Annealed Flow Transport Monte Carlo

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

[*]Equal Contribution

[1]Gatsby Computational Neuroscience Unit, University College London

[2]DeepMind

Overview

Problem

- Goal 1: Sampling from a target density π known up to a normalizing constant Z.
- Goal 2: Estimating the normalizing constant Z.

Applications

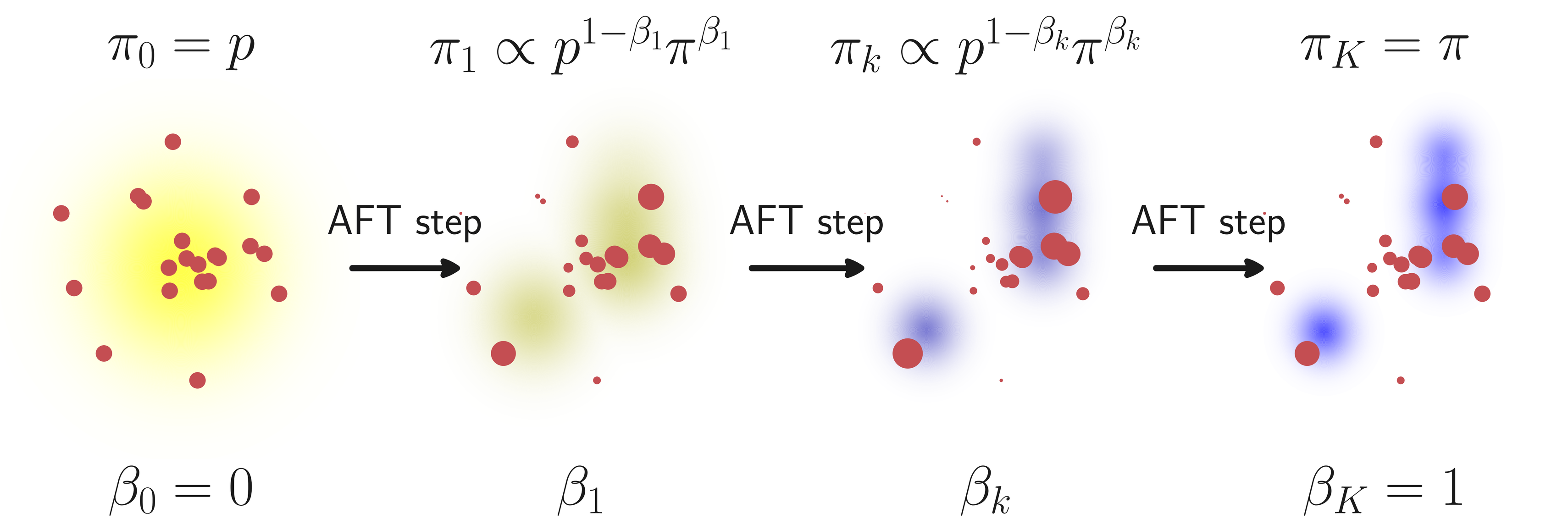
- Bayesian statistics, Compression, Statistical physics, Chemistry, etc...

Challenges

- Curse of dimensionality.
- Multimodality.

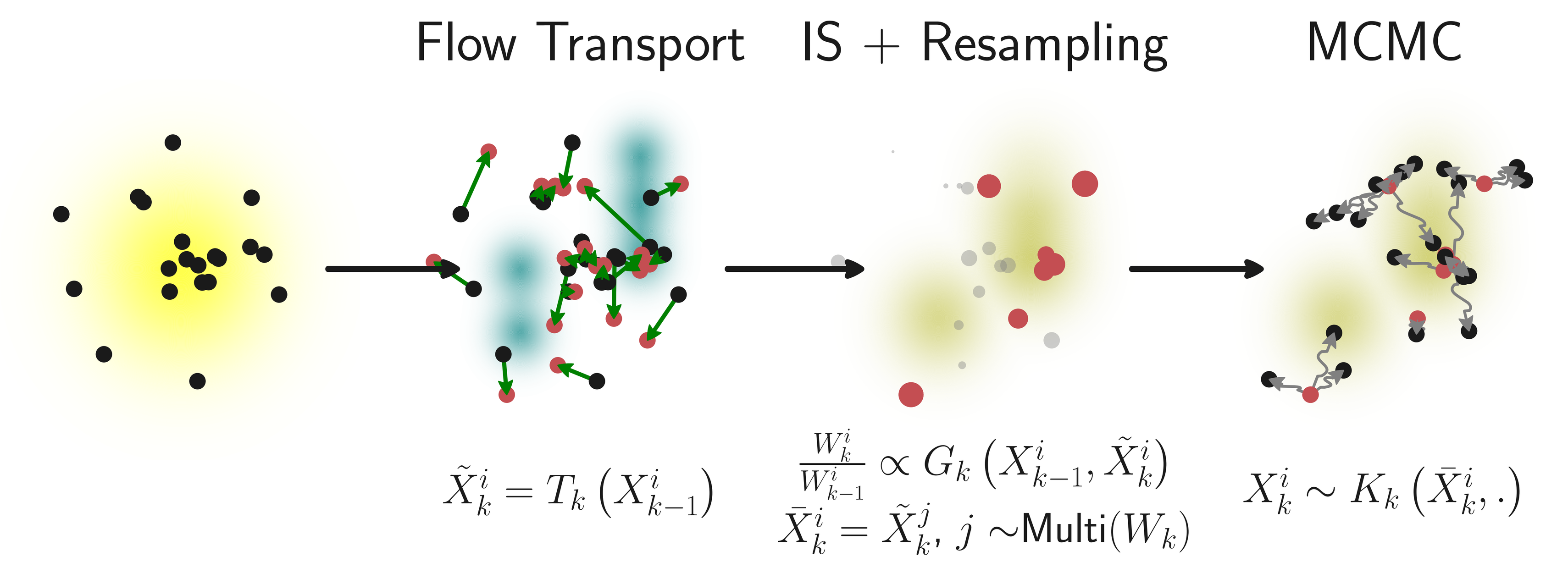
Annealed Flow Transport: Overview

Combines Sequential Monte Carlo (SMC) with Normalizing Flows (NFs).



- Similarly to SMC: Introduce a sequence of densities π_k interpolating between a proposal p and the target π.
- Sequential sampling: Use samples from π_{k-1} to compute samples from π_k.
- AFT step: NF transport step followed by standard SMC steps.

Annealed Flow Transport steps



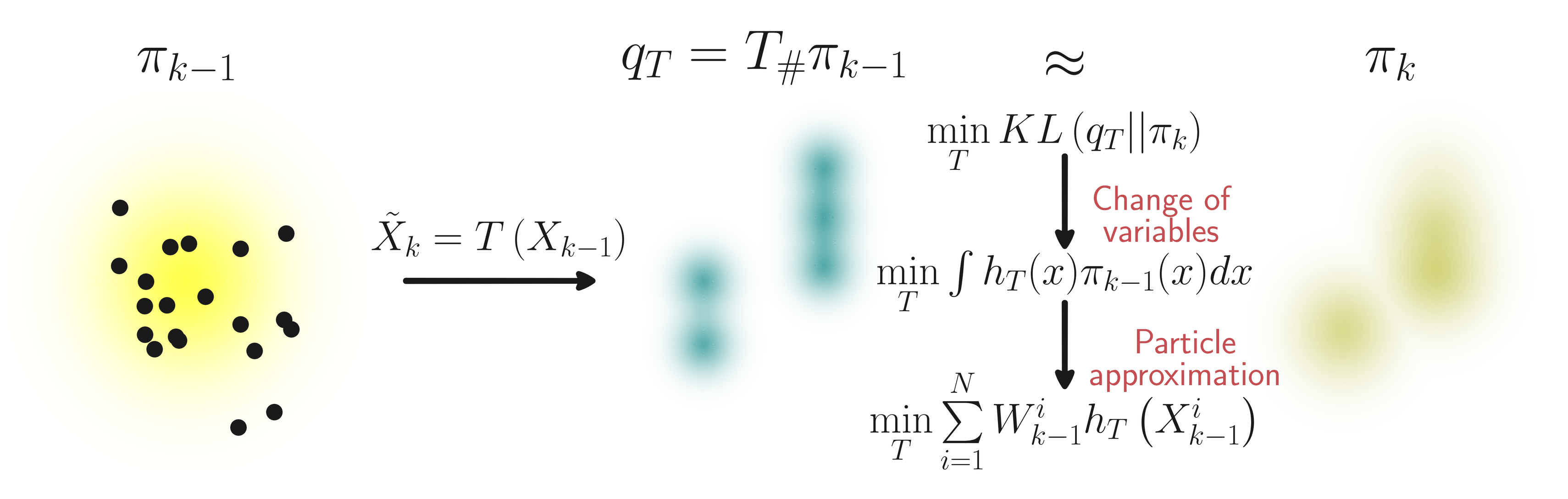
- *Flow Transport* T_k moves X_{k-1}^i to new particles $\tilde{X}_k^i$ close to π_k.
- *Closed-form* expression for the IS weights to correct for inexact flow:

$$G_k(X,Y) = \frac{\pi_k(Y)}{\pi_{k-1}(X)}|\nabla T_k(X)|$$

- Importance Sampling: re-weights particles $\tilde{X}_k^i$ proportionally to $G_k(X_{k-1}^i, \tilde{X}_k^i)$.
- Resampling: duplicate particles with large weights and discard those with small weights. (Recovers Annealed Importance Sampling (Neal, 2001) if no resampling.)
- MCMC step: Move particles according to a Markov Kernel K_k with invariant distribution π_k (HMC, Gibbs samplers, etc).
- Estimating normalizing constant Z_k sequentially:

$$Z_k^N := Z_{k-1}^N \left(\sum_{i=1}^N W_{k-1}^i G_k\left(X_{k-1}^i, \tilde{X}_k^i\right)\right)$$

Learning the flow sequentially



- Change of variables: $KL(q_T||\pi_k)$ as an expectation under π_{k-1} of a function $h_T(x)$

$$h_T(x) = \log \pi_{k-1}(x) - \log \pi_k(T(x)) - \log|\nabla T(x)| + C$$

- Particle approximation: Use particles X_{k-1}^i and weights W_{k-1}^i to estimate expectation of h_T under π_{k-1}.
- Extension to prevent overfitting and biased estimation: Use three sets of particles:
  - Train: Used to estimate the gradient of the loss.
  - Validation: Used for early stopping of training.
  - Test: Not used to estimate the flow. Gives unbiased estimates of normalizing constant and robust samples.

Theory I: Consistency and Asymptotic Nomality

- AFT produces estimates π_K^N and Z_K^N of π and Z that are consistent as N grows:

$$\pi_K^N[f] \xrightarrow{P} \pi[f],$$
$$Z_K^N \xrightarrow{P} Z.$$

- Fluctuations of the estimates satisfy a Central Limit theorem:

$$\sqrt{N}\left(\pi_K^N[f] - \pi[f]\right) \xrightarrow{\mathcal{D}} \mathcal{N}(0, V^\pi[f])$$
$$\sqrt{N}\left(Z_K^N - Z\right) \xrightarrow{\mathcal{D}} \mathcal{N}(0, V^Z)$$

- Extends results of SMC algorithms using tools from empirical process theory.
- $V^\pi[f]$ matches the variance under π if the flows T_k exactly map π_{k-1} to π_k.

Theory II: Continuous-time limit

- Setting:
  - Population limit: Infinitely many particles $N \to +\infty$
  - Unadjusted Langevin kernel for K_k.
  - Continuous-time limit: Infinitely many auxiliary densities $(\pi_k)_{k=1}^K \to (\pi_t)_{[0,1]}$.
- AFT recovers a weighted controlled diffusion:
  - Sample paths $X_{0,t}$ follows a controlled SDE with control α_t:

$$dX_t = (\alpha_t^\star(X_t) + \nabla_x \log \pi_t(X_t))dt + \sqrt{2}dB_t$$

  - Sample paths $X_{[0,t]}$ are re-weighted according to:

$$w_t^{\alpha^\star}(X_{[0,t]}) := \exp\left(\int_0^t g_s^{\alpha^\star}(X_s)ds\right), \quad g_s^\alpha(X_s) := \nabla_x \cdot \alpha_t + \alpha_t^\top \nabla_x \log \pi_t + \partial_t \log \pi_t$$

  - Weights ensure the marginals of weighted diffusion match π_t exactly.
  - *Instantaneous work* g_s^α measures how much the density of X_t differs from π_t.
  - Optimal control $\alpha^\star$ obtained by minimizing the variance of *Instantaneous work*:

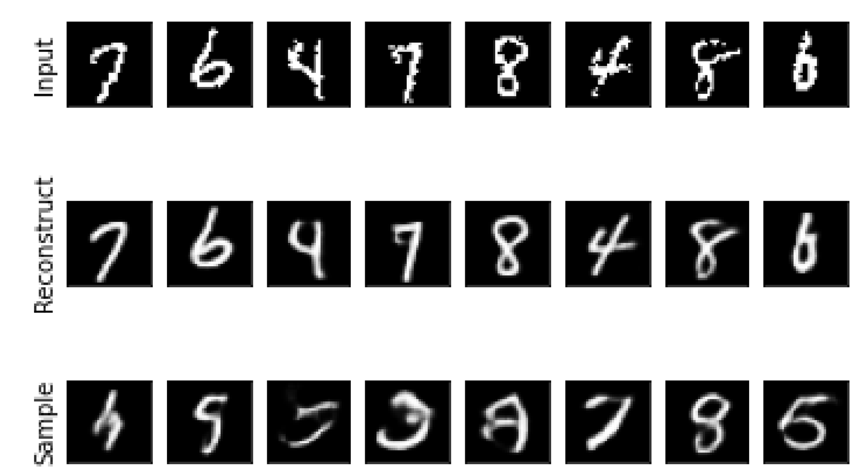
$$\alpha^\star := \arg\min_\alpha \int_0^1 dt \left(\pi_t[(g_t^\alpha)^2] - \pi_t[g_t^\alpha]^2\right).$$

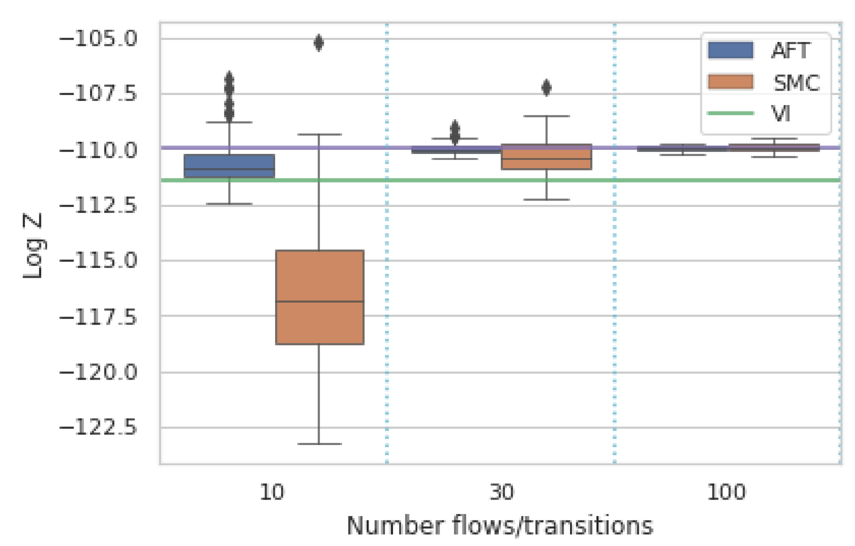
Empirical Evaluation: Setup

- Evaluation setup: We evaluate the trained extended algorithm (3 sets of particles)
  - Corresponds to using the test set particles with learned NFs.
  - Could be deployed in a larger setup and/or on massive parallel compute.
- Performance measure: Number of transitions/flows as a proxy for compute time.
  - Assumes overhead of flow is negligible relative to sampling.
  - Works for AFT and SMC our primary baseline. VI is fast where we use it.
- Choice of the Markov kernel: Same Markov kernel for AFT and SMC.
- Choice of the Flow: Element-wise affine flow.
  - Has the benefit of linear memory/time in the dimension.
  - Not very expressive on its own, and is closed under composition of the flows.

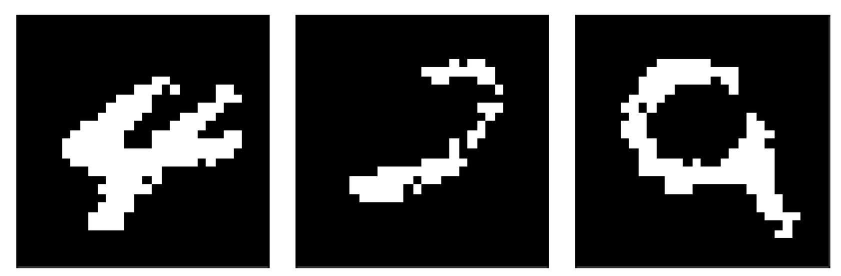
Empirical Evaluation I: VAE Latent Space sampling

- Task: Sampling from the posterior of trained VAE on Mnist digits.



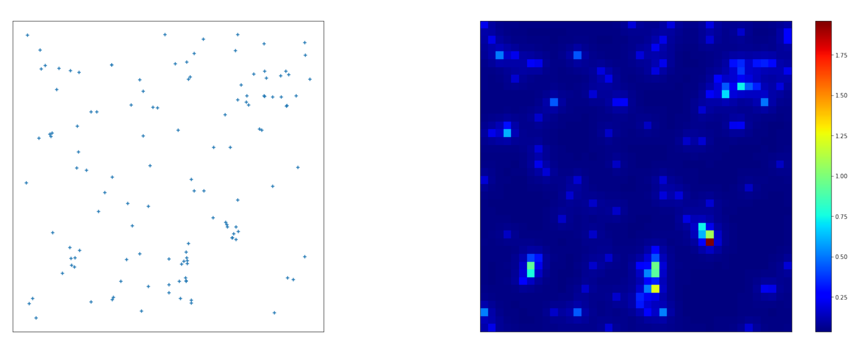


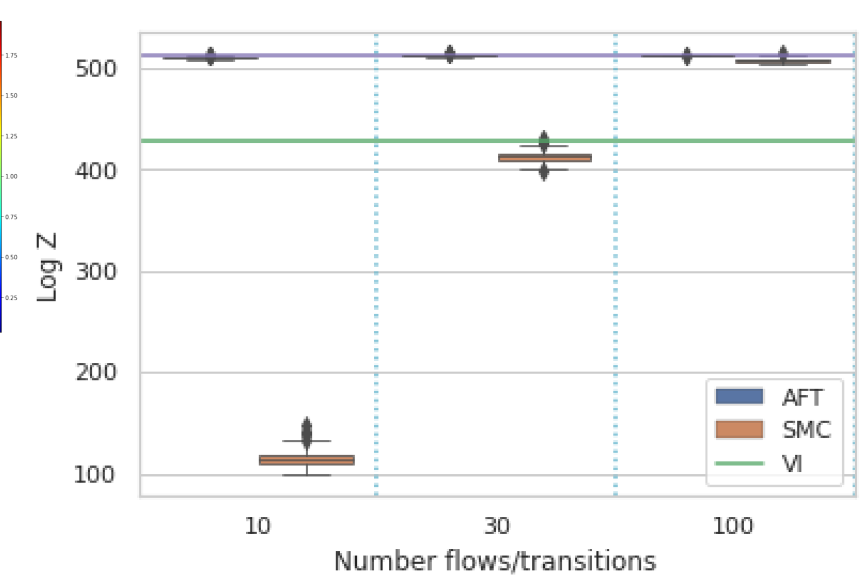
- We identify digits that are harder for variational inference:



- Variational inference works reasonably but is exceeded by SMC and AFT eventually.
- AFT has lower variance than SMC particularly for smaller number of temperatures.

Empirical Evaluation II: Log Gaussian Cox Process Posterior





$$\pi(x) \propto \mathcal{N}(x, \mu, K) \prod_{i \in [1:M]^2} e^{x_i y_i - a e^{x_i}}.$$

- Becomes harder as lattice resolution increases
- We use a 40×40 lattice giving 1600 dimensions.
- AFT *significantly outperforms baselines.*
- All methods could be further tailored.

This work was funded by DeepMind and the Gatsby Charitable Foundation.

Contact: michael.n.arbel@gmail.com, alexmatthews@google.com, arnauddoucet@google.com