity with the Nested Sampling estimator Skilling (2006). In particular, it can then be rewritten with stopping times on each of the energy level sets on O, building on the stopping times introduced at the beginning of this section and the Nested Sampling identity; see (Chopin and Robert, 2010, Section 2). We develop this remark in the supplementary material.

2.2 Conformal Hamiltonian transformation

We now return to the challenging target density $\pi(x) = \mathrm{L}(x)\rho(x)/Z$, where the normalizing constant Z is intractable. By applying Algorithm 1 to the test function L, an unbiased estimator of Z is derived as

$$\widehat{Z}_{X^i} = \sum\nolimits_{k=0}^{K} \mathrm{L}(\mathrm{T}^k(X^i)) w_k(X^i) \tag{11}$$

$$\widehat{Z}_{X^{1:N}} = \sum\nolimits_{i=1}^{N} \hat{Z}_{X^i}/N \; . \tag{12}$$

The efficiency of such an estimator relies heavily on the choice of T. Intuitively, a sensible choice of T requires that (i) T is able to drive samples to regions which contributes strongly to the computation of Z (aka regions where the likelihood L is high) and (ii) the Jacobian of T is cheap to compute. These constraints naturally lead to use a conformal Hamiltonian dynamics, as suggested in Rotskoff and Vanden-Eijnden (2019). Assume that $U(\cdot) = \log \pi(\cdot)$ is continuously differentiable. We consider an extended distribution $\tilde{\pi}(q,p) \propto \exp\{-U(q) - K(p)\}$ on $\mathbb{R}^{2d}$, where $K : p \mapsto p^T \mathrm{M}^{-1} p/2$, with M a positive definite mass matrix. Note that π is the marginal of $\tilde{\pi}$. In this setting, $q \in \mathbb{R}^d$ is the position and $U(q)$ is the *potential energy*, while $p \in \mathbb{R}^d$ is the momentum and $K(p)$ is the *kinetic energy*, by analogy with physics. The conformal Hamiltonian ODE associated with $\tilde{\pi}$ is defined by

$$\begin{aligned} \mathrm{d}q_t/\mathrm{d}t &= \nabla_p H(q_t, p_t) = \mathrm{M}^{-1} p_t \; , \\ \mathrm{d}p_t/\mathrm{d}t &= -\nabla_q H(q_t, p_t) - \gamma p_t = -\nabla U(q_t) - \gamma p_t \; , \end{aligned} \tag{13}$$

where $H(q,p) = U(q) + K(p)$, and $\gamma > 0$ is a damping constant. Any solution $(q_t, p_t)_{t \geq 0}$ of (13) satisfies $\mathrm{d}H/\mathrm{d}t(q_t, p_t) = -\gamma p_t^T \mathrm{M}^{-1} p_t \leq 0$. Hence, all orbits converge to fixed points that satisfy $\nabla U(q) = 0$ and $p = 0$; see e.g. Franca et al. (2019); Maddison et al. (2018).

In the applications below, we consider the conformal version of the symplectic Euler method of (13), see Franca et al. (2019). This integrator can be constructed as a splitting of the two conformal and conservative parts of the system (13). When composing a dissipative with a symplectic operator, we set for all $(q,p) \in \mathbb{R}^{2dn}$, $\mathrm{T}_h(q,p)$ to be

$$(q + h\mathrm{M}^{-1}\{\mathrm{e}^{-h\gamma} p - h\nabla U(q)\}, \mathrm{e}^{-h\gamma} p - h\nabla U(q)) \; ,$$

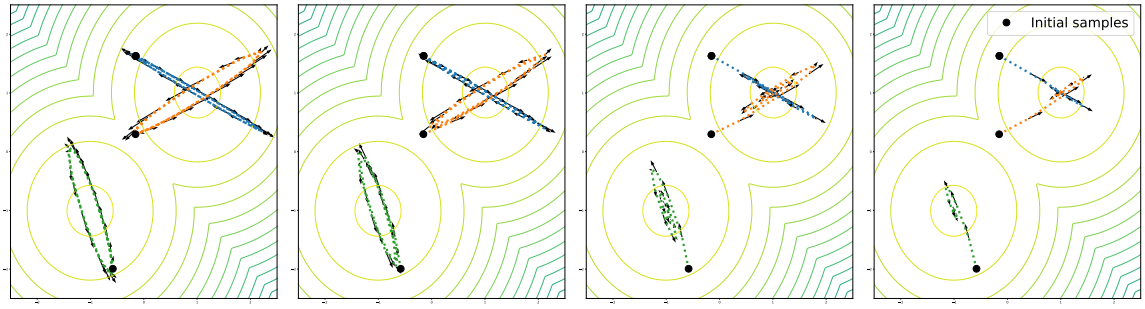


Figure 1: Conformal Hamiltonian paths for different values of the dissipation parameters for a mixture of two Gaussian distributions given different Hamiltonian parameters. From left to right, γ increasing from 0 to 0.3, 2 and 4.

where $h > 0$ is a discretization stepsize. This transformation can be connected with classical momentum optimization schemes, see (Franca et al., 2019, Section 4). By (Franca et al., 2019, Section 3), for any $h > 0$ T_h is a C^1-diffeomorphism on $\mathbb{R}^{2d}$ with Jacobian given by $\mathbf{J}_{\mathrm{T}_h}(q,p) = \mathrm{e}^{-\gamma h d}$. In addition, its inverse is $\mathrm{T}_h^{-1}(q,p) = (q - h\mathrm{M}^{-1}p, \mathrm{e}^{\gamma h}\{p + h\nabla U(q - h\mathrm{M}^{-1}p)\})$. Therefore, the weight (10) of the InFiNE estimator is given by

$$w_k(q,p) = \frac{\tilde{\rho}(\mathrm{T}_h^k(q,p))\mathrm{e}^{-\gamma k h d}}{\sum_{j=k-K}^{k} \tilde{\rho}(\mathrm{T}_h^j(q,p))\mathrm{e}^{-\gamma j h d}}, \tag{14}$$

where $\tilde{\rho}(q,p) \propto \rho(q)\mathrm{e}^{-K(p)}$. In the applications below, M is chosen as a diagonal matrix with positive entries, see the discussion in Section 5.1.

3 InFiNE-based MCMC

We describe here novel MCMC algorithms that leverage the InFiNE method to sample from π.

To motivate our sampler, let us recall the principle of the Sampling Importance Resampling method (SIR; Rubin (1987); Smith and Gelfand (1992)) whose goal is to approximately sample from the target distribution π using samples drawn from a proposal distribution ρ.

In SIR, a N-i.i.d. sample $X^{1:N}$ is first generated from the proposal distribution ρ. A sample X^* is approximately drawn from the target π by choosing randomly a value in $X^{1:N}$ with probabilities proportional to the importance weights $\{\tilde{w}(X^i)\}_{i=1}^N$, where $\tilde{w}(x) = \pi(x)/\rho(x)$. Note that the importance weights are required to be known only up to a constant factor. For SIR, as $N \to \infty$, the sample X^* is *asymptotically* distributed according to π; see Smith and Gelfand (1992). Two major drawbacks of SIR are that it is only asymptotically valid and that the number N of proposals should typically grow exponentially with the dimension d of the state-space to maintain a given accuracy.

A subsequent algorithm is the *iterated SIR* (ISIR) Andrieu et al. (2010). In this version, the sample size N is not necessarily large ($N \geq 2$), but the whole process of sampling a set of proposals, computing the importance weights, and picking a candidate, is iterated. At the n-th step of ISIR, the active set of N proposals $X_n^{1:N}$ and the index $I_n \in [N]$ of the conditioning proposal are kept. First ISIR updates the active set by setting $X_{n+1}^{I_n} = X_n^{I_n}$ (keep the conditioning proposal) and then draw independently $X_{n+1}^{1:N\setminus\{I_n\}}$ from ρ. Then it selects the next proposal index $I_{n+1} \in [N]$ by sampling with probability proportional

to $\{\tilde{w}(X_{n+1}^i)\}_{i=1}^N$. As shown in Andrieu et al. (2010), this algorithm defines a partially collapsed Gibbs sampler (PCG) of the augmented distribution (see Appendix B.2)

$$\bar{\boldsymbol{\pi}}(x^{1:N}, i) = \frac{1}{N}\pi(x^i)\prod_{j\neq i}\rho(x^j) = \frac{1}{N}\tilde{w}(x^i)\prod_{j=1}^N \rho(x^j) \ .$$

The PCG sampler can be shown to be ergodic provided that ρ and π are continuous and ρ is positive on the support of π. If in addition the importance weights are bounded, the Gibbs sampler can be shown to be uniformly geometrically ergodic Lindsten et al. (2015); Andrieu et al. (2018). It follows that the distribution of the conditioning proposal $X_n^* = X_n^{I_n}$ converges to π as the iteration index n goes to infinity. Indeed, for any integrable function f on $\mathbb{R}^d$, with $(X_{1:N}, I) \sim \bar{\boldsymbol{\pi}}$,

$$\mathbb{E}[f(X^I)] = \int \sum_{i=1}^N f(x^i)\bar{\boldsymbol{\pi}}(x^{1:N}, i)\mathrm{d}x^{1:N} = N^{-1}\sum_{i=1}^N \int f(x^i)\pi(x^i)\mathrm{d}x_i = \int f(x)\pi(x)\mathrm{d}x \ .$$

When the state space dimension d increases, designing a proposal distribution ρ guaranteeing proper mixing properties becomes more and more difficult. A way to circumvent this problem is to use dependent proposals, allowing in particular *local moves* around the conditioning path. To implement this idea, for each $i \in [N]$, we define a proposal transition, $r_i(x^i; x^{1:N\setminus\{i\}})$ which defines the the conditional distribution of $X^{1:N\setminus\{i\}}$ given $X^i = x^i$. The key property validating ISIR with dependent proposals (see Appendix B.2) is that all one-dimensional marginal distributions are equal to ρ, which requires that for each $i, j \in [N]$,

$$\rho(x^i)r_i(x^i; x^{1:N\setminus\{i\}}) = \rho(x^j)r_j(x^j; x^{1:N\setminus\{j\}}) \tag{15}$$

The (unconditional) joint distribution of the particles is therefore defined as

$$\rho_N\left(x^{1:N}\right) = \rho(x^1)r_1(x^1; x^{1:N\setminus\{1\}}) \ . \tag{16}$$

The resulting modification of the ISIR algorithm is straightforward: $X^{1:N\setminus\{I_n\}}$ is sampled jointly from the conditional distribution $r_{I_n}(X_n^{I_n}, \cdot)$ rather than independently from ρ.

There are many ways to make proposals dependent. For instance, dependence may be induced by using a Markov kernel reversible with respect to to the proposal ρ, i.e., such that $\rho(x)m(x, x') = \rho(x')m(x', x)$, assuming for simplicity that this kernel has density $m(x, x')$ Ruiz et al. (2020). In this case, for each $i \in [N]$, the conditional proposal kernel is

$$r_i(x^i, x^{1:N\setminus\{i\}}) = \prod_{j=1}^{i-1} m(x^{j+1}, x^j) \prod_{j=i+1}^{n} m(x^{j-1}, x^j) \ . \tag{17}$$

A straightforward induction shows that (15) is satisfied and that the joint distribution of the particles (see (38) is given by $\rho_N(x^{1:N}) = \rho(x^i)\prod_{j=2}^N m(x^{j-1}, x^j)$. If ρ is Gaussian, an appropriate choice is an autoregressive kernel $m(x, x') = \phi_d(x'; \alpha x, \sqrt{1-\alpha^2}\,\mathrm{Id}_d)$, where $\phi_d(x; \mu, \Sigma)$ is the d-dimensional Gaussian pdf with mean μ and covariance Σ as in Ruiz et al. (2020). More generally, we can use a Metropolis-Hastings kernel with invariant distribution ρ.

We now propose the InFiNE MCMC sampler which extends the ISIR algorithm to InFiNE construction. The input for the n-th iteration comprises an active set of N path initial states, $X^{1:N}$, the index $1 \leq I_n \leq N$ of the conditioning path, and the iteration index $0 \leq K_n \leq K$ along the conditioning path. Adopting the ISIR protocol, our sampler proceeds as follows.

1. Set $X_{n+1}^{I_n} = X_n^{I_n}$ and draw the remaining proposals $X_{n+1}^{1:N\setminus\{I_n\}} \sim r_{I_n}(X^{I_n}, \cdot)$.
2. For each initial value X_{n+1}^i, $i \in [N]$, compute the iterates $\{\mathrm{T}^k(X_{n+1}^i)\}_{k=1}^K$.
3. Draw the path index $I_{n+1} \in [N]$ with probability proportional to $(\widehat{Z}_{X_{n+1}^i})_{i\in[N]}$, with $\widehat{Z}_{X_{n+1}^i}$ defined in (11).
4. Draw the next iteration index $0 \leq K_{n+1} \leq K$ on the conditioning path with probability proportional to
$$w_k(X_{n+1}^{I_{n+1}})\mathrm{L}(\mathrm{T}^k(X_{n+1}^{I_{n+1}}))\ .$$

Similar to ISIR, InFiNE MCMC is a partially collapsed Gibbs sampler targeting the extended pdf (see Appendix B.3)

$$\begin{aligned}
&\bar{\boldsymbol{\pi}}(x^{1:N}, i, k)\\
&= w_k(x^i)\mathrm{L}(\mathrm{T}^k(x^i)\rho(x^i)r_i(x^i; x^{1:N\setminus\{i\}})\big/NZ\\
&= w_k(x^i)\mathrm{L}(\mathrm{T}^k(x^i))\rho_N(x^{1:N})\big/NZ\ .
\end{aligned} \tag{18}$$

whose marginal distribution satisfies

$$\bar{\boldsymbol{\pi}}(x^{1:N}, i) = \frac{1}{NZ}\widehat{Z}_{x^i}\rho(x^i)r_i(x^i; x^{1:N\setminus\{i\}})\ .$$

Under mild conditions (see Appendix B.4), this PCG sampler is ergodic, hence the distributions of the iterates $(X_n^{1:N}, I_n, K_n)$ and of their projections $X_n^* = \mathrm{T}^{K_n}(X_n^{I_n})$ converge to $\bar{\boldsymbol{\pi}}$ and to π, respectively. Indeed, for any integrable function f on $\mathbb{R}^d$, with $(X_{1:N}, I, K) \sim \bar{\boldsymbol{\pi}}$,

$$\begin{aligned}
\mathbb{E}[f(T^K(X^I))] &= \sum_{i=1}^N \int \sum_{k=0}^K \bar{\boldsymbol{\pi}}(x^{1:N}, i, k) f(T^k(x^i))\mathrm{d}x^{1:N}\\
&= (NZ)^{-1}\sum_{i=1}^N \int \sum_{k=0}^K \rho(x^i)w_k(x^i)\mathrm{L}(\mathrm{T}^k(x^i))f(\mathrm{T}^k(x^i))\mathrm{d}x^i\\
&= (NZ)^{-1}\sum_{i=1}^N \int \rho(x^i)\mathrm{L}(x^i)f(x^i)\mathrm{d}x^i = \int \boldsymbol{\pi}(y)f(y)\mathrm{d}y\ ,
\end{aligned}$$

following Theorem 1. The InFiNE MCMC sampler is thus a valid procedure to generate samples from $\boldsymbol{\pi}$. When the transformation T is chosen as in Section 2.2, our sampler draws samples based on optimization paths. Detailed experiments are discussed in Section 5.2.

4 ELBO for variational auto-encoders

Given a joint model $p_\theta(y, x)$, with data $y \in \mathbb{R}^p$ and latent variable $x \in \mathbb{R}^d$, variational inference (VI) provides us with a tool to both approximate the intractable posterior $p_\theta(x|y)$ and maximize the marginal likelihood $p_\theta(y) = \int p_\theta(x, y)\mathrm{d}x$ in the parameter θ. This is achieved by introducing a parameterized approximate posterior $q_\phi(x|y)$ and maximizing the Evidence Lower Bound (ELBO) (see Kingma and Welling

(2019))

$$\begin{aligned}\mathcal{L}_{\mathrm{ELBO}}(\theta,\phi) &= \int \log\left(\frac{p_\theta(x,y)}{q_\phi(x \mid y)}\right) q_\phi(x \mid y)\mathrm{d}x \\ &= \log p_\theta(y) - \mathrm{KL}(q_\phi(\cdot \mid y) \| p_\theta(\cdot \mid y)) \ ,\end{aligned} \tag{19}$$

where KL is the Kullback–Leibler divergence. Towards more flexibility, approximate posteriors can be defined as marginal distributions, $q_\phi(x|y) = \int \bar{q}_\phi(x,u|y)\mathrm{d}u$, where $u \in \mathsf{U}$ is an auxiliary variable (which can both have discrete ann dontinuous components) and $\bar{q}_\phi(x,u|y)$ is a generative closed-form density. Introducing auxiliary variables loses the tractability of (19) but they allow for their own ELBO as suggested in Agakov and Barber (2004); Lawson et al. (2019), leading to the objective

$$\int \bar{q}_\phi(x,u|y) \log\left(\frac{\bar{p}_\theta(x,u,y)}{\bar{q}_\phi(x,u|y)}\right) \mathrm{d}x\mathrm{d}u \ , \tag{20}$$

where $\bar{p}_\theta(x,u,y)$ is an extended joint likelihood satisfying $p_\theta(x,y) = \int \bar{p}_\theta(x,u,y)\mathrm{d}u$ (or equivalently $\bar{p}_\theta(x,u,y) = p_\theta(x,y)\bar{m}_\theta(x,y;u)$ where $\bar{m}_\theta(x,y;\cdot)$ is a Markov kernel). We now exploit this idea within the InFiNE framework. For that purpose, set prior, likelihood, and posterior as $\rho(x) = q_\phi(x \mid y)$, $\mathrm{L}(x) = p_\theta(x,y)/q_\phi(x \mid y)$, and $\pi(x) = p_\theta(x \mid y)$, respectively (the dependence of ρ, L, and π on both parameter (θ, ϕ) and observation y is implicit for notational simplicity). With these notations, the normalizing constant of $\rho(x)\mathrm{L}(x)$ is then $Z = p_\theta(y)$. The auxiliary variable u is naturally associated with the extended target $\bar{\boldsymbol{\pi}}$ defined in (18) (playing the role of $\bar{p}_\theta$), with

$$(x,u) = ([x, x^{1:N\setminus\{i\}}], i, k) \ ,$$

$[x, x^{1:N\setminus\{i\}}]$ being a shorthand notation for a N-tuple $x^{1:N}$ with $x^i = x$. An extended proposal playing the role of $\bar{q}_\phi(x,u|y)$ is derived from the InFiNE MCMC sampler, i.e.

$$\bar{\rho}(x^{1:N}, i, k) = \frac{\mathrm{L}(\mathrm{T}^k(x^i))w_k(x^i)}{N\widehat{Z}_{x^{1:N}}}\rho_N(x_{1:N}) \ . \tag{21}$$

where $\widehat{Z}_{x^{1:N}}$ is the InFiNE estimator (12) of the normalizing constant. Note that, by construction,

$$\sum_{i=1}^{N}\sum_{k=0}^{K} \bar{\rho}(x^{1:N}, i, k) = \rho_N(x^{1:N}) \tag{22}$$

showing that this joint proposal can be sampled by drawing the proposals $x^{1:N} \sim \rho_N$, then sampling the path index $i \in [N]$ with probability proportional to $(\widehat{Z}_{x^i})_{i=1}^N$ (with $\widehat{Z}_x$ defined in (11)) and finally the iteration index $k \in \{0, \ldots, K\}$ with probability proportional to $(w_k(x^i)\mathrm{L}(\mathrm{T}^k(x^i))_{k=0}^K$. Since the ratio of (18) over (21) is

$$\bar{\boldsymbol{\pi}}(x^{1:N}, i, k)/\bar{\rho}(x^{1:N}, i, k) = \widehat{Z}_{x^{1:N}}/Z \ . \tag{23}$$

The augmented ELBO (20) writes

$$\begin{aligned}\mathcal{L}_{\mathrm{InFiNE}} &= \int \rho_N(x^{1:N}) \log \widehat{Z}_{x^{1:N}} \mathrm{d}x^{1:N} \ , \\ &= \log Z - \mathrm{KL}(\bar{\rho}|\bar{\boldsymbol{\pi}}) \ ,\end{aligned} \tag{24}$$

where we have used (22) and that the ratio $\bar{\boldsymbol{\pi}}(x^{1:N}, i, k)/\bar{\rho}(x^{1:N}, i, k)$ does not depend on the path index i and the proposal index k along the path. When $K = 0$ and $\rho_N(x^{1:N}) = \prod j = 1^N \rho(x^j)$, we exactly retrieve the Importance Weighted AutoEncoder (IWAE); see e.g. Burda et al. (2016) and in particular the interpretation in Cremer et al. (2017).

Choosing the conformal Hamiltonian introduced in Section 2.2 allows for a family of invertible flows that depends on the parameter θ which itself is directly linked to the target distribution.

5 Numerical Experiments

5.1 Normalizing constant estimation

We first consider the problem of the estimation of the normalizing constant of Gaussian mixtures in dimension d in two different settings. In the first experiment, we consider an (unnormalized) mixture of two Gaussian distributions, with equal mixing weights. The mean of the two components are set to $(\mathbf{1}_d, -\mathbf{1}_d)$, where $\mathbf{1} = [1, \dots, 1]^T$ and covariance $\sigma^2 \,\mathrm{Id} = 0.02$. The second target is an unnormalized mixture of 25 d-dimensional Gaussian distributions in dimension $d = 10, 20$. Each component has the same covariance assumed to be diagonal with diagonal elements equal to $(0.01, 0.01, 0.1, \dots, 0.1)$. The means are given by $(i, j, 0, \dots, 0)$ with $i, j \in \{-2, \dots, 2\}$, see Figure 4. The normalizing constant in this case is 12.5. In both examples the proposal ρ is chosen to be a d-dimensional Gaussian, with zero mean and diagonal covariance $\sigma_\rho^2 \,\mathrm{Id}_d$, with $\sigma_\rho^2 = 5$. The performance of the InFiNE estimator (12) is first illustrated in this toy problem for $d \in \{5, 10, 15, 20\}$ and different choices of parameters.

Our approach is compared with a naïve IS estimator using the same proposal ρ. A state-of-the-art competitor for the estimation of normalizing constants, the AIS estimator of Neal (2001); Tokdar and Kass (2010) is also included in the comparison. AIS relies on a sequence of target distribution $\pi_k(x)$, $0 \leq k \leq K$ with $\pi_0(x) = \rho(x)$ and $\pi_K(x) = \pi(x)$. AIS defines an extended target and proposal using MCMC kernels which are reversible for each linking densities π_k; most often, these MCMC kernels use Langevin or Hamiltonian dynamics; see e.g. Buchholz et al. (2021). Therefore, AIS is directly comparable to the InFiNE estimator in terms of complexity. We focus here on the impact of the damping factor γ on the InFiNE estimations. Further investigation on the stepsize h and of the mass matrix M are given in the supplementary material.

The number of steps K is a proxy of our computational budget (*i.e.* the number of times our transformation is applied). The mass matrix M is chosen as the inverse of the covariance of the individual component of the mixture. Further tuning on this matrix is discussed in the supplementary material. A first intuition on the role of γ is shown in Figure 1. If $\gamma \ll 1$, then the trajectories are almost Hamiltonian, in which case we cannot easily explore all modes. On the other hand, if $\gamma \gg 1$, then trajectories are most often attracted by the "closest" mode. The resulting trade-off is easily observed on Figure 2, which displays the distribution of the different estimators.

The IS estimator is run with $4 \cdot 10^5$ samples. For the InFiNE estimator, the number of samples is $N = 2 \cdot 10^4$ and the trajectory length is $K = 20$. The stepsize is set to $h = 0.1$ for the conformal symplectic integrator. The number of levels for AIS in the annealing schedule of AIS is set to 200. At each intermediate temperature, an iteration of HMC is performed with 3 leapfrog steps (the size of leapfrog step is also set to 0.1). The number of gradient computations is therefore equal to $4 \cdot 10^5$ for InFiNE and $6 \cdot 10^6$ for AIS, which is therefore 10 times more costly.

We further emphasize how the InFiNE estimator compares favorably to the AIS estimator albeit requiring a smaller computational budget.

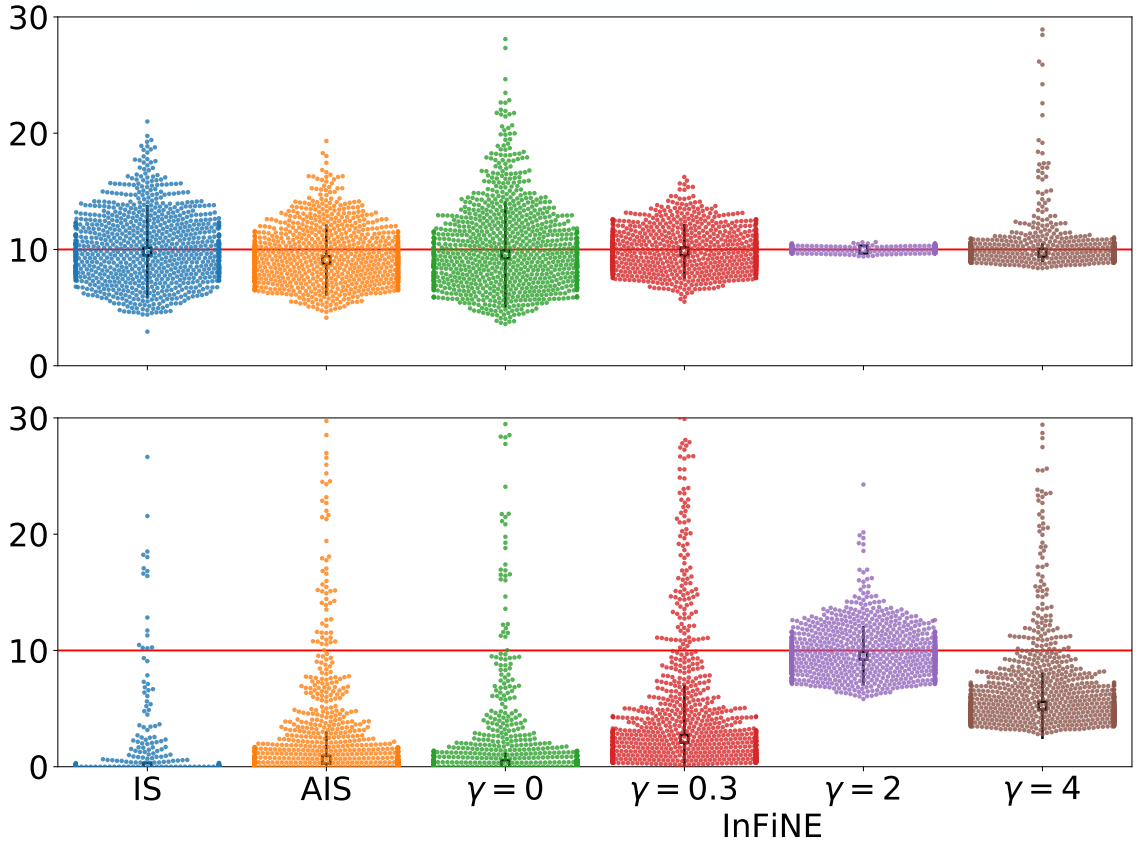


Figure 2: 1000 independent estimations of the normalizing constant for each algorithm in the toy example: a mixture of two Gaussian distributions, in dimension 5 (top) and 10 (bottom). The true value is $Z = 10$ (red line). The figure displays the median (square) and the interquartile range (solid lines) in each case.

The results for the 25-component Gaussian mixture are displayed in Figure 3. In high dimension, the vanilla IS estimator unsurprisingly fails, since importance weighted estimates have notoriously poor scaling properties w.r.t. dimension. While AIS predictably improves upon vanilla IS, its performances are rather unsatisfactory, the estimator showing a very large variance. Regardless of the dimension d, the InFiNE estimator is better behaved than the AIS estimator, although the computational burden for InFiNE is 10 times smaller.

5.2 MCMC experiments

We focus here on sampling of the 25 Gaussian mixture example introduced in Section 5.1. The dimension is set to $d = 40$ and all the mixture components have diagonal covariances $0.01\,\mathrm{Id}_d$. We compare the InFiNE MCMC sampler with dependent proposals, the No-U-Turn Sampler implemented with Pyro library Bingham et al. (2019), and the ISIR scheme Andrieu et al. (2010, 2018), with correlated proposal.

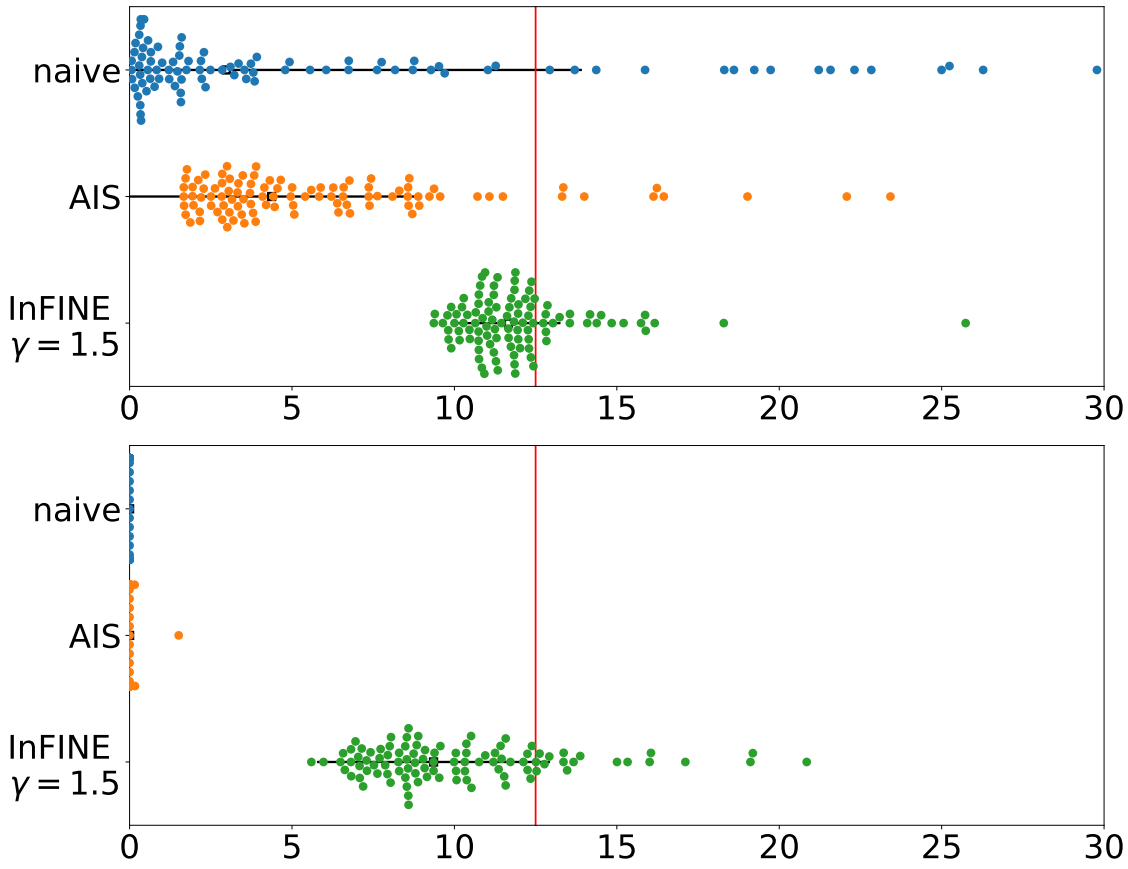


Figure 3: 100 independent estimations of the normalizing constant of the Gaussian mixture with 25 components in dimension 10 (top) and 20 (bottom) for each algorithm. The true value is $Z = 12.5$ (red line).

For the InFiNE MCMC, the number of particles is set to $N = 10$, the length of trajectory is $K = 10$, the stepsize of the conformal integrator is $h = 0.1$, the mass matrix is diagonal with diagonal elements equal to 100. The proposal distribution is zero-mean Gaussian with diagonal covariance $5\,\mathrm{Id}_d$. We use the proposal kernels r_i defined in (17) with a random walk Metropolis kernel m with zero-mean Gaussian increment distribution and covariance $0.01\,\mathrm{Id}$. For the iterated ISIR, we use the same number of proposals $N = 10$, proposal distribution ρ and proposal kernels r_i as for InFiNE. For NUTS, we use the default parameter (the mass matrix and stepsizes are adapted).

In Figure 4, the scatter plot of the first two components of the output is displayed. To make a fair comparison, we use the same wall clock time for all three algorithms. The number of iterations for CISIR, InFiNE, and NUTS are $n = 4 \cdot 10^6$, $n = 4 \cdot 10^5$, and $n = 5 \cdot 10^5$, respectively.

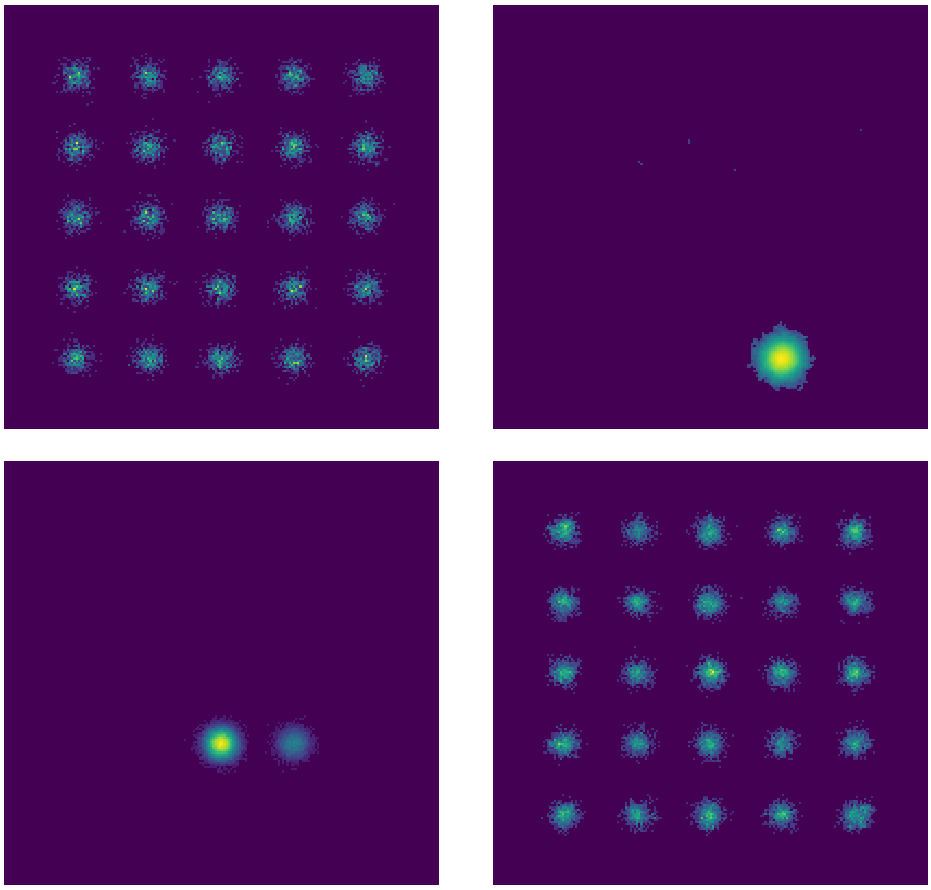


Figure 4: Empirical 2-D histogram of the 10,000 samples of different algorithms targeting the mixture of 25 Gaussian distributions. Top row, from left to right: samples from the target distribution, correlated ISIR samples. Bottom row: NUTS samples, InFiNE samples.

Table 1: Negative Log Likelihood estimates for VAE models for different latent space dimensions.

	$d = 4$		$d = 8$		$d = 16$		$d = 50$	
model	IS	InFiNE	IS	InFiNE	IS	InFiNE	IS	InFiNE
VAE	115.01	113.49	97.96	97.64	90.52	90.42	88.22	88.36
IWAE, $N = 5$	113.33	111.83	97.19	96.61	89.34	89.05	87.49	87.27
IWAE, $N = 30$	111.92	110.36	96.81	95.94	88.99	88.64	86.97	86.93
InFiNE VAE, $K = 3$	109.14	107.47	94.50	94.26	89.03	88.92	88.14	88.16
InFiNE VAE, $K = 10$	110.02	107.90	94.63	94.22	89.71	88.68	88.25	86.95

5.3 VAE experiments

Following Section 4, we propose numerical experiments to illustrate the relevance of InFiNE in the context of VAEs. We build InFiNE VAE by optimizing directly the ELBO (24) with respect to the parameters (θ, ϕ)

with a single trajectory. In practice, extending a standard VAE implementation with InFiNE is straightforward: after sampling the initial position q from the encoder distribution $q_\phi(\cdot \mid x)$, an initial moment p is sampled. The trajectory is then computed, followed by the weights and the ELBO. The *reparameterization trick* Kingma and Welling (2013) is used in a similar fashion as the VAE to ensure full differentiability of the whole architecture, enabling the optimization of all parameters. A full specification of the algorithm is provided in **??**.

We follow the experimental setting of Burda et al. (2016), using the MNIST dataset. Additional experiments on the FashionMNIST dataset are given in Appendix C.2. We compare our InFiNE VAE with a classical VAE, and IWAE (with $N = 5$ and $N = 30$ samples). For each setting, we use exactly the same architecture for the encoder and decoder, resulting in the same number of parameters. We followed as much as possible the implementation details (architecture and optimizer) detailed in Burda et al. (2016). All models are trained during 200 epochs.

Estimating the loglikelihood After training, VAEs are classically evaluated by computing an estimate of their negative loglikelihood (NLL) using either IS, the IWAE bound, or AIS Wu et al. (2016). We first show here how InFiNE competes with those methods for evaluating the NLL of VAEs. Note that the methods for evaluating VAEs always define a lower bound of the true likelihood (19). We can thus compare two evaluation methods if one consistently gives lower NLL estimates. Table 1 gives the NLL estimate for the different models (associated with a different dimension d of the latent variable). In both cases, the importance distribution is $q_\phi(\cdot \mid x)$. We set up the InFiNE estimator with a trajectory of length of $K = 10$, $h = 0.1$ and $\gamma = 2.5$ parameters. The InFiNE estimator consistently gives better estimates than the classical IS estimator.

Comparison of the different VAEs Table 1 displays the estimated NLL of all models provided by IS and the InFiNE method. It is interesting to note here again that InFiNE improves the training of the VAE when the dimension of the latent space is small to moderate. The relative improvement of InFiNE decreases when the dimension of the latent space increases, most likely because the mean-field variational distribution is accurate enough in such cases (this is at least the case for the MNIST dataset). InFiNE VAE has a better NLL than the VAE across all latent dimensions considered.

The results displayed in this section provide many insights for future research. Improving the NLL estimate in higher dimensions can be linked to the InFiNE hyperparameter tuning, which becomes crucial when the dimension increases. The optimal scaling of these hyperparameters remains an open (and challenging) problem left for future research as we aim here at highlighting the applicability of InFiNE in various contexts. Also, we have considered only the case $N = 1$. It is expected that extension to $N > 1$ (similar to IWAE) will further improve the results.

A Proofs of Section 2

A.1 Proof of Equation (5)

Let $f : \mathbb{R}^d \to \mathbb{R}_+$ be a measurable function and $k \in \{0, \dots, K\}$. 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.

A.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,

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

A.3 Proof of Lemma 2

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

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

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.

B Proofs of Section 3

B.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

$$\boldsymbol{\pi}(\mathrm{d}x) = \boldsymbol{\rho}(\mathrm{d}x)\mathrm{L}(x)/Z \ , \tag{25}$$

or equivalently, using Radon-Nikodym derivative

$$\frac{\mathrm{d}\boldsymbol{\pi}}{\mathrm{d}\boldsymbol{\rho}}(x) = \frac{\mathrm{L}(x)}{Z} \ . \tag{26}$$

For $k \in \{0, \ldots, K\}$, 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 $k \in \mathbb{N}$,

$$\int f(x)\boldsymbol{\rho}_k(\mathrm{d}x) = \int f(\mathrm{T}^k(x))\mathbb{1}_I(x,k)\boldsymbol{\rho}(\mathrm{d}x) \ . \tag{27}$$

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 $k \in \{0, \ldots, K\}$,

$$w_k(x) = \frac{1}{Z_{\mathrm{T}}}\frac{\mathrm{d}\boldsymbol{\rho}}{\mathrm{d}\boldsymbol{\rho}_T}(\mathrm{T}^k(x)) \ , \tag{28}$$

$$\boldsymbol{\rho}_T(\mathrm{d}x) = \frac{1}{Z_{\mathrm{T}}}\sum_{k=0}^{K}\boldsymbol{\rho}_k(\mathrm{d}x) \ . \tag{29}$$

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

$$\boldsymbol{\rho}(\mathrm{d}x^i)R_i(x^i; \mathrm{d}x^{1:N\setminus\{i\}}) = \boldsymbol{\rho}(\mathrm{d}x^j)R_j(x^j; \mathrm{d}x^{1:N\setminus\{j\}}) = \boldsymbol{\rho}_N(\mathrm{d}x^{1:N}) \ , \tag{30}$$

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 $\rho(\mathrm{d}x^i)$.

B.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 and Park (2008) of the extended distribution, given for $i \in \{1, \ldots, N\}$ by

$$\bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N}, i, \mathrm{d}y) = \frac{1}{N}\boldsymbol{\pi}(\mathrm{d}x^i)R_i(x^i, \mathrm{d}x^{1:N\setminus\{i\}})\delta_{x^i}(\mathrm{d}y) \ . \tag{31}$$

For ease of presentation, we added the selected sample y in the joint distribution. It is straightforward to establish that the marginal distributions of (31) are given by

$$\bar{\boldsymbol{\pi}}(\mathrm{d}y) = \boldsymbol{\pi}(\mathrm{d}y) \;, \tag{32}$$

$$\bar{\boldsymbol{\pi}}(i) = 1/N \;, \quad i \in \{1, \ldots, N\} \;, \tag{33}$$

$$\bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N}) = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{\pi}(\mathrm{d}x^i) R_i(x^i; \mathrm{d}x^{1:N\setminus\{i\}}) \;. \tag{34}$$

We now compute the conditional distributions and check that

$$K_1(i, y; \mathrm{d}x^{1:N}) = \bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N} \mid i, y) = \delta_y(\mathrm{d}x^i) R_i(x^i, \mathrm{d}x^{1:N\setminus\{i\}}) \;. \tag{35}$$

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 (35). We now establish that the conditional density of i satisfies

$$K_2(x_{1:n}; i) = \bar{\boldsymbol{\pi}}(i \mid x^{1:N}) = \frac{\mathrm{L}(x^i)}{\sum_{j=1}^{N} \mathrm{L}(x^j)} \;. \tag{36}$$

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 $\{f_j\}_{j=1}^N$,

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

where we have used (30). We conclude by noting that $\boldsymbol{\pi}(\mathrm{d}x) = \mathrm{L}(x)\boldsymbol{\rho}(\mathrm{d}x)/Z$ and using (34). We obviously have, by construction, that the conditional distribution of the auxiliary variable y satisfies

$$K_3(x^{1:N}, i; \mathrm{d}y) = \boldsymbol{\pi}(\mathrm{d}y \mid x^{1:N}, i) = \delta_{x^i}(\mathrm{d}y). \tag{37}$$

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 (35), 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

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

B.3 Invariance for InFiNE sampler

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

$$\bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N}, i, k, \mathrm{d}y) = \frac{1}{NZ} w_k(x^i) \mathrm{L}(\mathrm{T}^k(x^i)) \boldsymbol{\rho}(\mathrm{d}x^i) R_i(x^i; \mathrm{d}x^{1:N\setminus\{i\}}) \delta_{\mathrm{T}^k(x^i)}(\mathrm{d}y) \ . \tag{38}$$

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 and 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

$$\bar{\boldsymbol{\pi}}(i, k, \mathrm{d}y) = \frac{1}{NZ_{\mathrm{T}}} \frac{\mathrm{d}\boldsymbol{\pi}}{\mathrm{d}\boldsymbol{\rho}_T}(y) \boldsymbol{\rho}_k(\mathrm{d}y) \ . \tag{39}$$

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

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

which shows (39). Using (29),

$$\bar{\boldsymbol{\pi}}(\mathrm{d}y) = \sum_{i=1}^{N}\sum_{k=0}^{K}\bar{\boldsymbol{\pi}}(i,k,\mathrm{d}y) = \sum_{k=0}^{K}\frac{1}{Z_{\mathrm{T}}}\frac{\mathrm{d}\boldsymbol{\pi}}{\mathrm{d}\boldsymbol{\rho}_T}(y)\boldsymbol{\rho}_k(\mathrm{d}y) = \frac{\mathrm{d}\boldsymbol{\pi}}{\mathrm{d}\boldsymbol{\rho}_T}(y)\boldsymbol{\rho}_T(\mathrm{d}y) = \boldsymbol{\pi}(\mathrm{d}y)\ . \tag{40}$$

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

$$\bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N}, i) = \frac{\widehat{Z}_{x^i}}{NZ}\boldsymbol{\rho}_N(\mathrm{d}x^{1:N})\ , \tag{41}$$

where, see (11),

$$\widehat{Z}_x = \sum_{k=0}^{K}\mathrm{L}(\mathrm{T}^k(x))w_k(x)\ . \tag{42}$$

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

$$\bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N}) = \frac{\widehat{Z}_{x^{1:N}}}{Z}\boldsymbol{\rho}_N(\mathrm{d}x^{1:N})\ , \tag{43}$$

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 $i \in \{1,\ldots,N\}$ and $k \in \{0,\ldots,K\}$,

$$K_1(i,k,y;\mathrm{d}x^{1:N}) = \bar{\boldsymbol{\pi}}(\mathrm{d}x^{1:N} \mid i,k,y) = \delta_{\mathrm{T}^{-k}(y)}(\mathrm{d}x^i)R_i(x^i;\mathrm{d}x^{1:N\setminus\{i\}})\ . \tag{44}$$

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 (39). Next we show that, for $i \in \{1, \ldots, N\}$,

$$K_2(x_{1:N}; i) = \bar{\boldsymbol{\pi}}(i \mid x_{1:N}) = \frac{\widehat{Z}_{x^i}}{\sum_{j=1}^N \widehat{Z}_{x^j}} \,. \tag{45}$$

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 (43) in the last identity. This establishes (45). We finally prove that for $k \in \{0, \ldots, K\}$ and $i \in \{1, \ldots, N\}$,

$$K_3(i, x^{1:N}; k) = \bar{\boldsymbol{\pi}}(k \mid i, x^{1:N}) = \frac{w_k(\mathrm{T}^k(x^i)) \mathrm{L}(\mathrm{T}^k(x^i))}{\widehat{Z}_{x^i}} \,. \tag{46}$$

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 (41) in the last identity. This establishes (46). It follows directly from the definition of (38) that

$$K_4(x_{1:N},i,k;\mathrm{d}y)=\bar{\boldsymbol{\pi}}(\mathrm{d}y\mid x^{1:N},i,k)=\delta_{\mathrm{T}^k(x^i)}(\mathrm{d}y)\;. \tag{47}$$

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, (44), we use the full conditional. In the second step, (45) (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 (46) 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

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

B.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

$$P(i, y; j \times A) = \int \delta_y(\mathrm{d}x^i) R_i(x^i, \mathrm{d}x^{1:N\setminus\{i\}}) \frac{\mathrm{L}(x^j)}{\sum_{\ell=1}^N \mathrm{L}(x^\ell)} \delta_{x^j}(A) \ . \tag{48}$$

Consider the following assumptions:

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

$$\kappa = \inf_{x\in\mathbb{R}^d} \mathrm{L}(x) \Big/ \sup_{x\in\mathbb{R}^d} \mathrm{L}(x) > 0 \ . \tag{49}$$

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

$$R_{i,j}(x^i, A) = \int R_i(x^i, \mathrm{d}x^{1:N\setminus\{i\}}) \mathbb{1}_A(x^j) \ . \tag{50}$$

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 $i \neq j \in \{1, \ldots, N\}$*

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 4. *Assume **H**1 and **H**2. Then the conditional ISIR kernel P (see* (48)*) is irreducible, positive recurrent and ergodic. If for all $i \in \{1, \ldots, N\}$, $R_i(x^i; \mathrm{d}x^{1:N\setminus\{i\}}) = \prod_{j\neq i} \rho(\mathrm{d}x^j)$, then P is uniformly ergodic.*

Proof. For all $i \in \{1, \ldots, N\}$ 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 $D = \{1 \ldots, N\} \times C$ 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.

C Additional details about the experiments

C.1 Additional experiments

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

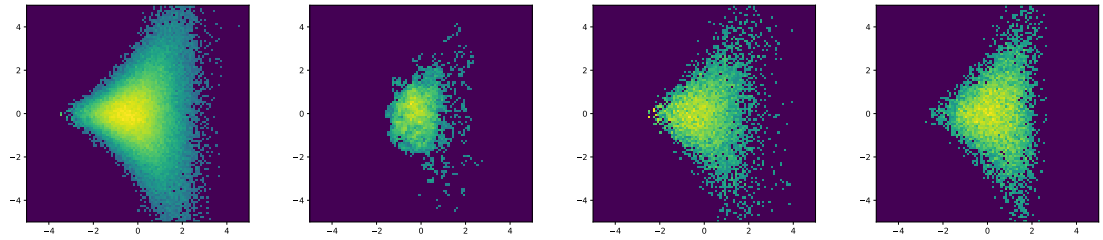
$$\pi(x) = \mathrm{N}(x_1; 0, a^2) \prod_{i=2}^{d} \mathrm{N}(x_i; 0, \mathrm{e}^{2bx_1}) \ ,$$

with $a = 1$ and $b = 0.5$ and where $x = (x_1, \ldots, x_d)$. 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 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

Figure 5: Empirical histograms of samples from the Funnel distribution. From left to right, target distribution (very long run of NUTS), ISIR, HMC and InFiNE



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.

C.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

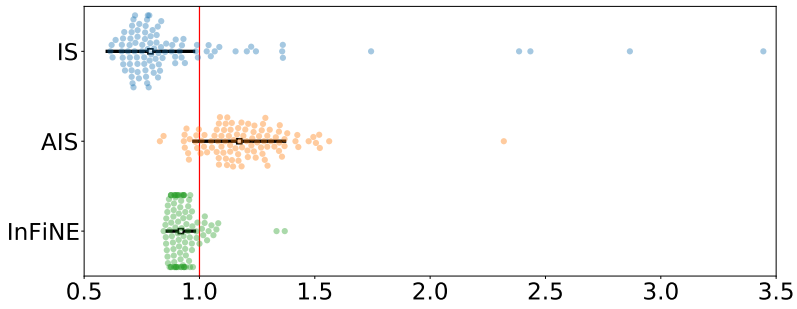


Figure 6: 200 independent estimations of the normalizing constant of π. The prior used is a centered Gaussian distribution with $4\mathbf{I}_d$ as covariance matrix. The true value is $Z = 1$ (red line). The figure displays the median (square) and the interquartile range (solid lines) in each case.

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

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

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 $\epsilon \sim \mathrm{g}$ 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

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

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

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

$$\begin{aligned}\nabla \widehat{\mathcal{L}}_{\mathrm{InFiNE}}(\theta,\phi;y) &= \sum_{i=1}^{N} \varpi_i \nabla \log \left(\sum_{k=0}^{K} w_k(V_{\phi,y}(\epsilon^i)) \frac{p_\theta(y, \mathrm{T}^k(V_{\phi,y}(\epsilon^i)))}{q_\phi(\mathrm{T}^k(V_{\phi,y}(\epsilon^i)) \mid y)} \right) \qquad (51) \\ &= \sum_{i=1}^{N} \varpi_i \nabla \log \widehat{Z}_{V_{\phi,y}(\epsilon^i)} \ , \quad \text{where } \epsilon^{1:n} \overset{\text{iid}}{\sim} \mathrm{g} \ , \qquad (52)\end{aligned}$$

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

Algorithm 2 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 ϖ_i.
end for
Compute the $\mathrm{ELBO}_{\theta,\phi}$ gradient estimator (51).
SGD update of parameters (θ, ϕ) using the gradient estimatior.

Table 2 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 2: 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

D 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 and Vanden-Eijnden (2019). The method proposed in Rotskoff and Vanden-Eijnden (2019) uses the identity

$$Z = \int \int_0^\infty \mathbb{1}(\mathrm{L}(x) > \ell)\rho(x)\mathrm{d}\ell\mathrm{d}x = \int_0^\infty \mathbb{P}_{X\sim\rho}(\mathrm{L}(X) > \ell)\mathrm{d}\ell\,, \tag{53}$$

which was instrumental in the construction of nested sampling Skilling (2006); Chopin and 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 and 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 and 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 and 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:

$$\widehat{\mathbb{P}}_{X\sim\rho}(\mathrm{L}(X) > \ell) = \frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K} \mathbb{1}_{\{\mathrm{L}(\mathrm{T}^k(X^i))>\ell\}} w_k(X^i) \;, \qquad X^i \overset{\text{iid}}{\sim} \rho \;, \tag{54}$$

where the weights $w_k(X^i)$, defined in (9), incorporate the stopping times. Integrating the above over $\ell \in \mathbb{R}^+$ as in (53) leads to an estimator of the normalizing constant Z:

$$\begin{aligned}\widehat{Z}_{X^{1:N}} &= \frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K} \int_{\mathbb{R}^+} \mathbb{I}(\mathrm{L}(\mathrm{T}^k(X^i)) > \ell) w_k(X^i) \mathrm{d}\ell \\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{k=0}^{K} \mathrm{L}(\mathrm{T}^k(X^i)) w_k(x^i) \;, \end{aligned} \tag{55}$$

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 (55).)

References

Agakov, F. V. and Barber, D. (2004). An auxiliary variational method. In *International Conference on Neural Information Processing*, pages 561–566. Springer.

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

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

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

Buchholz, A., Chopin, N., Jacob, P. E., et al. (2021). Adaptive tuning of Hamiltonian Monte Carlo within Sequential Monte Carlo. *Bayesian Analysis*.

Burda, Y., Grosse, R., and Salakhutdinov, R. (2016). Importance weighted autoencoders. In *The 4th International Conference on Learning Representations (ICLR)*.

Che, T., Zhang, R., Sohl-Dickstein, J., Larochelle, H., Paull, L., Cao, Y., and Bengio, Y. (2020). Your GAN is secretly an energy-based model and you should use discriminator driven latent sampling. *arXiv preprint arXiv:2003.06060*.

Chen, M.-H., Shao, Q.-M., and Ibrahim, J. G. (2000). *Monte Carlo Methods in Bayesian Computation*. Springer-Verlag, New York.

Chib, S. (1995). Marginal likelihood from the Gibbs output. *J. American Stat. Assoc.*, 90:1313–1321.

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

Cremer, C., Morris, Q., and Duvenaud, D. (2017). Reinterpreting importance-weighted autoencoders. *arXiv preprint arXiv:1704.02916*.

Cuendet, M. A. (2006). Statistical mechanical derivation of Jarzynski's identity for thermostated non-Hamiltonian dynamics. *Physical Review Letters*, 96(12):120602.

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B*, 68(3):411–436.

Ding, X. and Freedman, D. J. (2019). Learning deep generative models with annealed importance sampling. *arXiv preprint arXiv:1906.04904*.

Douc, R., Moulines, E., Priouret, P., and Soulier, P. (2018). *Markov chains*. Springer Series in Operations Research and Financial Engineering. Springer, Cham.

El Moselhy, T. A. and Marzouk, Y. M. (2012). Bayesian inference with optimal maps. *Journal of Computational Physics*, 231(23):7815–7850.

Franca, G., Sulam, J., Robinson, D. P., and Vidal, R. (2019). Conformal symplectic and relativistic optimization. *arXiv preprint arXiv:1903.04100*.

Gelman, A. and Meng, X.-L. (1998). Simulating normalizing constants: From importance sampling to bridge sampling to path sampling. *Statistical Science*, pages 163–185.

Geyer, C. (1993). Estimating normalizing constants and reweighting mixtures in Markov chain Monte Carlo. Technical Report 568, School of Statistics, Univ. of Minnesota.

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

Gutmann, M. U. and Hyvärinen, A. (2012). Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(1):307–361.

Jarzynski, C. (2002). Targeted free energy perturbation. *Physical Review E*, 65(4):046122.

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

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*.

Kingma, D. P. and Welling, M. (2019). An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691*.

Kong, A., McCullagh, P., Meng, X.-L., Nicolae, D., and Tan, Z. (2003). A theory of statistical models for Monte Carlo integration (with discussion). *Journal of the Royal Statistical Society (Series B)*, 65(3):585–618.

Lawson, D., Tucker, G., Dai, B., and Ranganath, R. (2019). Energy-inspired models: Learning with sampler-induced distributions. *arXiv preprint arXiv:1910.14265*.

Lindsten, F., Douc, R., and Moulines, E. (2015). Uniform ergodicity of the particle Gibbs sampler. *Scandinavian Journal of Statistics*, 42(3):775–797.

Maddison, C. J., Paulin, D., Teh, Y. W., O'Donoghue, B., and Doucet, A. (2018). Hamiltonian descent methods. *arXiv preprint arXiv:1809.05042*.

Meng, X.-L. and Schilling, S. (2002). Warp bridge sampling. *Journal of Computational and Graphical Statistics*, 11(3):552–586.

Mnih, A. and Rezende, D. J. (2017). Variational inference for Monte Carlo objectives. In *International Conference on International Conference on Machine Learning*, page 2188–2196.

Müller, T., McWilliams, B., Rousselle, F., Gross, M., and Novák, J. (2019). Neural importance sampling. *ACM Transactions on Graphics*, 38(145).

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, 11:125–139.

Neal, R. M. (2005). Hamiltonian importance sampling. `www.cs.toronto.edu/pub/radford/his-talk.ps`. Talk presented at the Banff International Research Station (BIRS) workshop on Mathematical Issues in Molecular Dynamics.

Newton, M. and Raftery, A. (1994). Approximate Bayesian inference by the weighted likelihood bootstrap (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 56:1–48.

Owen, A. and Zhou, Y. (2000). Safe and effective importance sampling. *Journal of the American Statistical Association*, 95(449):135–143.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2019). Normalizing flows for probabilistic modeling and inference. *arXiv preprint arXiv:1912.02762*.

Prangle, D. (2019). Distilling importance sampling. *arXiv preprint arXiv:1910.03632*.

Procacci, P., Marsili, S., Barducci, A., Signorini, G. F., and Chelli, R. (2006). Crooks equation for steered molecular dynamics using a Nosé-Hoover thermostat. *The Journal of Chemical Physics*, 125(16):164101.

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

Rubin, D. B. (1987). Comment: A noniterative sampling/importance resampling alternative to the data augmentation algorithm for creating a few imputations when fractions of missing information are modest: The SIR algorithm. *Journal of the American Statistical Association*, 82(398):542–543.

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

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

Smith, A. F. and Gelfand, A. E. (1992). Bayesian statistics without tears: a sampling–resampling perspective. *The American Statistician*, 46(2):84–88.

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

Tokdar, S. T. and Kass, R. E. (2010). Importance sampling: a review. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(1):54–60.

Turner, R., Hung, J., Frank, E., Saatchi, Y., and Yosinski, J. (2019). Metropolis-Hastings generative adversarial networks. In *International Conference on Machine Learning*, pages 6345–6353. PMLR.

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

Wirnsberger, P., Ballard, A. J., Papamakarios, G., Abercrombie, S., Racanière, S., Pritzel, A., Rezende, D. J., and Blundell, C. (2020). Targeted free energy estimation via learned mappings. *arXiv preprint arXiv:2002.04913*.

Wu, Y., Burda, Y., Salakhutdinov, R., and Grosse, R. (2016). On the quantitative analysis of decoder-based generative models. *arXiv preprint arXiv:1611.04273*.

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