Assessing Advanced Annealing Samplers for Difficult Distributions

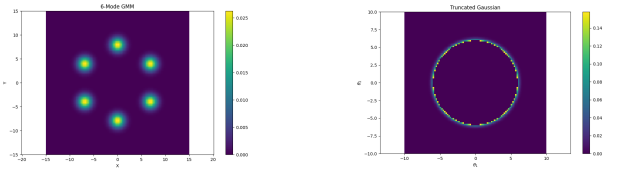
Mauricio Barba de Costa [*1] **Fritzgerald Duvigneaud** [*1]

Abstract

We survey a sampling algorithm introduced in a recent paper called "Annealing Flow" (AF)(Wu & Xie, 2024). This algorithm combines annealed importance sampling (AIS)(Neal, 1998) which uses intermediate distributions between the initial and target distributions to transform samples from the initial distribution "smoothly" using weights, and normalizing flow(Parno & Marzouk, 2018) which seeks to learn an explicit map between an initial and target distribution during sampling steps to directly transform samples drawn from the initial distribution across. Annealing flow learns neural-network parameterized maps between each intermediate distribution of AIS. Its authors argue that the gradual evolution afforded by the annealing makes their method better suited than MCMC-based approaches for sampling from difficult target distributions such as Gaussian mixture models. In this work we benchmark their algorithm in sampling from difficult distributions against our implementation of AIS, a hybrid sampler we experimentally developed that combines AF with Metropolis-Hastings updates, and a parallelized Hamiltonian Monte Carlo sampler. We report empirical evidence of our hybrid samplers improving upon AF and of HMC's weaknesses being mitigated by the use of multiple chains.

1. Introduction

Practitioners often find that MCMC sampling methods can get stuck in local optima or fail to traverse the breadth of a distribution efficiently, especially if there are multiple modes. This can be due to random walk behavior leading to samplers getting stuck in local modes and causing poor exploration. Additionally, despite providing asymptotic convergence guarantees, MCMC methods often suffer from very slow convergence rates to the target distribution. We term distributions that pose challenges to MCMC samplers such as multimodal distributions or those with very dramatic changes in gradient of the log probability over small distances in parameter space "difficult distributions". This report centers around two in particular: A radially symmetrical two-dimensional Gaussian Mixture Model (GMM) with six modes at a radius 8, and a truncated two-dimensional Gaussian distribution with zero mean, such that all probability mass lies outside the truncation radius 6. These distributions are visualized as heatmaps in Figure 1. For our samplers that rely on gradients of the log probability, we add infinitesimal amounts of probability mass to regions of zero mass to avoid numerical errors in sampling attempts.



(a) (b)

Figure 1. Probability density heatmaps of the radial GMM and truncated Gaussian distributions.

To visually portray the ability of our sampling algorithms to sample from these difficult distributions, in this work we follow the convention of overlaying the samples generated by these algorithms over these heatmaps as light blue pixels.

2. AIS

Annealed Importance Sampling (AIS) is a sampling algorithm that evolves samples from an initial distribution to a target distribution through a sequence of intermediate distributions (intended to heuristically "smooth" the transformation from the initial to the target distribution) and corrects approximation error using importance weights.

AIS addresses the challenge of sampling from complex target distributions by constructing a path of intermediate distributions between an easy-to-sample initial distribution $\pi_0(x)$ and the target distribution $\pi_K(x) \propto \tilde{q}(x)$. These intermediate distributions are typically defined as:

[*]Equal contribution [1]Department of EECS, MIT, Cambridge, Massachusetts, USA. Correspondence to: Mauricio Barba de Costa <barba@mit.edu>, Fritzgerald Duvigneaud <fritzduv@mit.edu>.

Template borrowed from *Proceedings of the 42^{nd} International Conference on Machine Learning*, Vancouver, Canada. PMLR 267, 2025.

$$\pi_k(x) \propto \pi_0(x)^{1-\beta_k} \tilde{q}(x)^{\beta_k} \tag{1}$$

where $0 = \beta_0 < \beta_1 < \cdots < \beta_K = 1$ is a sequence of inverse temperatures that gradually shifts from the initial distribution to the target.

The algorithm proceeds by first drawing samples from the initial distribution $\pi_0(x)$. Then, for each intermediate distribution π_k, it applies MCMC transition kernels T_k that leave π_k invariant. As samples evolve through this sequence, importance weights accumulate to correct for the discrepancy between the sampling path and the target distribution.

The importance weight for a sample trajectory $\{x_0, x_1, \ldots, x_K\}$ is computed as:

$$w = \prod_{k=1}^{K} \frac{\pi_k(x_{k-1})}{\pi_{k-1}(x_{k-1})} = \prod_{k=1}^{K} \left(\frac{\tilde{q}(x_{k-1})}{\pi_0(x_{k-1})} \right)^{\beta_k - \beta_{k-1}} \tag{2}$$

This approach effectively navigates multimodal distributions where standard MCMC methods might get trapped in local modes. It provides unbiased estimates of the normalizing constant, crucial for model comparison. AIS generally exhibits better mixing properties than single-chain MCMC when intermediate distributions are well-chosen.

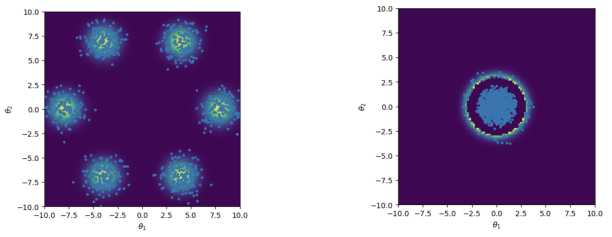
Despite these advantages, AIS still faces efficiency challenges in high-dimensional spaces or with particularly complex target distributions. The method's performance depends critically on the choice of intermediate distributions and transition kernels. These limitations motivated our investigation of Annealing Flow, which combines the annealing strategy with normalizing flows to potentially improve sample efficiency and accuracy.

While AIS does well on many distributions, there are still some cases where it does poorly. Figure 2 shows 1000 AIS samples from the radial 6-mode GMM and truncated Gaussian distributions with 15 intermediate distributions. While samples from all six modes are visible for the radial distribution, the truncated distribution has most samples inside the truncation region with effectively no probability mass.

We see that many samples get stuck in the region outside the support.

3. Annealing Flow

The Annealing Flow (AF) algorithm extends the annealing concept by learning a continuous normalizing flow to gradually map an initial easy-to-sample density $\pi_0(x)$ to the target density $q(x)$. Rather than relying on Metropolis-Hastings steps to correct for approximation errors as in AIS,



(a) (b)

Figure 2. 1000 AIS samples truncated Gaussian overlaid on difficult distribution heatmaps (a) 6-Mode Radial GMM (b) Truncated Gaussian.

Table 1. Performance comparison of sampling methods on the Truncated Gaussian posterior (c = 6, n = 5016 samples, 0 tuning samples)

	NUTS 4-Chains	**NUTS 24-Chains**	**AIS**	**AF**
ESS	440	279	NA	NA
t (in secs)	9	52	NA	NA
ESS$/t$	48.89	5.37	NA	NA

Table 2. Performance comparison of sampling methods on the Radial Gausian posterior (m = 6 modes, r = 8, n = 5016 samples, 0 tuning samples)

	NUTS 4-Chains	**NUTS 24-Chains**	**AIS**	**AF**
ESS	5	30	NA	NA
t (in secs)	13	49	NA	NA
ESS$/t$	0.385	0.612	NA	NA

ESS on GMM	
NUTS	598.00
AIS	175.99
Metropolis-Hastings	47.67
AF with wrong flow alone	1.06
AF with wrong flow + MCMC	199.67
AF with correct flow alone	623.87
AF + MCMC	700.12
Actual	1000
ESS on Truncated Gaussian	
NUTS	453.32
AIS	NA
Metropolis-Hastings	47.80
AF with wrong flow alone	30.09
AF with wrong flow + MCMC	NA
AF with correct flow alone	501.50
AF + MCMC	NA
Actual	1000

Table 3. Effective sample size, estimated expected value, and estimated variance of the three samplers. We could not determine importance weights for some methods because the initial distribution of our samples is in the region of the truncated Gaussian with no support, causing Equation 2 to be undefined.

AF directly learns an explicit transport map between the initial and target distributions.

Once trained, the continuous normalizing flow map T transforms samples from $\pi_0(x)$ to $q(x)$ by integrating the learned velocity field $v_k(x(t), t)$ for each intermediate block:

For each block k, the velocity field v_k is learned using a neural network trained to minimize a dynamic optimal transport objective derived from the Benamou-Brenier equation:

This objective consists of a KL divergence term between the pushforward density $T_k \# f_{k-1}$ and the target intermediate density f_k, regularized by a Wasserstein distance term that encourages smooth transitions. The parameter $\gamma > 0$ controls the trade-off between these two terms. The training process learns each transport map T_k sequentially, with each block transforming samples from density f_{k-1} to f_k.

Unlike standard MCMC methods, AF produces independent samples without correlation issues, making it particularly effective for high-dimensional and multi-modal distributions. The annealing procedure is crucial for successfully sampling from distributions with widely separated modes, where direct transformation without intermediate steps would likely fail. After training, the sequential application of these learned transport maps enables efficient sampling from complex target distributions. We call samples that are transformed under the flow map **pushforward samples**.

The Annealed Flow algorithm is designed to test the hypothesis posed by its authors that:

1. It is difficult to learn an accurate flow map between two extremely different distributions, especially when the target exhibits properties disadvantageous to sampling.

2. By constructing blended and thus less different intermediate distributions between the two, more accurate flow maps (between intermediate distributions) can be learned. Thus, samples pushed through all intermediate distributions can be expected to more closely track the target distribution than when only one flow map is used.

Our empirical results experimenting with the number of annealed distributions seem to support this hypothesis. Figure 3 shows 1000 pushforward samples through AF's learned flow maps from the radial GMM and truncated GMM using 5 intermediate distributions. Figure 4 shows 1000 pushforward samples through flowmaps of the same distributions but that were trained on 20 intermediate distributions. Clearly, the samples pushed through twenty flowmaps demonstrate a closer fit to the underlying distributions. The key problem here is that if intermediate distributions are too far apart, samples produced by AF will lag behind, causing poorly the behavior observed in Figures 3 and 4

Algorithm 1 Block-wise Training of Annealing Flow Net as defined by (Wu & Xie, 2024)

Require: Unnormalized target density $\tilde{q}(x)$; an easy-to-sample $\pi_0(x)$; $\{\beta_1, \beta_2, \cdots, \beta_{K-1}\}$; Total number of blocks K.
Set $\beta_0 = 0$ and $\beta_K = 1$
for $k = 1, 2, \cdots, K$ **do**
  Set $\hat{f}_k(x) = \pi_0(x)^{1-\beta_k} \tilde{q}(x)^{\beta_k}$;
  Sample $\{x^{(i)}(t_0)\}_{i=1}^n$ from $\pi_0(x)$;
  Compute the pushed samples $x^{(i)}(t_{k-1})$ from the trained $(k-1)$ blocks via (14);
  Optimize $\mathbf{v}_k(\cdot, t)$ upon minimizing the objective function.
end for
(Optional Refinement Blocks)
for $k = K+1, K+2, \cdots, L$ **do**
  Set $\beta_k = 1$ and optimize $\mathbf{v}_k(\cdot, t)$ following the procedures outlined above.
end for

Algorithm 2 Inference with Annealing Flow Net

Require: Trained flow blocks $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K\}$; Number of samples n
Output: Samples from target distribution $\{x^{(i)}\}_{i=1}^n$
Sample $\{z^{(i)}\}_{i=1}^n$ from standard normal distribution $\mathcal{N}(0, I)$
Initialize $\{x_0^{(i)}\}_{i=1}^n = \{z^{(i)}\}_{i=1}^n$
for $k = 1, 2, \cdots, K$ **do**
  Apply k-th flow map: $x_k^{(i)} = \Phi_k(x_{k-1}^{(i)})$ using trained velocity field $\mathbf{v}_k$
end for
Return samples $\{x_K^{(i)}\}_{i=1}^n$ as samples from target distribution

4. Annealing Flow With Metropolis-Hastings

After observing that in the case of the truncated Gaussian distribution, AF's pushforward samples did not seem to match the true posterior well (Figure 2), we set out to improve the sample accuracy. To do so we posited whether the transformation of samples under Metropolis updates of the form

$$\begin{cases} x' & \text{with probability } \frac{p(x')}{q(x'; x_k^{(i)})} \\ x_k^{(i)} & \text{with probability } 1 - \frac{p(x')}{q(x'; x_k^{(i)})} \end{cases}$$

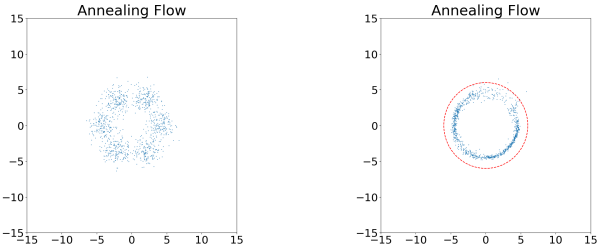
applied after each sample was pushed through each intermediate learned flow map of AF could "move" the samples to track the true posterior more closely. After implementing this algorithm using the trained AF normalizing flows and using 50 additional Metropolis updates, we observed that

$$x^{(i)}(t_k) = \mathcal{T}_k(x^{(i)}(t_{k-1})) = x^{(i)}(t_{k-1}) + \int_{t_{k-1}}^{t_k} v_k(x^{(i)}(s), s)ds, \quad k = 1, 2, \ldots, K. \tag{3}$$

For each intermediate distribution, indexed by k, we continuously evolve samples x through an optimal transport map $\mathcal{T}$. The transport map is determined by a velocity map v_k that we represent as a neural ODE. The velocity map v_k is trained to learn an optimal transport between distributions π_{k-1} and π_k using the objective from Equation 4.

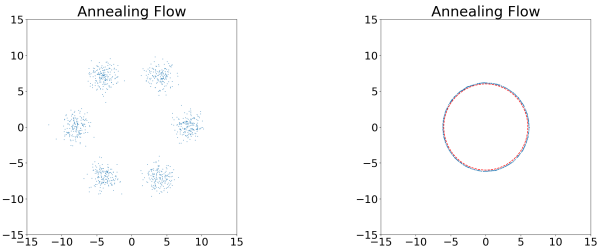
$$\mathcal{T}_k = \arg\min_{\mathcal{T}} \left\{ \mathrm{KL}(\mathcal{T}_{\#} f_{k-1} || f_k) + \gamma \int_{t_{k-1}}^{t_k} \mathbb{E}_{x(t)\sim\rho_k(\cdot,t)} \|\mathbf{v}_k(x(t), t)\|^2 dt \right\}, \tag{4}$$

We derive an objective for the optimal transport $\mathcal{T}_k$ from the Benamou-Brenier equations (Benamou & Brenier, 2000). Minimizing this objective is a matter of taking unbiased stochastic gradients and using them in a hill climbing algorithm.



(a) (b)

Figure 3. 1000 Pushforward samples through Annealing Flow maps trained on the difficult distributions of interest with 5 intermediate distributions. (a) 6-Mode Radial GMM (b) Truncated Gaussian



(a) (b)

Figure 4. 1000 Pushforward samples through Annealing Flow maps trained on the difficult distributions of interest with 20 intermediate distributions. (a) 6-Mode Radial GMM (b) Truncated Gaussian

Algorithm 3 Inference with Annealing Flow Net and MCMC Refinement

Require: Trained flow blocks $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_K\}$; Number of samples n; MCMC transition kernel $\mathcal{T}$; Number of MCMC steps M
Output: Samples from target distribution $\{x^{(i)}\}_{i=1}^n$
Sample $\{z^{(i)}\}_{i=1}^n$ from standard normal distribution $\mathcal{N}(0, I)$
Initialize $\{x_0^{(i)}\}_{i=1}^n = \{z^{(i)}\}_{i=1}^n$
for $k = 1, 2, \cdots, K$ **do**
  Apply k-th flow map: $x_k^{(i)} = \Phi_k(x_{k-1}^{(i)})$ using trained velocity field $\mathbf{v}_k$
  for $i = 1, 2, \cdots, n$ **do**
    for $j = 1, 2, \cdots, M$ **do**
      Apply MCMC transition kernel: $x_k^{(i)} \sim \mathcal{T}(x_k^{(i)}, \cdot)$
    end for
  end for
end for
Return samples $\{x_K^{(i)}\}_{i=1}^n$ as samples from target distribution

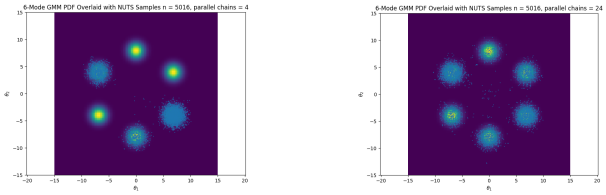
in the case of the Truncated Gaussian, the samples did in fact more closely correspond. Figure 10 shows the resulting MCMC corrected samples of the truncated Gaussian using the AF flow maps at each intermediate distribution.

Applying this algorithm with the incorrect trained flow maps (those corresponding to the 6-Mode GMM) revealed an even more interesting result: Despite the complete mismatch evident in the samples transformed under the wrong flow maps when overlaid with the truncated posterior, when the interspersed 50 Metropolis updates were applied, the 1000 original samples did in fact move to significantly better track the true posterior. See the experiments section for more info.

5. Hamiltonian Monte Carlo

The authors of AF report the results of Hamiltonian Monte Carlo (HMC) being used to sample from the same posteriors they test AF on to serve as a benchmark and highlight HMC's weaknesses. To replicate these results and further investigate these weaknesses, we used HMC via the No-U-Turn Sampler (Hoffman & Gelman, 2011) (NUTS) which automatically determines the number of steps L and step size ϵ parameters of HMC. We conducted tests on the same multimodal radial Gaussian Mixture and Truncated Gaussian distributions using PyMC(Abril-Pla et al., 2023) to model the distributions and sample from them using NUTS. Additionally, PyMC's parallelized implementation of NUTS allowed us to make use of a multicore cpu with 24 cores to run up to 24 sampler chains in parallel. Each parallel chain is randomly initiated in the parameter space.

We tracked the time to yield roughly 5000 samples and the resulting ESS of those samples for both distributions of interest in tables 1 and 2. We compared the results of dividing the samples among 4 and 24 parallelized NUTS chains. This yielded mixed results indicating that in some cases dividing sampling across more independent chains can be both an advantage and a disadvantage.



(a) (b)

Figure 5. 5016 NUTS samples 6-mode Radial GMM overlaid on heatmaps at different parallel chain counts (a) 4 parallel chains (b) 24 parallel chains.

Visually, we were able to see that in the case of the six mode GMM, sampling from NUTS using 24 chains led to complete exploration of all modes, whereas the use of 4 led to some modes not being explored for the same amount of samples. 5 This suggests that using multiple or even many chains may be a way to overcome the exploratory weakness of some MCMC methods in some cases. Additionally, parallelization could reduce the time cost of doing so.

6. Methods

We implement our experiments in Python and our work is accessable in our Github repository. We run all our MCMC experiments with 1000 chains and using only the last sample in our plots and in our ESS calculations. In the AIS and AF implementations, we use an annealing schedule β that starts at $\beta_0 = 0$ and increments by 0.125 for 8 steps until staying at $\beta_8 = 1$ for another 7 iterations. For AIS and our AF with MCMC experiment, we run 50 iterations of Metropolis-Hastings between each intermediate distribution. The proposals of our Metropolis-Hastings step is a normal distribution with standard deviation 0.1. The initial distribution for all our methods is a standard normal distribution. We run our NUTS and Metropolis-Hastings baselines for 750 iterations.

We use the annealing flow network proposed in (Wu & Xie, 2024), which is a neural network with 2 hidden layers of size 32 and Softplus activation functions composed with a neural ODE.

We compute importance weights for our baseline methods (NUTS, AIS, Metropolis-Hastings) and our proposed methods (AF, AF+MCMC) and measure the performance of the methods with the effective sample size. We were unable to obtain importance weights for the truncated Gaussian, however, as our initial distribution starts in the region of the truncated Gaussian outside the support. Our results are reported in Table 3.

7. Experiments

While it is evident i.e. from 8 that annealing demonstrates success in transforming samples from the initial to the target distribution, we were interested in assessing empirically how the number of intermediate distributions affects the accuracy of the pushforward samples.

Figure 6 shows the results of running annealing flow an a misaligned flow map. We observe that sample points are scattered according to the truncated Gaussian instead of the 6-mode GMM. Table3 shows that the effective sample of this method is expectedly low compared to the baselines.

Now, if we simply apply Metropolis-Hastings to every sample between each intermediate distribution, then we get Figure 7. The samples clearly are still not perfect but the figure suggests that combining AF with MCMC methods offers higher quality samples than AF alone.

Figure 8 shows samples drawn using the correct trained AF map. The problem with this example is that neural networks are not perfect approximators. While assigning importance weights may produce asymptotically unbiased expected value estimates with annealing flows, the quality of those estimates can still have extremely slow convergence if the samples are poor.

Finally, Figure 9 shows the result of combining AF with MCMC. We observe the samples cross the 15 intermediate distributions. We see the value that the flow map offers compared to the AIS sampler depicted in Figure 2. The flow map has the effect of transporting samples out of the region of no support in the truncated Gaussian. We can observe

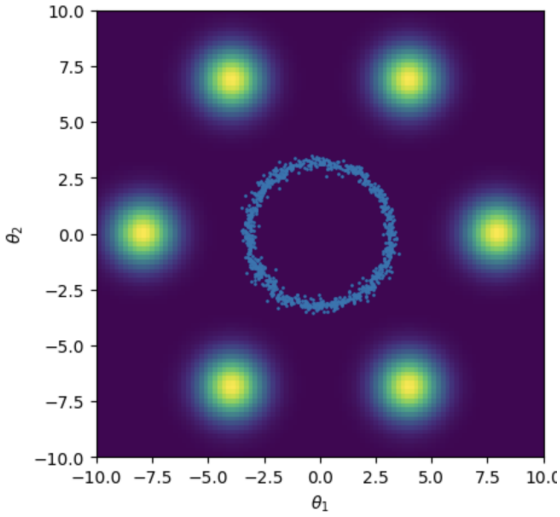


Figure 6. Points sampled using the wrong flow map (blue scatter points) overlayed on a heatmap of the 6-mode GMM target distribution.

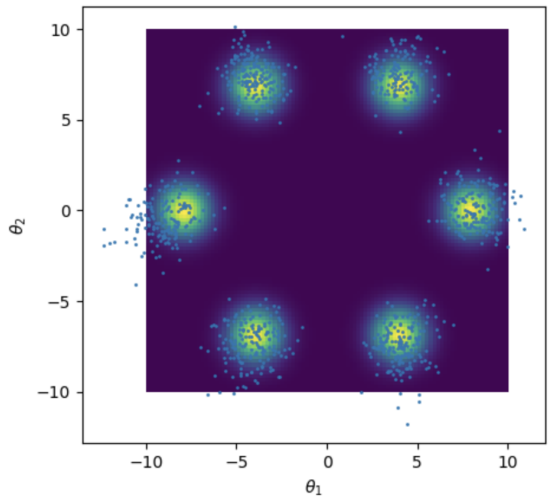


Figure 7. The result of combining AF with MCMC on the 6-mode GMM using the flow map that was trained for the truncated Gaussian. Observe that samples are more spread out than Figure 6 might suggest. Interestingly, the reason for this is that the flow map has the effect of stretching out space after approximately 10 iterations. A visualization of the flow map is offered in in Figure 10. This demonstrates the sensitivity of neural networks to out-of-distribution data.

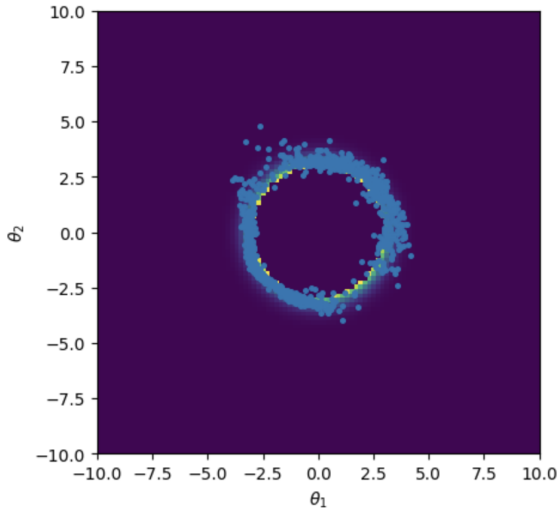


Figure 8. Samples yielded by AF with the correct trained flow maps overlayed on a heatmap of the truncated Gaussian. While samples are reasonable, the points have an asymmetric density with greater density on the top right and bottom left of the ring.

this clearly in Figure 10, where we observe that every flow map learns to push samples away from the origin. Together with the MCMC adjustment, it appears that the samples at the end could plausibly have been sampled from the target distribution.

While results in Figure 5 confirm that HMC can struggle in sampling from multimodal distributions due to chains being trapped in the typical set of a mode, we sought to assess if the use of multiple chains can mitigate this weakness.

8. Conclusion

In the practical application of Bayesian inference, posterior distributions may exhibit features like multimodality and large variations in the gradient of the log density. Such characteristics pose significant challenges for traditional sampling approaches like MCMC. The well-prepared statistician must thus have a trick up their sleeve for such cases. We have demonstrated that the combination of Annealing Flow (AF) with Metropolis-Hastings updates represents a powerful hybrid approach that overcomes limitations of both constituent methods.

While AF offers an elegant approach to sampling from multimodal distributions by learning explicit transport maps between intermediate distributions, our experiments reveal that these learned maps can sometimes fail to capture the true target distribution accurately, especially in challenging cases like the truncated Gaussian. This limitation stems from the inherent difficulties in neural network approximation of complex transformations. With this in mind, we demonstrated that incorporating Metropolis-Hastings up-

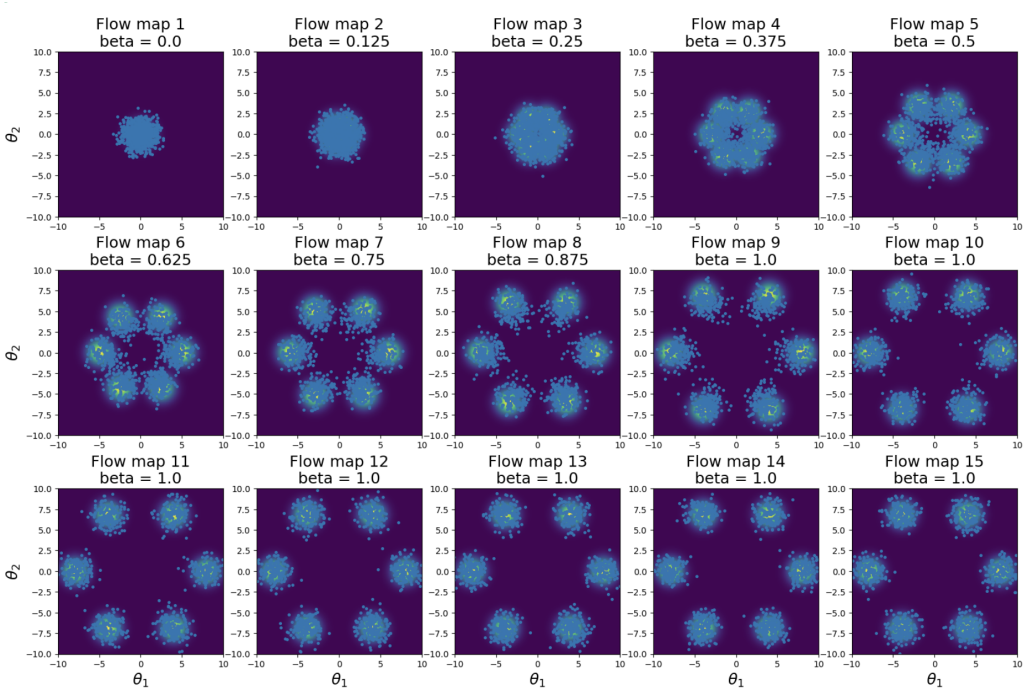


Figure 9. Samples yielded by AF with MCMC across all intermediate distributions.

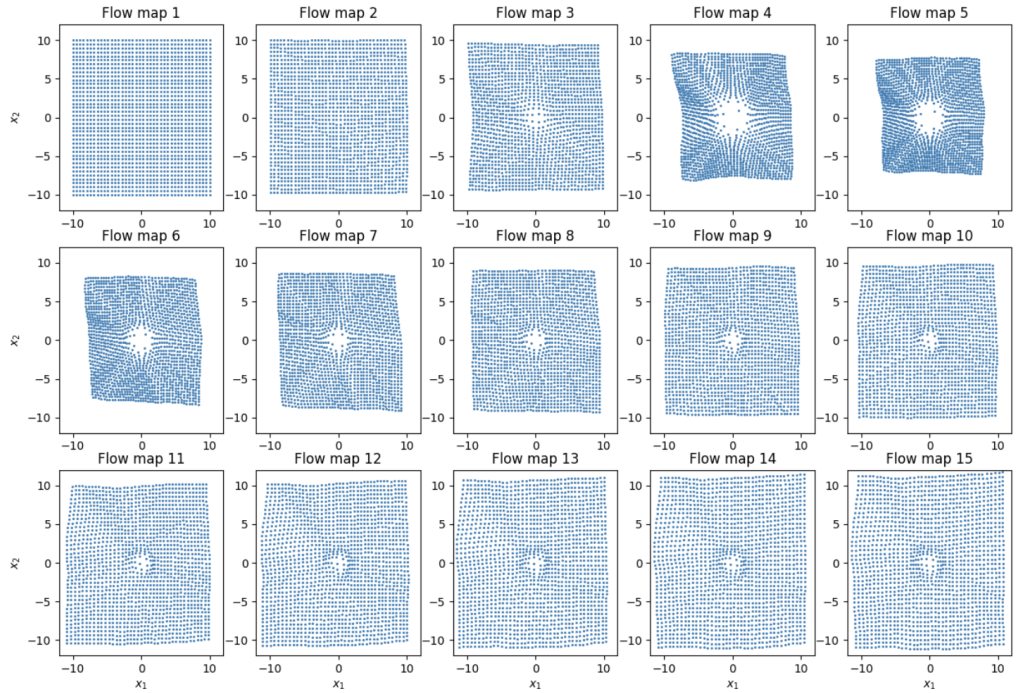


Figure 10. Visualization of flow map learned for truncated Gaussian. The top left plot shows the grid $\mathbf{xs}$ given by 10,000 points spaced evenly on the grid $[-10, 10]^2$. The remaining 14 plots show $\mathcal{T}_k(\mathbf{xs})$ where $\mathcal{T}_k$ is the flow map learned from the $(k-1)$th to the kth intermediate distribution.

dates between intermediate distributions substantially improves sample quality, even when using imperfect or misaligned flow maps. This finding suggests that the combination of global exploration through normalizing flows and local refinement through MCMC can be more effective than either approach alone. The hybrid approach achieves higher effective sample sizes than standard methods on both test distributions.

Additionally, some of our findings suggested that for some types of difficult distributions like multimodal distributions, MCMC methods' exploratory weakness may be compensated for by the use of multiple or several independent sampler chains. Additionally, parallelization can make this approach more feasible in terms of required computation time.

Future work could explore adaptive strategies for determining the optimal number of MCMC steps based on the quality of the learned flow maps, methods for automatically detecting and addressing flow map misalignment, and extensions to higher-dimensional problems with more complex geometries. Additionally, developing theoretical guarantees for the convergence of these hybrid samplers presents an novel direction for further research.

8.1. Divison of work

Mauricio and Fritz worked equally on all aspects of this project.

Acknowledgements

We thank Professor Tamara Broderick, Renato Berlinghieri, and Vishwak Srinivasan for their invaluable insight into and advice for our project.

"This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here."

The above statement can be used verbatim in such cases, but we encourage authors to think about whether there is content which does warrant further discussion, as this statement will be apparent if the paper is later flagged for ethics review.

References

Abril-Pla, O., Andreani, V., Carpenter, B., Fonnesbeck, C., Kumar, R., Martin, O., Salvatier, J., Warmenhoven, J., and Wiecki, T. Pymc: A modern and comprehensive probabilistic programming framework in python. *PeerJ Computer Science*, 9:e1516, 2023. doi: 10.7717/peerj-cs.1516.

Benamou, J.-D. and Brenier, Y. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84(3):375–393, 2000.

Hoffman, M. D. and Gelman, A. The no-u-turn sampler: Adaptively setting path lengths in hamiltonian monte carlo, 2011. URL https://arxiv.org/abs/1111.4246.

Neal, R. M. Annealed importance sampling, 1998. URL https://arxiv.org/abs/physics/9803008.

Parno, M. D. and Marzouk, Y. M. Transport map accelerated markov chain monte carlo. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):645–682, January 2018. ISSN 2166-2525. doi: 10.1137/17m1134640. URL http://dx.doi.org/10.1137/17M1134640.

Wu, D. and Xie, Y. Annealing flow generative model towards sampling high-dimensional and multi-modal distributions, 2024. URL https://arxiv.org/abs/2409.20547.