Annealed Flow Transport Monte Carlo

Michael Arbel [*] [1] **Alexander G. D. G. Matthews** [*] [2] **Arnaud Doucet** [2]

Abstract

Annealed Importance Sampling (AIS) and its Sequential Monte Carlo (SMC) extensions are state-of-the-art methods for estimating normalizing constants of probability distributions. We propose here a novel Monte Carlo algorithm, Annealed Flow Transport (AFT), that builds upon AIS and SMC and combines them with normalizing flows (NFs) for improved performance. This method transports a set of particles using not only importance sampling (IS), Markov chain Monte Carlo (MCMC) and resampling steps - as in SMC, but also relies on NFs which are learned sequentially to push particles towards the successive annealed targets. We provide limit theorems for the resulting Monte Carlo estimates of the normalizing constant and expectations with respect to the target distribution. Additionally, we show that a continuous-time scaling limit of the population version of AFT is given by a Feynman–Kac measure which simplifies to the law of a controlled diffusion for expressive NFs. We demonstrate experimentally the benefits and limitations of our methodology on a variety of applications.

1. Introduction

Let π be a target density on $\mathcal{X} \subseteq \mathbb{R}^d$ w.r.t. the Lebesgue measure known up to a normalizing constant Z. We want to estimate Z and approximate expectations with respect to π. This has applications in Bayesian statistics but also variational inference (VI) (Mnih and Rezende, 2016) and compression (Li and Chen, 2019; Huang et al., 2020) among others. AIS (Neal, 2001) and its SMC extensions (Del Moral et al., 2006) are state-of-the art Monte Carlo methods addressing this problem which rely on a sequence of annealed targets $\pi_k \propto \pi_0^{1-\beta_k}\pi_K^{\beta_k}$ bridging smoothly an easy-to-sample distribution π_0 to $\pi_K := \pi$ for $0 = \beta_0 < \beta_1 < \cdots < \beta_K = 1$ and MCMC kernels of invariant distributions π_k (Zhou et al., 2016; Llorente et al., 2020). In their simplest instance, SMC samplers propagate N particles approximating π_k at time k. These particles are reweighted according to weights proportional to π_{k+1}/π_k at time $k+1$ to build an IS approximation of π_{k+1}, then one resamples N times from this approximation and finally mutate the resampled particles according to MCMC steps of invariant distribution π_{k+1}. This procedure can provide high-variance estimators if the discrepancy between π_k and π_{k+1} is significant as the resulting IS weights then have a large variance and/or if the MCMC kernels mix poorly. This can be reduced by increasing K and the number of MCMC steps at each temperature but comes at an increasing computational cost.

An alternative approach is to build a transport map $T : \mathcal{X} \rightarrow \mathcal{X}$ to ensure that if $X \sim \pi_0$ then the distribution of $X' = T(X)$ denoted $T_\#\pi_0$ is approximately equal to π. In (El Moselhy and Marzouk, 2012), this map is parameterized using a polynomial chaos expansion and learned by minimizing a regularized Kullback-Leibler (KL) divergence between $T_\#\pi_0$ and π; see also (Marzouk et al., 2016). Taghvaei et al. (2020) and Olmez et al. (2020) obtain transport maps by solving a Poisson equation. However, they do not correct for the discrepancy between $T_\#\pi_0$ and π using IS. Doing so would incur a $O(d^3)$ cost when computing the Jacobian. Normalizing Flows (NFs) are an alternative flexible class of diffeomorphisms with easy-to-compute Jacobians (Rezende and Mohamed, 2015). These can be used to parameterize T and are also typically learned by minimizing KL$(T_\#\pi_0||\pi)$ or a regularized version of it. This approach has been investigated in many recent work; see e.g. (Gao et al., 2020; Nicoli et al., 2020; Noé et al., 2019; Wirnsberger et al., 2020). Although it is attractive, it is also well-known that optimizing this 'mode-seeking' KL can lead to an approximation of the target $T_\#\pi_0$ which has thinner tails than the target π and ignore some of its modes; see e.g. (Domke and Sheldon, 2018).

In this paper, our contributions are as follows.

- We propose Annealed Flow Transport (AFT), a methodology that takes advantages of the strengths of both

[*] Equal contribution [1] Gatsby Computational Neuroscience Unit, University College London [2] DeepMind. Correspondence to: Michael Arbel <michael.n.arbel@gmail.com>, Alexander G. D. G. Matthews <alexmatthews@google.com>, Arnaud Doucet <arnauddoucet@google.com>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

SMC and NFs. Given particles approximating π_k at time k, we learn a NF T_{k+1} minimizing the KL between $(T_{k+1})_{\#}\pi_k$ and π_{k+1}. As π_k is closer to π_{k+1} than π_0 is from $\pi_K = \pi$, learning such a NF is easier and less prone to mode collapse. Additionally the use of MCMC steps in SMC samplers allows the particles to diffuse and further prevent such collapse. Having obtained T_{k+1}, we then apply this mapping to the particles before building an IS approximation of π_{k+1} and then use resampling and MCMC steps.

- We establish a weak law of large numbers and a Central Limit Theorem (CLT) for the resulting Monte Carlo estimates of Z and expectations w.r.t. π. Available CLT results for SMC (Chopin, 2004; Del Moral, 2004; Künsch, 2005; Beskos et al., 2016) do not apply here as the transport maps are learned from particles.
- When one relies on Unadjusted Langevin algorithm (ULA) kernels to mutate particles, a time-rescaled population version of AFT without resampling is shown to converge as $K \to \infty$ towards a Feynman–Kac measure. For NFs expressive enough to include exact transport maps between successive distributions, this measure corresponds to the measure induced by a controlled Langevin diffusion.
- We demonstrate the performance of AFT on a variety of benchmarks, showing that it can improve over SMC for a given number of temperatures.

Related Work. The use of deterministic maps with AIS (Vaikuntanathan and Jarzynski, 2011) and SMC (Akyildiz and Míguez, 2020; Everitt et al., 2020; Heng et al., 2021) has already been explored. However, Everitt et al. (2020) and Vaikuntanathan and Jarzynski (2011) do not propose a generic methodology to build such maps while Akyildiz and Míguez (2020) introduce mode-seeking maps and do not correct for the incurred bias. Heng et al. (2021) rely on quadrature and a system of time-discretized nonlinear ordinary differential equations: this can be computationally cheaper than learning NFs but is application specific. NFs benefit from easy-to-compute Jacobians and a large and quickly expanding literature (Papamakarios et al., 2019); e.g., as both MCMC and NFs on manifolds have been developed, our algorithm can be directly extended to such settings.

Evidence Lower Bounds (ELBOs) based on unbiased estimators of Z have also been mentioned in (Salimans et al., 2015; Goyal et al., 2017; Caterini et al., 2018; Huang et al., 2018; Wu et al., 2020; Thin et al., 2021). These estimators generalize AIS, and are obtained using sequential IS, transport maps and MCMC. However, when MCMC kernels such as Metropolis–Hastings (MH) or Hamiltonian Monte Carlo (HMC) are used, accept/reject steps lead to high variance estimates of ELBO gradients (Thin et al., 2021). Moreover, while SMC (i.e. combining sequential IS and resampling) can also be used to define an ELBO, resampling steps correspond to sampling discrete distributions and lead to high variance gradient estimates; see e.g. (Maddison et al., 2017; Le et al., 2018; Naesseth et al., 2018) in the context of state-space models. The algorithm proposed here does not rely on the ELBO, so it can use arbitrary MCMC kernels and exploit the benefits of resampling. Moreover, it only requires a single pass through the $K+1$ annealed distributions: there is no need to iteratively run sequential IS or SMC for estimating Z and an ELBO gradient estimate.

Optimal control ideas have also been proposed to improve SMC by introducing an additive drift to a time-inhomogeneous ULA to improve sampling; see Richard and Zhang (2007); Kappen and Ruiz (2016); Guarniero et al. (2017); Heng et al. (2020). The proposed iterative algorithms require estimating value functions but, to be implementable, the approximating function class has to be severely restricted. The algorithm proposed here is much more widely applicable and can use sophisticated MCMC kernels.

Finally, alternative particle methods based on gradient flows in the space of probability measures have been proposed to provide an approximation of π, such as Stein Variational Gradient Descent (SVGD) (Liu and Wang, 2016; Liu et al., 2019; Wang and Li, 2019; Zhu et al., 2020; Reich and Weissmann, 2021). However, their consistency results require both K, the number of time steps, and N, the number of particles, to go to infinity. In contrast, AFT only needs $N \to \infty$. Moreover, they require specifying a suitable Reproducing Kernel Hilbert Space or performing kernel density estimation, which can be challenging in high dimension. Additionally, contrary to AFT, these methods do not provide an estimate of Z. One recent exception is the work of Han and Liu (2017) which combines SVGD with IS to estimate Z but this requires computing Jacobians of computational cost $O(d^3)$.

2. Sequential Monte Carlo samplers

We provide here a brief overview of SMC samplers and their connections to AIS. More details can be found in (Del Moral et al., 2006; Dai et al., 2020).

We will rely on the following notation for the annealed densities $(\pi_k)_{0\leq k\leq K}$ targeted by SMC:

$$\pi_k(x) = \frac{\gamma_k(x)}{Z_k} = \frac{\exp(-V_k(x))}{Z_k},$$

where $Z_0 = 1$ so $\pi_0(x) = \gamma_0(x)$ and $V_k(x) = (1-\beta_k)V_0 + \beta_k V_K$ for $0 = \beta_0 < \beta_1 < \cdots < \beta_K = 1$. However, we could use more generally any sequence of distributions bridging smoothly π_0 to $\pi_K = \pi$.

2.1. Sequential importance sampling

Let us first ignore the key resampling steps used by SMC. In this case, SMC boils down to a sequential IS technique where one approximates π_k at time k. We first sample $X_0 \sim \pi_0$ at time $k = 0$, then at time $k \geq 1$, obtain a a new sample $X_k \sim M_k(X_{k-1}, \cdot)$ using a Markov kernel M_k. For the distribution of X_k to be closer to π_k than the one of X_{k-1}, M_k is typically selected as a MCMC kernel of invariant density π_k such as MH or HMC, or of approximate invariant density π_k such as ULA. Hence, by construction, the joint density of $X_{0:k}$ is

$$\bar{\eta}_k(x_{0:k}) = \pi_0(x_0) \textstyle\prod_{l=1}^{k} M_l(x_{l-1}, x_l). \tag{1}$$

The resulting marginal η_k of X_k under $\bar{\eta}_k$ usually differs from π_k. If one could evaluate η_k pointwise, then IS could be used to correct for the discrepancy between η_k and π_k using the IS weight $w_k(x_k) = \gamma_k(x_k)/\eta_k(x_k)$. Unfortunately, η_k is intractable in all but toy scenarios. Instead, SMC samplers introduce joint target densities $\bar{\pi}_k(x_{0:k})$ to compute tractable IS weights $w_k(x_{0:k})$ over the whole path $X_{0:k}$ defined by

$$\bar{\pi}_k(x_{0:k}) = \pi_k(x_k) \textstyle\prod_{l=0}^{k-1} L_l(x_{l+1}, x_l), \tag{2}$$

here L_l are "backward" Markov kernels moving each sample X_{l+1} into a sample X_l starting from a *virtual* sample X_k from π_k[1]. Hence by construction π_k is the marginal of $\bar{\pi}_k$ at time k. The backward kernels L_{k-1} are chosen so that the following *incremental IS weights* are well-defined

$$G_k(x_{k-1}, x_k) = \frac{\gamma_k(x_k) L_{k-1}(x_k, x_{k-1})}{\gamma_{k-1}(x_{k-1}) M_k(x_{k-1}, x_k)}, \tag{3}$$

and, from (1) and (2), one obtains

$$w_k(x_{0:k}) := \frac{\bar{\gamma}_k(x_{0:k})}{\bar{\eta}_k(x_{0:k})} = \prod_{l=1}^{k} G_l(x_{l-1}, x_l), \tag{4}$$

where $\bar{\gamma}_k(x_{0:k}) = Z_k \bar{\pi}_k(x_{0:k})$ is the unnormalized joint target. Using IS, it is thus straightforward to check that

$$Z_k = \bar{\eta}_k[w_k], \quad \bar{\pi}_k[f] = \frac{\bar{\eta}_k[w_k f]}{\bar{\eta}_k[w_k]}, \tag{5}$$

where $f(x_{0:k})$ is a function of the whole trajectory $x_{0:k}$ and $\mu[g]$ is a shorthand notation for the expectation $\mathbb{E}_{X\sim\mu}[g(X)]$. As π_k is a marginal of $\bar{\pi}_k$, we can also estimate expectations w.r.t. to π_k using $\bar{\pi}_k[f] = \pi_k[f]$ for $f(x_{0:k}) = f(x_k)$. From (5), it is thus possible to derive consistent estimators of Z_k and $\pi_k[f]$ by sampling N 'particles' $X^i_{0:k} \sim \bar{\eta}_k$ where $i = 1, ..., N$ and using

$$Z^N_k = \frac{1}{N}\sum_{i=1}^{N} w_k(X^i_{0:k}), \quad \pi^N_k[f] = \sum_{i=1}^{N} W^i_k f(X^i_k), \tag{6}$$

where $W^i_k \propto w_k(X^i_{0:k})$, $\sum_{i=1}^N W^i_k = 1$.

When the kernels M_k are π_k-invariant and we select L_{k-1} as the reversal of M_k, i.e. $\pi_k(x)M_k(x, x') = \pi_k(x')L_{k-1}(x', x)$, it is easy to check that $G_l(x_{l-1}, x_l) = \gamma_l(x_{l-1})/\gamma_{l-1}(x_{l-1})$. In that case, (5) corresponds to AIS (Neal, 2001) and is also known as the Jarzynski–Crooks identity (Jarzynski, 1997; Crooks, 1998). When π_k is a sequence of posterior densities, a similar construction was also used in (MacEachern et al., 1999; Gilks and Berzuini, 2001; Chopin, 2002). The generalized identity (5) allows the use of more general dynamics, including deterministic maps which will be exploited by our algorithm.

In practice, the choice of the backward transition kernels has a large impact on the variance of the estimates (6). (Del Moral et al., 2006) identified the backward kernels minimizing the variance of the IS weights (3)-(4) and proposed various approximations to them.

2.2. Sequential Monte Carlo

To reduce the variance of the IS estimators (6), SMC samplers combine sequential IS steps with resampling steps. Given an IS approximation $\pi^N_{k-1} = \sum_{i=1}^N W^i_{k-1}\delta_{X^i_{k-1}}$ of π_{k-1} at time $k-1$, one resamples N times from π^N_{k-1} to obtain particles approximately distributed according to π_{k-1}. This has for effect of discarding particles with low weights and replicating particles with high weights, this helps focusing subsequent computation on "promising" regions of the space. Empirically, resampling usually provides lower variance unbiased estimates of normalizing constants and is computationally very cheap; see e.g. (Chopin, 2002; Hukushima and Iba, 2003; Del Moral et al., 2006; Rousset and Stoltz, 2006; Zhou et al., 2016; Barash et al., 2017). The resampled particles are then evolved according to M_k, weighted according to G_k and resampled again.

3. Annealed Flow Transport Monte Carlo

We now introduce AFT, a new flexible adaptive Monte Carlo method that leverages NFs. Given the particle approximations $\pi^N_{k-1} := \sum_{i=1}^N W^i_{k-1}\delta_{X^i_{k-1}}$ and Z^N_{k-1} at time $k-1$, AFT computes an approximation π^N_k and Z^N_k by performing four main sub-steps: *Transport, Importance Sampling, Resampling and Mutation*, as summarized in Algorithm 1. Whenever the index i is used in the algorithm, we mean 'for all $i \in \{1, ..., N\}$'. These four sub-steps are now detailed below.

[1] As in (Crooks, 1998; Neal, 2001; Del Moral et al., 2006; Dai et al., 2020), we do not use measure-theoretic notation here but it should be kept in mind that the kernels M_l do not necessarily admit a density w.r.t. Lebesgue measure; e.g. a MH kernel admits an atomic component. For completeness, a formal measure-theoretic presentation of the results of this section is given in Appendix A.

Algorithm 1 Annealed Flow Transport

```
Input: number of particles $N$, unnormalized annealed targets $\{\gamma_k\}_{k=0}^K$ such that $\gamma_0 = \pi_0$ and $\gamma_K = \gamma$, resampling threshold $A \in [1/N, 1)$.
Ouput: Approximations $\pi_K^N$ and $Z_K^N$ of $\pi$ and $Z$.
Sample $X_0^i \sim \pi_0$ and set $W_0^i = \frac{1}{N}$ and $Z_0^N = 1$.
for $k = 1, \dots, K$ do
    Compute $\mathcal{L}_k^N(T)$ using (8).
    Solve $T_k \leftarrow \text{argmin}_{T \in \mathcal{T}} \mathcal{L}_k^N(T)$ using e.g. SGD.
    Transport particles: $\widetilde{X}_k^i = T_k(X_{k-1}^i)$.
    Estimate normalizing constant $Z_k$:
    $Z_k^N \leftarrow Z_{k-1}^N \left( \sum_{i=1}^N W_{k-1}^i G_{k,T_k}(X_{k-1}^i) \right)$.
    Compute IS weights:
    $w_k^i \leftarrow W_{k-1}^i G_{k,T_k}(X_{k-1}^i)$ // unnormalized
    $W_k^i \leftarrow \frac{w_k^i}{\sum_{j=1}^N w_k^j}$ // normalized
    Compute effective sample size $\text{ESS}_k^N$ using (10).
    if $\text{ESS}_k^N / N \leq A$ then
        Resample $N$ particles denoted abusively also $\widetilde{X}_k^i$ according to the weights $W_k^i$, then set $W_k^i = \frac{1}{N}$.
    end if
    Sample $X_k^i \sim K_k(\widetilde{X}_k^i, \cdot)$. // MCMC
end for
```

3.1. Transport map estimation

In this step, we learn a NF T_k that moves each sample X_{k-1} from π_{k-1} to a sample $\widetilde{X}_k = T_k(X_{k-1})$ as close as possible to π_k by minimizing an estimate of $\text{KL}(T_\# \pi_{k-1} || \pi_k)$ over a set $\mathcal{T}$ of NFs. This KL can be decomposed as a sum of a loss term $\mathcal{L}_k(T)$ and a term $\log \frac{Z_k}{Z_{k-1}}$ that can be ignored as it is independent of the NF T. A simple change of variables allows us to express the loss term $\mathcal{L}_k(T)$ as an expectation under π_{k-1} of some tractable function $x \mapsto h_T(x)$:

$$\begin{aligned} \mathcal{L}_k(T) :=& \pi_{k-1}[h_T], \\ h_T(x) :=& V_k(T(x)) - V_{k-1}(x) - \log |\nabla T(x)|. \end{aligned} \tag{7}$$

The Jacobian determinant of T in (7) can be evaluated efficiently for NFs while the expectation under π_{k-1} can be estimated using π_{k-1}^N thus yielding the empirical loss:

$$\mathcal{L}_k^N(T) := \textstyle\sum_{i=1}^N W_{k-1}^i h_T(X_{k-1}^i). \tag{8}$$

In practice, (8) is optimized over the NF parameters using gradient descent. The resulting NF T_k is then used to transport each particle X_{k-1}^i to $\widetilde{X}_k^i = T_k(X_{k-1}^i)$[2]. However, the loss (8) being not necessarily convex, the solution T_k is likely to be sub-optimal. This is not an issue, since IS is used to correct for such approximation error as we will see next. We also emphasize that the convergence results for this scheme presented in Section 4 do not require finding a global minimizer of this non-convex optimization problem.

3.2. Importance Sampling, Resampling and Mutation

Importance Sampling. This step corrects for the NF T_k being only an approximate transport between π_{k-1} and π_k. In this case, we have $M_k^{\text{trans}}(x, x') = \delta_{T_k(x)}(x')$ and by selecting $L_{k-1}^{\text{trans}}(x, x') = \delta_{T_k^{-1}(x')}(x)$ then the incremental IS weight (3) is given by a simple change-of-variables formula

$$G_{k,T_k}(x_{k-1}) = \frac{\gamma_k(T_k(x_{k-1}))|\nabla T_k(x_{k-1})|}{\gamma_{k-1}(x_{k-1})}. \tag{9}$$

Using (9), we can update the weights $w_k^i = W_{k-1}^i G_{k,T_k}(X_{k-1}^i)$ to account for the errors introduced by T_k. When T_k are exact transport maps from π_{k-1} to π_k, the incremental weight in (9) becomes constant and equal to the ratio Z_k / Z_{k-1}. Thus, introducing the NF T_k can be seen as a way to reduce the variance of the IS weights in the SMC sampler.

Resampling. As discussed in Section 2.2, resampling can be very beneficial but it should only be performed when the variance of the IS weights is too high (Liu and Chen, 1995) as measured by the Effective Sample Size (ESS)

$$\text{ESS}_k^N = \left(\sum_{i=1}^N \left(W_k^i \right)^2 \right)^{-1}, \tag{10}$$

which is such that $\text{ESS}_k^N \in [1, N]$. When ESS_k^N / N is smaller than some prescribed threshold $A \in [1/N, 1)$ (we use $A = 0.3$), resampling is triggered and each particle $\widetilde{X}_k^i$ is then resampled without replacement from the set of N available particles $\{\widetilde{X}_k^i\}_{i \in [1:N]}$ according to a multinomial distribution with weights $\{W_k^i\}_{i \in [1:N]}$. The weights are then reset to uniform ones; i.e. $W_k^i = \frac{1}{N}$. More sophisticated lower variance resampling schemes have also been proposed; see e.g. (Kitagawa, 1996; Chopin, 2004).

Mutation. The final step consists in mutating the particles using a π_k-invariant MCMC kernel K_k , i.e. using $X_k^i \sim K_k(\widetilde{X}_k^i, \cdot)$. This allows particles to better explore the space.

Note that if the transport maps T_k were known, Algorithm 1 could be reinterpreted as a specific instance of a SMC as detailed in Section 2 where at each time $k \geq 1$ we perform two time steps of a standard SMC sampler by applying first a transport step $M_k^{\text{trans}}(x, x') = \delta_{T_k(x)}(x')$ then a mutation step $M_k^{\text{mut}}(x, x') = K_k(x, x')$; see Appendix B.1 for details.

[2] We should write T_k^N to indicate the dependence of our estimate of N but do not to simplify notation.

3.3. Variants and Extensions

Contrary to standard SMC, the estimates Z_k^N returned by Algorithm 1 are biased because of the dependence of the NF T_k on the particles. To obtain unbiased estimates of Z_k and to avoid over-fitting of the NF to the N particles, a variant of Algorithm 1 described in Algorithm 2 (see Appendix F) is used in the experimental evaluation. This variant employs three sets of particles: the *training set* is used to evaluate the loss (8), the *validation set* is used in a stopping criterion when learning the NF and the *test set* is independent from the rest and is computed sequentially using the learned NFs. It would also be possible to combine AFT with various extensions to SMC that were already proposed in the literature. For example, we can select adaptively the annealing parameters β_k to ensure the ESS only decreases by a pre-determined percentage (Jasra et al., 2011; Schäfer and Chopin, 2013; Beskos et al., 2016; Zhou et al., 2016) or use the approximation of π_k obtained at step 13 of Algorithm 1 to determine the parameters of the MCMC kernel K_k (Del Moral et al., 2012a; Buchholz et al., 2021).

4. Asymptotic analysis

We establish here a law of large numbers and a CLT for the particle estimates $\pi_k^N[f]$ and Z_k^N of $\pi_k[f]$ and Z_k. We denote by $\xrightarrow{P}$ convergence in probability and by $\xrightarrow{\mathcal{D}}$ convergence in distribution.

4.1. Weak law of large numbers

Theorem 1 shows that $\pi_k^N[f]$ and Z_k^N are consistent estimators of $\pi_k[f]$ and Z_k, hence of $\pi[f]$ and Z at time $k = K$.

Theorem 1 (weak law of large numbers)**.** *Let f be a function s.t. $|f(x)| \leq C(1 + \|x\|^4)$ for all $x \in \mathcal{X}$ and for some $C > 0$. Under Assumptions (A) to (D) and for any $k \in 0, ..., K$:*

$$(\mathcal{R}_k): \qquad \pi_k^N[f] \xrightarrow{P} \pi_k[f], \quad Z_k^N \xrightarrow{P} Z_k.$$

The result is proven in Appendix C.3 and relies on four assumptions stated in Appendix C.1: **(A)** on the smoothness of the Markov kernels K_k, **(B)** on the moments of π_k, **(C)** on the smoothness of the family of NFs and **(D)** on the boundedness of the incremental IS weight $G_{k,T}(x)$. Perhaps surprisingly, Theorem 1 does not require the NFs to converge as $N \to \infty$. This is a consequence of Proposition 9 in Appendix C.3 which ensures uniform consistency of the particle approximation regardless of the choice of the NFs. However, convergence of the NFs is required to obtain a CLT result as we see next. Theorem 4 of Appendix C.3 states a similar result for Algorithm 2 of Appendix F.

4.2. Central Limit theorem

Besides assumptions **(A)** to **(D)**, we make five assumptions stated in Appendix C.1: **(E)** on the Markov kernels K_k strengthens **(A)** and is satisfied by many commonly used Markov kernels as shown in C.2. The smoothness assumptions **(F)** and **(G)** on the family $\mathcal{T}$ of NFs and potentials V_k are also standard. Finally, **(H)** and **(I)** describe the asymptotic behavior of T_k. We do not require T_k to be a global minimizer of the loss $\mathcal{L}_k^N$, neither do we assume it to be an exact local minimum of $\mathcal{L}_k^N$. Instead, **(H)** only needs T_k to be an approximate local minimum of $\mathcal{L}_k^N$ and **(I)** implies that T_k converges in probability towards a *strict local* minimizer $T_k^\star$ of $\mathcal{L}_k$ as $N \to \infty$.

Before stating the CLT result, we need to introduce the asymptotic incremental variance $\mathbb{V}_k^{\text{inc}}[f]$ at iteration k. To this end, consider the set of limiting re-sampling times $\mathcal{K}_{\text{opt}} := \{k_0, ...k_P\} \subset \{0, ..., K\}$ defined recursively by $k_{p+1} := \inf\{k_p < k : \text{nESS}_k \leq A\}$ and $k_{P+1} := K + 1$ where $\text{ESS}_k^N / N \overset{N\to\infty}{\to} \text{nESS}_k$ with

$$\text{nESS}_k = \frac{\pi_{k_p}\left[\mathbb{E}\left[w_k^\star \middle| X_{k_p}\right]\right]^2}{\pi_{k_p}\left[\mathbb{E}\left[(w_k^\star)^2 \middle| X_{k_p}\right]\right]},$$

the expectation being w.r.t. to $X_s \sim K_s(T_s^\star(X_{s-1}), \cdot)$ for $k_p + 1 \leq s \leq k$, while $X_{k_p} \sim \pi_{k_p}$ and $w_k^\star = \prod_{s=k_p+1}^k G_{s,T_s^\star}(X_{s-1})$ is the product of the incremental IS weights using the locally optimal NFs $T_s^\star$. The variance $\mathbb{V}_k^{\text{inc}}[f]$ at time k is given by:

$$\mathbb{V}_k^{\text{inc}}[f] = \begin{cases} Z_k^2 \text{Var}_{\pi_k}[f], & k \in \mathcal{K}, \\ Z_{k_p}^2 \pi_{k_p}\left[\mathbb{E}\left[(w_k^\star)^2 \mathcal{G}_k[f] \middle| X_{k_p}\right]\right], & k_p < k < k_{p+1}, \end{cases}$$

with $\mathcal{G}_k[f] := K_k\left[f^2\right](T_k^\star(X_{k-1})) - K_k[f]^2(T_k^\star(X_{k-1}))$.

Theorem 2 (Central limit theorem)**.** *Let f be a real valued function s.t., for some $C > 0$, $f(x) \leq C(1 + \|x\|^2)$ and*

$$\|f(x) - f(x')\| \leq C\left(1 + \|x\|^3 + \|x'\|^3\right)\|x - x'\|.$$

Then, under Assumptions (A) to (I) and for $0 \leq k \leq K$:

$$(CLT_k): \quad \begin{cases} \sqrt{N}\left(\pi_k^N[f] - \pi_k[f]\right) & \xrightarrow{\mathcal{D}} \mathcal{N}(0, \mathbb{V}_k^\pi[f]), \\ \sqrt{N}\left(Z_k^N - Z_k\right) & \xrightarrow{\mathcal{D}} \mathcal{N}(0, \mathbb{V}_k^\gamma[1]). \end{cases}$$

$\mathbb{V}_k^\gamma[f]$ and $\mathbb{V}_k^\pi[f]$ are defined recursively with $\mathbb{V}_0^\gamma[f] = \text{Var}_{\pi_0}[f]$ and

$$\mathbb{V}_k^\gamma[f] = \mathbb{V}_k^{\text{inc}}[f] + \mathbb{V}_{k-1}^\gamma\left[Q_{k,T_k^\star}[f]\right],$$
$$\mathbb{V}_k^\pi[f] = Z_k^{-2}\mathbb{V}_k^\gamma[f - \pi_k[f]],$$

where $Q_{k,T}(x, \text{d}y) := G_{k,T}(x)K_k(T(x), \text{d}y)$.

The asymptotic variances $\mathbb{V}_k^\gamma$ and $\mathbb{V}_k^\pi$ depend only on the maps T_k^* and not on the local variations of the family $\mathcal{T}$ around T_k^*. This is a consequence of the particular form of the IS weights which provide an exact correction regardless of the NF selected as summarized by the following identity:

$$\pi_k[f] = \frac{\pi_{k-1}[Q_{k,T}[f]]}{\pi_{k-1}[G_{k,T}]}, \qquad \forall T \in \mathcal{T}.$$

In the ideal case when $T_k^\star$ are exact transport maps from π_{k-1} to π_k, the ESS resampling criterion ESS_k^N/N is always equal to 1 and thus resampling is never triggered. Moreover, a direct computation shows that the asymptotic variance $\mathbb{V}_k^\pi[f]$ is exactly equal to $\mathrm{Var}_{\pi_k}[f]$. This illustrates the benefit of introducing NFs to improve SMC. A proof is provided in Appendix C.4 along with a similar result (Theorem 5) for Algorithm 2.

5. Continuous-time scaling limit

We consider the setting where π_k arise from the time-discretization of a continuous-time path $(\Pi_t)_{[0,1]}$ of densities connecting π_0 to π; i.e. π_k is of the form $\pi_k = \Pi_{t_k}$ with $t_k = k\lambda$ and $\lambda = \frac{1}{K}$. We write $V_t(x)$ and Z_t to denote the potential and unknown normalizing constant of Π_t and $\Gamma_t(x) = \exp(-V_t(x))$. We are here interested in identifying the "population" behavior of AFT (i.e. $N \to \infty$) as $\lambda \to 0$ when ULA kernels are used and no resampling is performed as in AIS. To simplify the analysis, we further consider in this Section the ideal situation where T_k is an exact minimizer of the population loss $\mathcal{L}_k$. Rigorous proofs of the results discussed here can be found in Appendix E.

5.1. Settings

Without resampling, the population version of AFT behaves as a sequential IS algorithm as defined in Section 2.1 where it is possible to *collapse* the transport step and mutation step into one single Markov kernel $M_k(x, x') = K_k(T_k(x), x')$. Similarly we can collapse the corresponding backward kernels and the resulting extended target distributions $\bar{\pi}_k$ are still given by (5) with modified IS weights

$$w_k(x_{0:k}) = \underbrace{\prod_{l=1}^{k} \frac{\gamma_l(x_l)}{\gamma_l K_l(x_l)}}_{r_k(x_{1:k})} \prod_{l=1}^{k} G_{l,T_l}(x_{l-1}), \qquad (11)$$

where $r_k(x_{1:k}) = 1$ for π_l-invariant MCMC kernels K_l as used in Algorithm 1; see Appendix B.2 for a derivation. To ensure that the laws $\bar{\eta}_k$ and $\bar{\pi}_k$ of the Markov chain $X_{0:k}$ converge to some continuous-time limits, K_k are chosen to be ULA kernels[3]; i.e. $K_k(x, x')$ is a Gaussian density in x' with mean $x - \lambda V_k(x)$ and covariance $2\lambda I$. In this case, $\gamma_k K_k(x) = \int \gamma_k(y) K_k(y, x)\,\mathrm{d}y$ is intractable and so is $r_k(x_{1:k})$. This is not an issue as we are only interested here in identifying the theoretical scaling limit. To ensure $\bar{\eta}_k$ and $\bar{\pi}_k$ admit a limit, we also consider NFs of the form:

$$T(x) = x + \lambda A_\theta(x),$$

where $(\theta, x) \mapsto A_\theta(x)$ is from $\Theta \times \mathcal{X}$ to $\mathcal{X}$ and Θ is a compact parameter space. The continuous-time analogues of NFs sequences $(T_k)_{k\in\{1,\dots,K\}}$ are represented by a set $\mathcal{A}$ of time-dependent controls of the form $\alpha_t(x) = A_{\theta_t}(x)$, where $t \mapsto \theta_t$ is a 1-Lipschitz trajectory in Θ. To any control α corresponds an NFs sequence $(T_k)_{k\in\{1,\dots,K\}}$ defined by $T_k(x) = x + \lambda \alpha_{t_k}(x)$.

5.2. Continuous-time limits

Limiting forward process. Using a similar approach to (Dalalyan, 2017), the Markov chain $X_{0:K}$ under $\bar{\eta}_K$ converges towards a stochastic process $X_{[0,1]}$ defined by the following Stochastic Differential Equation (SDE)

$$\mathrm{d}X_t = (\alpha_t(X_t) - \nabla V_t(X_t))\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}B_t, \qquad (12)$$

where $X_0 \sim \pi_0$ and $(B_t)_{t\geq 0}$ is a standard Brownian motion. We denote by $\bar{\Lambda}_t^\alpha$ the joint distribution of this process up to time t and by Λ_t^α its marginal at time t.

Limiting weights. The weight $w_K(X_{0:K})$ in (11) is such that $r_K(X_{1:K}) \to 1$ as the invariant distribution of the ULA kernel K_k converges to π_k when $\lambda \to 0$ while the logarithm of the product of $G_{l,T_l}(X_{l-1})$ is a Riemann sum whose limiting value is the following integral:

$$\sum_{l=1}^{K} \log(G_{l,T_l}(X_{l-1})) \xrightarrow[\lambda\to 0]{} \int_0^1 g_s^\alpha(X_s)\,\mathrm{d}s,$$

with $X_{[0,1]}$ defined in (12) and $g_t^\alpha(x)$ being the dominating term in the Taylor expansion of $\log(G_{l,T_l}(x))$ w.r.t. time:

$$g_t^\alpha(x) = \nabla \cdot \alpha_t(x) - \nabla_x V_t(x)^\top \alpha_t(x) - \partial_t V_t(x).$$

The limit of IS weights $w_k(X_{0:k})$ is thus identified as

$$w_t^\alpha(X_{[0,t]}) = \exp\left(\int_0^t g_s^\alpha(X_s)\,\mathrm{d}s\right).$$

In the context of *non-equilibrium dynamics*, $g_t^\alpha(x)$ is known as *instantaneous work* (Rousset and Stoltz, 2006) and is constant in the ideal case where $\Pi_t = \Lambda_t^\alpha$.

Limiting objective. To identify a non-trivial limiting loss, we consider the following aggregation of all $\mathcal{L}_k(T_k)$

$$\mathcal{L}_\lambda^{tot}(\alpha) := \lambda^{-1} \sum_{k=1}^{K} \mathcal{L}_k(T_k). \qquad (13)$$

[3] The random walk MH algorithm also admits a Langevin diffusion as scaling limit when $\lambda \to 0$ (Gelfand and Mitter, 1991; Choi, 2019) but the technical analysis is much more involved.

The next result shows that (13) converges towards a non-trivial loss $\mathcal{M}(\alpha)$ as $\lambda \to 0$ under three assumptions stated in Appendix E.2: **(a)** and **(b)** on the smoothness of $V_t(x)$ and $A_\theta(x)$ and **(c)** on the moments of Π_t.

Proposition 1. *Under Assumptions **(a)** to **(c)**, for λ small enough, it holds that for all $\alpha \in \mathcal{A}$*

$$\left|\mathcal{L}_\lambda^{tot}(\alpha) - \mathcal{M}(\alpha)\right| \leq \lambda C,$$

where C is independent of λ and

$$\mathcal{M}(\alpha) = \frac{1}{2}\int_0^1 \left(\Pi_t\left[(g_t^\alpha)^2\right] - \Pi_t[g_t^\alpha]^2\right) \mathrm{d}t. \qquad (14)$$

The optimal NFs $(T_k)_{1:K}$ are thus expected to converge towards some $\alpha^\star$ minimizing $\mathcal{M}(\alpha)$ over $\mathcal{A}$ as made precise in Proposition 29 of Appendix E.6. Moreover when the class of NFs is expressive, i.e. $\mathcal{A}$ is rich enough, then $\mathcal{M}(\alpha^\star) = 0$ and thus g_t^α are constant and $\alpha^\star$ satisfies the Partial Differential Equation (PDE)

$$0 = \nabla \cdot \alpha_t^\star(x) - \nabla_x V_t(x)^\top \alpha_t^\star(x) - \partial_t V_t(x) + \Pi_t[\partial_t V_t].$$

This PDE has appeared, among others, in Lelièvre et al. (2010, pp. 273–275) and (Vaikuntanathan and Jarzynski, 2008; Reich, 2011; Heng et al., 2021). Its solution defines a deterministic flow $\alpha_t^\star$ that transports mass along the path $(\Pi_t^\alpha)_{[0,1]}$; i.e. if X_t is a solution to an ODE of the form $\dot{X}_t = \alpha_t^\star(X_t)$ with initial values $X_0 \sim \Pi_0$, then $X_t \sim \Pi_t$.

Feynman–Kac measure. Given a control α, we consider the Feynman–Kac measure $\overline{\Pi}_t$ defined for any bounded continuous functional f of the process $X_{[0,t]}$ in (12)

$$\overline{\Pi}_t^\alpha[f] = \frac{\overline{\Lambda}_t^\alpha[w_t^\alpha f]}{\overline{\Lambda}_t^\alpha[w_t^\alpha]}. \qquad (15)$$

By a similar argument as in (Rousset and Stoltz, 2006), we show in Proposition 22 of Appendix E.3 that $\overline{\Pi}_t^\alpha$ admits Π_t as a marginal at time t regardless of the choice of α. Using the optimal control $\alpha^\star$ in (12) and (15) gives rise to $\overline{\Lambda}^\star$ and $\overline{\Pi}_t^\star$ which are equal when $\mathcal{M}(\alpha^\star) = 0$. Next, we show that $\overline{\Pi}_t^\star$ is the scaling limit of $\overline{\pi}_k$.

5.3. Convergence to the continuous-time limit

As the measures $\overline{\pi}_k$ and $\overline{\Pi}_t^\star$ are defined on different spaces, we construct a sequence of interpolating measures $\overline{\Pi}_t^\lambda$ defined over the same space as $\overline{\Pi}_t^\star$ and whose marginal at the joint times $\{t_0, ..., t_K\}$ is exactly equal to $\overline{\pi}_k$; see Appendix E.1 for details. Theorem 3 provides a convergence rate for the interpolating measures $\overline{\Pi}_t^\lambda$ towards $\overline{\Pi}_t^\star$ as $\lambda \to 0$, thus establishing $\overline{\Pi}_t^\star$ as the scaling limit of $\overline{\pi}_k$; see Appendix E.6 for the proof.

Theorem 3. *Under Assumptions **(a)** to **(g)**, then for λ small enough there exists a finite C such that for any $t \in [0,1]$:*

$$\mathrm{KL}(\overline{\Pi}_t^\star || \overline{\Pi}_t^\lambda) \leq C\sqrt{\lambda}.$$

This result relies on Assumptions **(d)** to **(g)** in addition to Assumptions **(a)** to **(c)** which are also stated in Appendix E.2. **(d)** strengthens assumption **(c)** on the moments of Π_t. **(e)** guarantees the existence of a solution $\alpha^\star$ in $\mathcal{A}$ minimizing $\mathcal{M}$ and controls the local behavior of $\mathcal{M}$ near $\alpha^\star$. **(f)** guarantees the existence of solutions α^λ in $\mathcal{A}$ minimizing $\mathcal{L}_\lambda^{tot}(\alpha)$ for any $\lambda = \frac{1}{K}$. Finally, **(g)** ensures the optimal control $\alpha^\star$ induces bounded IS weights.

6. Applications

In this section we detail the practical implementation of AFT and empirically investigate performance against relevant baselines.

As discussed in Section 3.3, we use three sets of particles-'train, test and validation' which improves robustness, avoids overfitting the flow to the particles and gives unbiased estimates of Z when using the test set. We initialize our flows to the identity for the optimization at each time step. Algorithm 2, in the supplement gives a summary.

We concentrate our empirical value evaluation on the learnt flow, which is equivalent to using the test set particles. The learnt flow is of interest in deploying an efficient sampler on large scale distributed parallel compute resources. It is also of interest for inclusion as a subroutine in a larger system. Since modern hardware enables us to do large computations in parallel, the computation is dominated by algorithmic steps that are necessarily done serially, particularly repeat applications of the Markov kernel (Lee et al., 2010).

As our primary, strong, baseline for AFT, we use a standard instance of SMC samplers (Del Moral et al., 2006; Zhou et al., 2016) which corresponds to AIS with adaptive resampling and is also known as population annealing in physics (Hukushima and Iba, 2003; Barash et al., 2017). As observed many times in the literature and in our experiments, SMC estimates are of lower variance than AIS estimates. This SMC baseline is closely related to AFT since it corresponds to using AFT with an identity transformation $T_k(x) = x$ instead of a learnt flow.

We largely use the number of transitions K as a proxy for compute time. This is valid when the cost of evaluating the flow is modest relative to that of the other algorithmic steps, as it is for the trained flows in all non-trivial cases we consider. We only consider flows of no more than a few layers per transition, but deeper flows could start to form an appreciable part of the serial computation. In some cases, we use variational inference (VI) as a measure of behaviour without MCMC. In this case, evaluation time is not comparable and faster. Since we concentrate on trained flows, we do not evaluate training time in the benchmarks considered, though fast training of AFT could be of interest in further work. Both SMC and AFT use the same Markov kernels K_k, using HMC except where otherwise stated. We tune the step size to have a reasonable acceptance probability based on preliminary runs of SMC using a modest K. Then for larger K experiments, we linearly interpolate the step sizes chosen on the preliminary runs. We always use a linearly spaced geometric schedule and the initial distribution is always a multivariate standard normal. We repeat experiments 100 times. Further experimental details may be found in Appendix G. We plan to make the code available within `https://github.com/deepmind`.

6.1. Illustrative example

We start with an easily visualized two dimensional target density as shown in Figure 1. All sensible methods should work in such a low dimensional case but it can still be informative. We investigate two families of flows based on rational quadratic splines (Durkan et al., 2019). The first (termed AFTmf for mean field) operates on the two dimensions separately. The second family (denoted AFT in Figure 1) adds dependence to the splines using inverse autoregressive flows (Kingma et al., 2016). Figure 1 shows weighted samples from AFT as we anneal from a standard normal distribution. Figure 2 (a) shows that AFT reduces the variance of the normalizing constant estimator relative to SMC. Conversely, we see that AFTmf actually *increases* the variance relative to SMC for small numbers of transitions. Since the factorized approximation cannot model the dependence of variables the optimum of the KL underestimates the variance of the target. Later, in Sections 6.3 and 6.4, we discuss examples where even a simple NF leads to an improvement for a modest number of transitions.

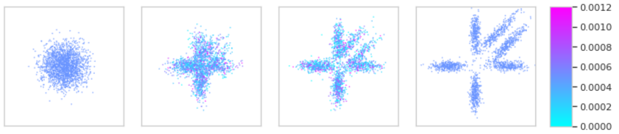

Figure 1: Weighted samples for a 2-D target density with AFT. The colours show the normalized weights which are clipped at the 95th percentile for clarity. The final samples are visually indistinguishable from the target.

6.2. Funnel distribution

We next evaluate the performance of the method on Neal's ten-dimensional 'funnel' distribution (Neal, 2003):

$$x_0 \sim \mathcal{N}(0, \sigma_f^2), \quad x_{1:9}|x_0 \quad \sim \mathcal{N}(\mathbf{0}, \exp(x_0)\mathbf{I}).$$

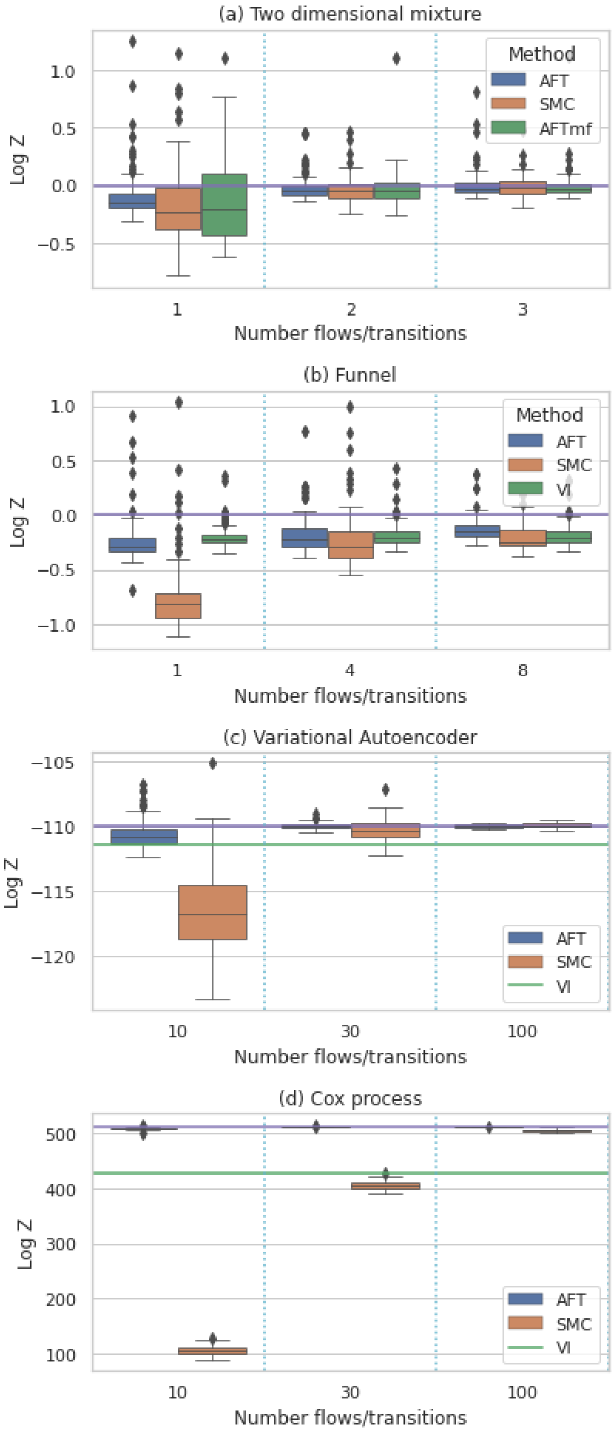

Figure 2: Results from the four different examples. Cyan lines denote gold standard values of the log normalizing constant. In (c) and (d) green horizontal lines denote the median value for an importance sampling estimate based on variational inference. Note that in (d) the small AFT error bars can make it difficult to see - it can be found next to the gold standard value in each case.

Here, $\sigma_f^2 = 9$. Many MCMC methods find this example challenging because there is a variety of length scales depending on the value of x_0 and because marginally $x_{1:9}$ has heavy tails. We use here slice sampling instead of HMC for the Markov kernels as recommended in (Neal, 2003). For each flow we use an affine inverse autoregressive flow (Kingma et al., 2016). In this example, we also compare against VI (Rezende et al., 2014) which uses the same number of flows. We then apply a simple importance correction to the VI samples to give an unbiased estimate of the normalizing constant. Figure 2 (b) shows the results. We see that for small number of flows/transitions VI performs best, followed by AFT. However, VI shows little further improvement with additional flows and in this regime AFT, SMC and VI perform similarly.

6.3. Variational Autoencoder latent space

For our next example, we trained a variational autoencoder (Kingma and Welling, 2014; Rezende et al., 2014) with convolution on the binarized MNIST dataset (Salakhutdinov and Murray, 2008) and a normal encoder distribution with diagonal covariance. Using the fixed, trained, generative decoder network we investigated the quality of normalizing constant estimation which in this case corresponds to the likelihood of a data point with the distribution over the 30 latent variables marginalized out (Wu et al., 2017).

Using long run SMC on the 10000 point test set we estimate that the hold out log-likelihood per data point for the network is -86.3. For each data point we also found the optimal variational normal approximation with diagonal covariance rather than using the amortized variational approximation. Using this optimal normal approximation we investigated its variance when used as an importance proposal for the likelihood. We estimate the mean absolute error for the estimator across the test set was 0.6 nats per data point which indicates that the VI is often performing well. There was a tail of digits where VI performed relatively worse. Since these ‘difficult’ digits constituted a more challenging inference problem, we used one of these, with a VI/SMC error of 1.5 nats, to comparatively benchmark AFT in the detailed manner used in our other examples.

For the AFT flow we used an affine transformation with diagonal linear transformation matrix. The baseline VI approximation can be thought of the pushforward of a standard normal through this ‘diagonal affine’ flow. Note that since diagonal affine transformations are closed under composition there would obtain no additional expressiveness in the baseline VI approximation from adding more of them.

Figure 2 (c) shows the results for this example. Both AFT and SMC reduce in variance as the number of temperatures increases and exceed the performance of the variational baseline. AFT has a notably lower variance than SMC for 10 and 30 temperatures- which shows the incorporation of the flows is beneficial in this case. Results for other difficult digits are shown in the appendix where the qualitative trend is similar.

6.4. Log Gaussian Cox process

We evaluate here the performance of AFT for estimating the normalizing constant of a log Gaussian Cox process applied to modelling the positions of pine saplings in Finland (Møller et al., 1998). We consider points on a discretized $d = M \times M = 1600$ grid. This results in the target density

$$\gamma(x) = \mathcal{N}(x; \mu, K) \prod_{i \in [1:M]^2} \exp(x_i y_i - a \exp(x_i)).$$

This challenging high-dimensional problem is a commonly used benchmark in the SMC literature (Heng et al., 2020; Buchholz et al., 2021). The mean and covariance function match those estimated by (Møller et al., 1998) and are detailed in the Appendix. The supplement also discusses the effect of pre-conditioners on the mixing of the Markov kernel. For the NF we again used the diagonal affine transformation. The approximating family is the push forward of the previous target distribution and thus even a simple flow can result in a good approximation. It is also fast to evaluate. Figure 2 (d) shows that the baseline VI approximation is unable to capture the posterior correlation and that AFT gives significantly more accurate results than SMC for a given number of transitions. As such, the Markov kernel and flow complement each other in this case.

7. Conclusion

We proposed Annealed Flow Transport which combines SMC samplers and normalizing flows. We studied its asymptotic behavior and showed the benefit of introducing learned flows to reduce the asymptotic variance. We identified the scaling limit of AFT as a controlled Feynman–Kac measure whose optimal control solved a flow transport problem in an idealized setting. Empirically we found multiple cases where trained AFT gave lower variance estimates than SMC for the same number of transitions, showing that we can combine the advantages of both SMC and normalizing flows. We believe AFT will be particularly useful in scenarios where it is both difficult to design fast mixing MCMC kernels and very good flows so that neither SMC nor VI provide low variance estimates.

8. Acknowledgements

The authors would like to thank Danilo Rezende, Arthur Gretton and Taylan Cemgil.

References

Akyildiz, Ö. D. and Míguez, J. (2020). Nudging the particle filter. *Statistics and Computing*, 30(2):305–330.

Barash, L. Y., Weigel, M., Borovskỳ, M., Janke, W., and Shchur, L. N. (2017). GPU accelerated population annealing algorithm. *Computer Physics Communications*, 220:341–350.

Beskos, A., Jasra, A., Kantas, N., and Thiery, A. (2016). On the convergence of adaptive sequential Monte Carlo methods. *The Annals of Applied Probability*, 26(2):1111–1146.

Beskos, A., Pinski, F., Sanz-Serna, J., and Stuart, A. (2011). Hybrid Monte Carlo on Hilbert spaces. *Stochastic Processes and their Applications*, 121(10):2201 – 2230.

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. (2018). JAX: composable transformations of Python+NumPy programs.

Buchholz, A., Chopin, N., and Jacob, P. E. (2021). Adaptive tuning of Hamiltonian Monte Carlo within sequential Monte Carlo. *Bayesian Analysis to appear - arXiv preprint arXiv:1808.07730*.

Caterini, A. L., Doucet, A., and Sejdinovic, D. (2018). Hamiltonian variational auto-encoder. In *Advances in Neural Information Processing Systems*, pages 8167–8177.

Choi, M. C. (2019). Universality of the Langevin diffusion as scaling limit of a family of Metropolis–Hastings processes i: fixed dimension. *arXiv preprint arXiv:1907.10318*.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika*, 89(3):539–552.

Chopin, N. (2004). Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics*, 32(6):2385–2411.

Crooks, G. E. (1998). Nonequilibrium measurements of free energy differences for microscopically reversible Markovian systems. *Journal of Statistical Physics*, 90(5-6):1481–1487.

Dai, C., Heng, J., Jacob, P. E., and Whiteley, N. (2020). An invitation to sequential Monte Carlo samplers. *arXiv preprint arXiv:2007.11936*.

Dalalyan, A. S. (2017). Theoretical guarantees for approximate sampling from smooth and log-concave densities. *Journal of the Royal Statistical Society: Series* B, 3(79):651–676.

Del Moral, P. (2004). *Feynman-Kac Formulae: Genealogical and Interacting Particle Approximations*. Springer.

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.

Del Moral, P., Doucet, A., and Jasra, A. (2012a). An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing*, 22(5):1009–1020.

Del Moral, P., Doucet, A., and Jasra, A. (2012b). On adaptive resampling strategies for sequential Monte Carlo methods. *Bernoulli*, 18(1):252–278.

Dillon, J. V., Langmore, I., Tran, D., Brevdo, E., Vasudevan, S., Moore, D., Patton, B., Alemi, A., Hoffman, M., and Saurous, R. A. (2017). TensorFlow Distributions. *arXiv preprint arXiv:1711.10604*.

Domke, J. and Sheldon, D. R. (2018). Importance weighting and variational inference. In *Advances in Neural Information Processing Systems*, pages 4470–4479.

Douc, R. and Moulines, E. (2008). Limit theorems for weighted samples with applications to sequential Monte Carlo methods. *The Annals of Statistics*, 36(5):2344–2376.

Dudley, R. M. (2018). *Real analysis and Probability*. CRC Press.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. (2019). Neural spline flows. In *Advances in Neural Information Processing Systems*.

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

Everitt, R. G., Culliford, R., Medina-Aguayo, F., and Wilson, D. J. (2020). Sequential Monte Carlo with transformations. *Statistics and Computing*, 30(3):663–676.

Gao, C., Isaacson, J., and Krause, C. (2020). i-flow: High-dimensional Integration and Sampling with normalizing flows. *arXiv preprint arXiv:2001.05486*.

Gelfand, S. B. and Mitter, S. K. (1991). Weak convergence of Markov chain sampling methods and annealing algorithms to diffusions. *Journal of Optimization Theory and Applications*, 68(3):483–498.

Germain, M., Gregor, K., Murray, I., and Larochelle, H. (2015). Made: Masked autoencoder for distribution estimation. In Bach, F. and Blei, D., editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 881–889, Lille, France. PMLR.

Gilks, W. R. and Berzuini, C. (2001). Following a moving target - Monte Carlo inference for dynamic Bayesian models. *Journal of the Royal Statistical Society: Series B*, 63(1):127–146.

Goyal, A. G. A. P., Ke, N. R., Ganguli, S., and Bengio, Y. (2017). Variational walkback: Learning a transition operator as a stochastic recurrent net. In *Advances in Neural Information Processing Systems*, pages 4392–4402.

Guarniero, P., Johansen, A. M., and Lee, A. (2017). The iterated auxiliary particle filter. *Journal of the American Statistical Association*, 112(520):1636–1647.

Han, J. and Liu, Q. (2017). Stein variational adaptive importance sampling. *Uncertainty in Artificial Intelligence*.

Heng, J., Bishop, A. N., Deligiannidis, G., and Doucet, A. (2020). Controlled sequential Monte Carlo. *The Annals of Statistics*, 48(5):2904–2929.

Heng, J., Doucet, A., and Pokern, Y. (2021). Gibbs flow for approximate transport with applications to Bayesian computation. *Journal of the Royal Statistical Society Series* B, 83(1):156–187.

Hennigan, T., Cai, T., Norman, T., and Babuschkin, I. (2020). Haiku: Sonnet for JAX.

Hessel, M., Budden, D., Viola, F., Rosca, M., Sezener, E., and Hennigan, T. (2020). Optax: composable gradient transformation and optimisation, in jax.

Hoffman, M. D. (2017). Learning deep latent Gaussian models with Markov chain Monte Carlo. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1510–1519. PMLR.

Huang, C.-W., Tan, S., Lacoste, A., and Courville, A. C. (2018). Improving explorability in variational inference with annealed variational objectives. In *Advances in Neural Information Processing Systems*, pages 9701–9711.

Huang, S., Makhzani, A., Cao, Y., and Grosse, R. (2020). Evaluating lossy compression rates of deep generative models. In *International Conference on Machine Learning*, pages 4444–4454. PMLR.

Hukushima, K. and Iba, Y. (2003). Population annealing and its application to a spin glass. In *AIP Conference Proceedings*, volume 690, pages 200–206. American Institute of Physics.

Jarzynski, C. (1997). Nonequilibrium equality for free energy differences. *Physical Review Letters*, 78(14):2690–2963.

Jasra, A., Stephens, D. A., Doucet, A., and Tsagaris, T. (2011). Inference for Lévy-driven stochastic volatility models via adaptive sequential Monte Carlo. *Scandinavian Journal of Statistics*, 38(1):1–22.

Kappen, H. J. and Ruiz, H. C. (2016). Adaptive importance sampling for control and inference. *Journal of Statistical Physics*, 162(5):1244–1266.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Kingma, D. P. and Welling, M. (2014). Auto-encoding variational Bayes. *ICLR*.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, 5(1):1–25.

Künsch, H. R. (2005). Recursive Monte Carlo filters: algorithms and theoretical analysis. *The Annals of Statistics*, 33(5):1983–2021.

Le, T. A., Igl, M., Rainforth, T., Jin, T., and Wood, F. (2018). Auto-encoding sequential Monte Carlo. In *ICLR*.

Lee, A., Yau, C., Giles, M. B., Doucet, A., and Holmes, C. C. (2010). On the utility of graphics cards to perform massively parallel simulation of advanced monte carlo methods. *Journal of Computational and Graphical Statistics*, 19(4):769–789.

Lei Ba, J., Kiros, J. R., and Hinton, G. E. (2016). Layer Normalization. *arXiv e-prints*.

Lelièvre, T., Rousset, M., and Stoltz, G. (2010). *Free Energy Computations: A Mathematical Perspective*. World Scientific.

Li, Q. and Chen, Y. (2019). Rate distortion via deep learning. *IEEE Transactions on Communications*, 68(1):456–465.

Liu, C., Zhuo, J., Cheng, P., Zhang, R., and Zhu, J. (2019). Understanding and accelerating particle-based variational inference. In *International Conference on Machine Learning*, pages 4082–4092.

Liu, J. S. and Chen, R. (1995). Blind deconvolution via sequential imputations. *Journal of the American Statistical Association*, 90(430):567–576.

Liu, Q. and Wang, D. (2016). Stein variational gradient descent: a general purpose Bayesian inference algorithm. In *Advances in Neural Information Processing Systems*.

Llorente, F., Martino, L., Delgado, D., and Lopez-Santiago, J. (2020). Marginal likelihood computation for model selection and hypothesis testing: an extensive review. *arXiv preprint arXiv:2005.08334*.

MacEachern, S. N., Clyde, M., and Liu, J. S. (1999). Sequential importance sampling for nonparametric Bayes models: The next generation. *Canadian Journal of Statistics*, 27(2):251–267.

Maddison, C. J., Lawson, J., Tucker, G., Heess, N., Norouzi, M., Mnih, A., Doucet, A., and Teh, Y. (2017). Filtering variational objectives. In *Advances in Neural Information Processing Systems*, pages 6573–6583.

Marzouk, Y., Moselhy, T., Parno, M., and Spantini, A. (2016). Sampling via measure transport: An introduction. *Handbook of Uncertainty Quantification*, pages 1–41.

Mnih, A. and Rezende, D. (2016). Variational inference for Monte Carlo objectives. In *International Conference on Machine Learning*, pages 2188–2196. PMLR.

Møller, J., Syversveen, A. R., and Waagepetersen, R. P. (1998). Log Gaussian Cox processes. *Scandinavian Journal of Statistics*, 25(3):451–482.

Naesseth, C. A., Linderman, S. W., Ranganath, R., and Blei, D. M. (2018). Variational sequential Monte Carlo. In *AISTATS*.

Neal, R. (2011). MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo*.

Neal, R. M. (2001). Annealed importance sampling. *Statistics and Computing*, 11(2):125–139.

Neal, R. M. (2003). Slice sampling. *The Annals of Statistics*, 31(3):705–767.

Nicoli, K. A., Nakajima, S., Strodthoff, N., Samek, W., Müller, K.-R., and Kessel, P. (2020). Asymptotically unbiased estimation of physical observables with neural samplers. *Physical Review E*, 101(2):023304.

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

Olmez, S. Y., Taghvaei, A., and Mehta, P. G. (2020). Deep fpf: Gain function approximation in high-dimensional setting. In *59th IEEE Conference on Decision and Control (CDC)*, pages 4790–4795. IEEE.

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*.

Reich, S. (2011). A dynamical systems framework for intermittent data assimilation. *BIT Numerical Mathematics*, 51(1):235–249.

Reich, S. and Weissmann, S. (2021). Fokker–Planck particle systems for Bayesian inference: Computational approaches. *SIAM/ASA Journal on Uncertainty Quantification*, 9(2):446–482.

Rezende, D. J. and Mohamed, S. (2015). Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning - Volume 37*, ICML'15, pages 1530–1538. JMLR.org.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, pages 1278–1286.

Richard, J.-F. and Zhang, W. (2007). Efficient high-dimensional importance sampling. *Journal of Econometrics*, 141(2):1385–1411.

Rousset, M. and Stoltz, G. (2006). Equilibrium sampling from nonequilibrium dynamics. *Journal of Statistical Physics*, 123(6):1251–1272.

Salakhutdinov, R. and Murray, I. (2008). On the quantitative analysis of Deep Belief Networks. In *Proceedings of the 25th Annual International Conference on Machine Learning (ICML 2008)*, pages 872–879.

Salimans, T., Kingma, D., and Welling, M. (2015). Markov chain Monte Carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pages 1218–1226.

Schäfer, C. and Chopin, N. (2013). Sequential Monte Carlo on large binary sampling spaces. *Statistics and Computing*, 23(2):163–184.

Sen, B. (2018). A gentle introduction to empirical process theory and applications. *Lecture Notes, Columbia University*.

Taghvaei, A., Mehta, P. G., and Meyn, S. P. (2020). Diffusion map-based algorithm for gain function approximation in the feedback particle filter. *SIAM/ASA Journal on Uncertainty Quantification*, 8(3):1090–1117.

Thin, A., Kotelevskii, N., Durmus, A., Panov, M., Moulines, E., and Doucet, A. (2021). Monte Carlo variational auto-encoders. *International Conference on Machine Learning*.

Turner, R. E. and Sahani, M. (2011). Two problems with variational expectation maximisation for time-series models. In Barber, D., Cemgil, T., and Chiappa, S., editors, *Bayesian Time Series Models*, chapter 5, pages 109–130. Cambridge University Press.

Vaikuntanathan, S. and Jarzynski, C. (2008). Escorted free energy simulations: Improving convergence by reducing dissipation. *Physical Review Letters*, 100(19):190601.

Vaikuntanathan, S. and Jarzynski, C. (2011). Escorted free energy simulations. *The Journal of Chemical Physics*, 134(5):054107.

Van der Vaart, A. W. (2000). *Asymptotic Statistics*. Cambridge University Press.

Wang, Y. and Li, W. (2019). Accelerated information gradient flow. *arXiv preprint arXiv:1909.02102*.

Wirnsberger, P., Ballard, A. J., Papamakarios, G., Abercrombie, S., Racanière, S., Pritzel, A., Rezende, D., and Blundell, C. (2020). Targeted free energy estimation via learned mappings. *The Journal of Chemical Physics*, 153(14):144112.

Wu, H., Köhler, J., and Noé, F. (2020). Stochastic normalizing flows. In *Advances in Neural Information Processing Systems*.

Wu, Y., Burda, Y., Salakhutdinov, R., and Grosse, R. B. (2017). On the quantitative analysis of decoder-based generative models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Zhou, Y., Johansen, A. M., and Aston, J. A. (2016). Toward automatic model comparison: An adaptive sequential Monte Carlo approach. *Journal of Computational and Graphical Statistics*, 25(3):701–726.

Zhu, M., Liu, C., and Zhu, J. (2020). Variance reduction and quasi-Newton for particle-based variational inference. In *International Conference on Machine Learning*, pages 11576–11587.