Folded Hamiltonian Monte Carlo for Bayesian Generative Adversarial Networks

Narges Pourshahrokhi N.POURSHAHROKHI@SURREY.AC.UK
Yunpeng Li YUNPENG.LI@SURREY.AC.UK
School of Computer Science and Electronic Engineering, University of Surrey, UK

Samaneh Kouchaki SAMANEH.KOUCHAKI@SURREY.AC.UK
School of Computer Science and Electronic Engineering, University of Surrey, UK
UK Dementia Research Institute, Care Research and Technology Centre

Payam Barnaghi P.BARNAGHI@IMPERIAL.AC.UK
Imperial College London, Department of Brain Sciences, UK
UK Dementia Research Institute, Care Research and Technology Centre

Editors: Berrin Yanıkoğlu and Wray Buntine

Abstract

Probabilistic modelling on Generative Adversarial Networks (GANs) within the Bayesian framework has shown success in estimating the complex distribution in literature. In this paper, we develop a Bayesian formulation for unsupervised and semi-supervised GAN learning. Specifically, we propose Folded Hamiltonian Monte Carlo (F-HMC) methods within this framework to learn the distributions over the parameters of the generators and discriminators. We show that the F-HMC efficiently approximates multi-modal and high dimensional data when combined with Bayesian GANs. Its composition improves run time and test error in generating diverse samples. Experimental results with high-dimensional synthetic multi-modal data and natural image benchmarks, including CIFAR-10, SVHN and ImageNet, show that F-HMC outperforms the state-of-the-art methods in terms of test error, run times per epoch, inception score and Frechet Inception Distance scores.

Keywords: Generative Adversarial Networks; Hamiltonian Monte Carlo; Data Imputation; Multi-modal;

1. Introduction

Generative adversarial networks (GANs) (Goodfellow et al., 2014) received traction in the field of deep generative models. The development of GANs covers a wide range of neural network architecture from multi-layer perceptrons to the BigGAN framework (Brock et al., 2019) with residual blocks and self-attention layers (Zhang et al., 2019) to synthesise realistic images.

Despite GAN's effectiveness in generating realistic images, it experiences mode collapse, which occurs when the generator over-optimises for a particular discriminator and the discriminator never learns how to escape the trap. Work has previously focused on alternative metrics such as f-diversities (Nowozin et al., 2016) or Wasserstein divergences (Arjovsky et al., 2017) to substitute the Jensen-Shannon divergence inherent in traditional GAN training to alleviate several practical issues.

Moreover, GANs encounter additional challenges when dealing with complex and diverse data sources, particularly in cases where multi-modal and highly correlated distributions come into play. For instance, in tasks like generating natural scenes, style transfer, speech synthesis, and molecular structure generation. They challenge GANs to capture complex relationships within data, highlighting the need for adaptable and expressive architectures. Addressing these challenges enables us to generate a wide range of diverse and high-quality samples across various applications. Saatci and Wilson (2017) proposed the Bayesian GAN, a probabilistic framework for GANs based on Bayesian inference. It demonstrates how modelling the distribution of generators alleviates mode collapse and motivates the interpretability of learned generators. GAN training measures the full posterior distribution across network weights in a single-mode based on mini-max optimisation. Even if the generator does not recall training instances, samples from the generator are expected to be excessively compact compared with data distribution samples. As a posterior distribution over the generators' parameters can have long tails and be highly multi-modal, the Bayesian GAN aims to model the real data distribution by fully reflecting the posterior distribution over the generators and discriminators' parameters. A similar approach, named as the probGAN He et al. (2019), iteratively learns a distribution over generators but with a carefully crafted prior. A tailored Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) is used in both the Bayesian GAN and the ProbGAN to approximate the posterior of the generators and discriminators' parameters.

In this paper, we explore regional adaptation to construct different samplers to efficiently produce samples of generators and discriminators' parameters, in order to mitigate GANs' mode collapse issues. To achieve this, we propose a Folded Hamiltonian Monte Carlo (F-HMC) to replace the SGHMC part of the Bayesian GAN framework. This proposed method is experimentally well adjusted to train GANs due to the adoption of Hamiltonian dynamics.

1. We design the Folded Hamiltonian Monte Carlo method with the Bayesian GAN framework to sample parameters of generators in using regional adaptation.

2. We analyse its theoretical properties such as ergodicity in converging to the target distribution.

3. We provide empirical evidence that it can accurately and efficiently explore multi-modal high-dimensional distribution in terms of similarity to the target distribution and convergence speed.

4. We apply the F-HMC method on natural image datasets (ImageNet, SVHN, and CIFAR10) and show that it outperforms probabilistic Bayesian GAN methods in terms of inception scores (IS) and Frechet Inception Distance scores (FID).

2. Problem Formulation

Suppose we have access to observed data $\mathcal{D} = \{x_i\}$ whose samples are distributed according to an unknown probability distribution $p_{\text{data}}(x)$. Our goal is to construct generators to sample from this potentially high-dimensional multi-modal distribution $p_{data}(x)$. Instead of finding one point in mini-max optimisation on the generator and discriminator in classic

GAN, Bayesian GAN (Saatci and Wilson, 2017) introduces a new formulation for GANs. It creates the distribution of the generator and discriminator's weight as an infinity space of generators and discriminators corresponding to every possible configuration of these weight vectors.

We build upon the problem formulation in the Bayesian GAN (Saatci and Wilson, 2017) and estimate $p_{\text{data}}(x)$ as $\text{Gen}(\hat{\alpha}_g, z)$ where z represents white noise sampled from $p(z)$, and $\hat{\alpha}_g$ represents distribution over generator parameters. We denote that parameter set α consisting of two sub parameter set $\hat{\alpha}_g$ related to the generators and $\hat{\alpha}_d$ associated with the discriminators. The Bayesian GAN can model the true data distribution by fully representing the posterior distribution over parameters of both the generator and the discriminator. Therefore, we require the generator and discriminators' weight candidates. In this regard, we need to estimate the posterior over $\hat{\alpha}_g$, $\hat{\alpha}_d$. We refer to this posterior as π in this paper.

First, generator weights $\hat{\alpha}_g$ are sampled from a prior $p(\hat{\alpha}_g|\beta_g)$ with β_g as hyperparameter, and a generative neural network is constructed conditioning on these samples $(\text{Gen}(.;\hat{\alpha}_g))$. Then, white noise z derived from $p(z)$ is transformed through the network $\text{Gen}(z;\hat{\alpha}_g)$ to generate candidate data samples. A discriminator conditioned on its weights $\text{Disc}(.;\hat{\alpha}_d)$ produces the probability that these candidate samples are generated from the true data distribution. The discriminator in Bayesian GAN is a link function that distinguishes true data from generated data and gets penalised by misclassifying the true data. As shown in Equation 1, if the discriminator outputs high probabilities, then the posterior will increase in a neighbourhood of the sampled setting of $\hat{\alpha}_g$, considering $L : [0,1] \to [0,1]$ as the likelihood term which is the product of the output probabilities of the discriminator.

$$p(\hat{\alpha}_g|z, \hat{\alpha}_d) \propto \exp\{L(\text{Disc}(\text{Gen}(z, \hat{\alpha}_g), \hat{\alpha}_d))\}p(\hat{\alpha}_g|\beta_g) \ . \tag{1}$$

From the discriminator side, it need to form classification likelihood that classifies actual data from the generated samples and can be formulated as:

$$p(\hat{\alpha}_d|z, X, \hat{\alpha}_g) \propto \exp\{L(\text{Disc}(X, \hat{\alpha}_d))\} \times \exp\{L(1 - \text{Disc}(\text{Gen}(z, \hat{\alpha}_g), \hat{\alpha}_d))\}p(\hat{\alpha}_d|\beta_d) \ . \tag{2}$$

Here $p(\hat{\alpha}_d|\beta_d)$ refers to the prior for $\hat{\alpha}_d$ with β_d as hyperparameter and $X = \{x_i\}_{i=1}^n$. As discussed in the Bayesian GAN, instead of implicitly conditioning the parameter posterior in traditional GAN on a set of noise samples z, we can marginalise z from the posterior using Monte Carlo:

$$p(\hat{\alpha}_g|\hat{\alpha}_d) = \int p(\hat{\alpha}_g, z|\hat{\alpha}_d)dz = \int p(\hat{\alpha}_g|z, \hat{\alpha}_d)p(z|\hat{\alpha}_d)dz \approx \frac{1}{I}\sum_{i=1}^{I} p(\hat{\alpha}_g|z^{(i)}, \hat{\alpha}_d), z^{(i)} \sim p(z) \ . \tag{3}$$

Similarly, $p(\hat{\alpha}_d|\hat{\alpha}_g) \approx \frac{1}{I}\sum_{i=1}^{I} p(\hat{\alpha}_d|z^{(i)}, X, \hat{\alpha}_g)$, $z^{(i)} \sim p(z)$. We can approximate the posterior π over $\hat{\alpha}_g$ and $\hat{\alpha}_d$ by iteratively sampling from $p(\hat{\alpha}_g|\hat{\alpha}_d)$ and $p(\hat{\alpha}_d|\hat{\alpha}_g)$. We then use the generators and discriminators sampled from π to generate candidate samples from the true data distribution $p_{\text{data}}(x)$. Figure 1 visualises the framework of Bayesian GAN. In practice, the distribution of $\hat{\alpha}_g$ and $\hat{\alpha}_d$ can be high-dimensional and multi-modal and using SGHMC falls short of covering the distribution. This motivated us to introduce our F-HCM method as an efficient sampling approach for this setup, particularly when the target is high-dimensional, multi-modal and highly correlated.

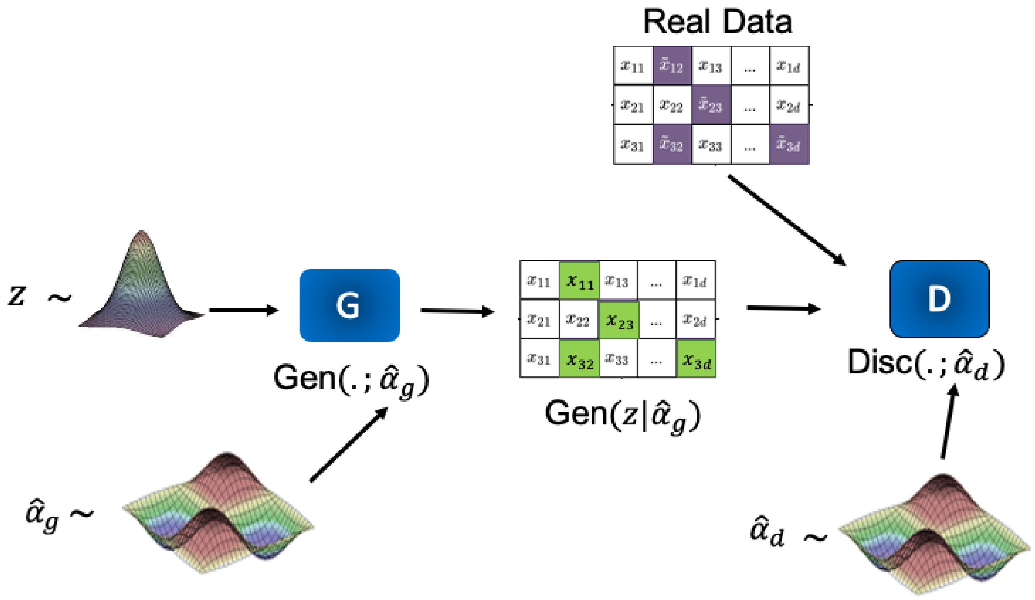

Real Data
x_{11} $\tilde{x}_{12}$ x_{13} ... x_{1d}
x_{21} x_{22} $\tilde{x}_{23}$ ... x_{2d}
x_{31} $\tilde{x}_{32}$ x_{33} ... $\tilde{x}_{3d}$
z ~
G
$\mathrm{Gen}(.;\hat{\alpha}_g)$
x_{11} x_{11} x_{13} ... x_{1d}
x_{21} x_{22} x_{23} ... x_{2d}
x_{31} x_{32} x_{33} ... x_{3d}
$\mathrm{Gen}(z|\hat{\alpha}_g)$
D
$\mathrm{Disc}(.;\hat{\alpha}_d)$
$\hat{\alpha}_g$ ~
$\hat{\alpha}_d$ ~

Figure 1: Bayesian GAN framework; a new formulation for classic GAN that explores distribution over generator and discriminator parameter ($\hat{\alpha}_g$, $\hat{\alpha}_d$)

3. Background

A well-known statistics problem is inadequate mixing of typical Markov Chain Monte Carlo (MCMC) techniques on multi-modal target distributions with isolated modes (Pompe et al., 2020). These algorithms have difficulty traversing through the low probability barriers that separate the modes and consequently require a long time for transition between the modes. Algorithms for sampling from a multi-modal target distribution need to address three major challenges: (i) determining high probability regions where the modes are located; (ii) crossing low probability boundaries to move between modes; (iii) considering homogeneity and local geometry of modes to sample efficiently from them.

If the modes have different local covariance structures, tempering-based techniques (Geyer, 1991; Miasojedow et al., 2012; Kou et al., 2022) were shown to integrate them exponentially slowly in dimension. On the other hand, the Smart Darting Monte Carlo method relies on two moves: leaps between modes, which are only permitted in non-overlapping pi-spheres surrounding the previously established local maxima, and local moves (Random Walk Metropolis steps). Sminchisescu and Welling (2011) expanded on this concept by enabling the leaping zones to overlap and have any volume and shape. Another line of research is optimisation-based approaches (Andricioaei et al., 2001; Sminchisescu and Welling, 2011), which aims to find local maxima of the target distribution.

Lan et al. (2013) proposed the Wormhole Hamiltonian Monte Carlo method, the extension of the Riemannian Manifold HMC as another optimisation-based approach. The key idea is to build a network of “wormholes” linking the modes (neighbourhoods of straight line segments between the local maxima of $p_{\text{data}}(x)$). Adjustments to the algorithm’s parameters, including the network system, are permitted during regeneration intervals. As we will see later, the algorithm we suggest also fits under the category of optimisation-based approaches. However, most of the methods described earlier only utilise a single transition kernel, restricting their performance in complex multi-modal scenarios which often require a diverse set of transition kernels to explore different regions.

We address this problem by proposing the Folded Hamiltonian Monte Carlo (F-HMC) method to increase the quality of data generated in the Bayesian GAN framework by alleviating its mode collapse and generate diverse samples to cover the target distribution.

4. Folded Hamiltonian Monte Carlo

In the scenario where the target distribution is multi-modal and highly correlated, the Stochastic Gradient Hamiltonian Monte Carlo (SGHMC) might not efficiently explore the target density (Ye and Zhu, 2018). As a pragmatic approach, we propose the Folded Hamiltonian Monte Carlo method, an algorithm that, instead of finding a single chain that samples from the whole distribution, combines samples from several chains. Those chains each explore a different region of the state space (e.g., a few modes only).

We consider regional adaptation in the design of the F-HMC sampler, in which the proposal distribution differs across parts of the sample space. The regions evolve as the simulation progresses since the intended distribution is uncertain. Suppose the target is well approximated using a mixture of Gaussians. In that case, it is reasonable to assume that each mixture component is a good proposal in a given region of the sample space. As a result, F-HMC uses a Gaussian mixture to approximate the target distribution (π), and the mixture parameters are updated in real-time using simulated samples. Suppose the approximation of the target distribution π with J modes at time n is:

$$q_n(x) = \sum_{j=1}^{J} \psi^{(j)} \mathcal{N}(x; \mu_n^{(j)}, \sigma_n^{(j)}) \ . \tag{4}$$

where $\psi^{(j)} > 0$ and $\sum_{j=1}^{J} \psi^{(j)} = 1$ and $\mathcal{N}(x; \mu_n^{(j)}, \sigma_n^{(j)})$ determines Gaussian distribution with mean $\mu_n^{(j)}$ and $\sigma_n^{(j)}$ covariance matrix. Using the mixture representation in (4), we define the sample space regions $S = \bigcup_{j=1}^{J} S_n^{(j)}$ for J modes. For each set $S_n^{(j)}$, π is more similar to $\mathcal{N}(x; \mu_n^{(j)}, \sigma_n^{(j)})$ than any other distribution entering (4). To be more precise, we define S such that the sum of differences between Kullback-Leibler (KL) divergence of regions $\Delta KL(S_n^{(1)}, S_n^{(2)}, \ldots, S_n^{(j)})$ be the maximum. By maximising the sum of differences in KL divergences, the aim is to guide the mixture model towards a setup that mitigates the divergence between the approximated distribution and the desired target distribution. Consider $J = 2$, for regions g, h, the ΔKL is defined on space A as follows:

$$\Delta KL(g, h|A) = \int_A \log(\frac{g(x)}{h(x)}) g(x) dx \ . \tag{5}$$

Here, we measure ΔKL on the partition of S for distributions of π and $\mathcal{N}(x; \mu_n^{(j)}, \sigma_n^{(j)})$. In other words, $S_n^{(j)}$, the j-th component of the set of mixture density components, dominates the others and is defined as follow:

$$S_n^{(j)} = \{x : \underset{j'}{\text{argmax}} \mathcal{N}(x; \mu_n^{(j')}, \sigma_n^{(j')}) = j\} \ . \tag{6}$$

Lemma 1: The sum of differences between Kullback-Leibler (KL) divergence of regions $\Delta KL(S_n^{(1)}, S_n^{(2)}, .., S_n^{(j)})$ is maximum on S. We refer the readers to Appendix A for the proof of Lemma 1. The approximation (4) along with the regions defined in (6) enable us to determine the proposal distribution of F-HMC. One may consider proposal distribution at time n as:

$$Q_n(x) = \sum_{j=1}^{J} \psi^{(j)} \mathcal{N}(x; \mu_n^{(j)}, \sigma_n^{(j)}) \ . \tag{7}$$

In other words, we would use the mixture's dominant component as a suggested distribution in each region $S_n^{(j)}$. While such a model may have acceptable local characteristics, it may not guarantee proper flow across different regions. Therefore, we are using a second or (fold) HMC on top of these samplers concerning the cross-correlation of partitions to allow a flow between different regions. Thus the F-HMC proposal distribution models are as follows:

$$Q_n'(x) = \epsilon \sum_{j=1}^{J} \mathcal{N}(x; \mu_n^{(j)}, \sigma_n^{(j)}) + (1-\epsilon) \mathcal{N}(x; \hat{\mu}_n^{(k)}, \hat{\sigma}_n^{(k)}) \ . \tag{8}$$

where $\sigma_n^{(j)}$ is the sample covariance matrix derived from those samples in $S_n^{(i)}$, whereas $\hat{\sigma}_n^{(k)}$ is a covariance matrix with respect to cross-correlation between all components estimated from all n samples in S. F-HMC sampling consists of two options for each step. The local-mode option preserves mode sampling, while the cross-mode option allows for jumps across separate posterior region. The parameter ϵ controls the selection of these two options.

4.1. Simulation parameter updates

As mentioned earlier, the simulation parameters $\mu_n^{(j)}$ and $\sigma_n^{(j)}$ should be updated on the fly each time new draws from the target distribution are added to the Monte Carlo sample. Our method is inspired by Andrieu and Moulines (2006) and Bai et al. (2011), who developed the technique in mixture with an adaptable independent Metropolis algorithm. At time $n-1$, suppose the parameter estimations are $\mu_{n-1}^{(j)}$, $\sigma_{n-1}^{(j)}$, and the current samples are $\{x_0, x_1, ..., x_{n-1}\}$. We define the mixture indicator w_n so that it equals j if and only if x_n is created from the j-th component of the mixture (6), in other words, $\omega_n^{(j)} = P(w_n = j|x_n; \mu_n^{(j)}, \sigma_n^{(j)})$. Now we have:

$$\omega_n^{(j)} = \frac{\mathcal{N}(x; \mu_n^{(j)}, \sigma_n^{(j)})}{\sum_{j'=1}^{J} \mathcal{N}(x; \mu_n^{(j')}, \sigma_n^{(j')})} \ . \tag{9}$$

To adhere to the guidelines suggested by Haario et al. (2001) for adaptive Metropolis algorithm with Gaussian proposal changing on the fly. for all $1 \leq j \leq J$, the estimation for parameter $\mu_n^{(j)}$ and $\sigma_n^{(j)}$ is as follow:

$$\begin{aligned}\mu_n^{(j)} &= \mu_{n-1}^{(j)} + \frac{\omega_n^{(j)}}{\sum_{i=1}^n \omega_i^{(j)}}(x_n - \mu_{n-1}^{(j)}) \\ \sigma_n^{(j)} &= \sigma_{n-1}^{(j)} + \frac{\omega_n^{(j)}}{\sum_{i=1}^n \omega_i^{(j)}}((x_n - \mu_{n-1}^{(j)})(x_n - \mu_{n-1}^{(j)})^T - \sigma_{n-1}^{(j)}) \ .\end{aligned} \tag{10}$$

Details of the derivations are available in Appendix B. As stated in Section 4, the covariance matrix of samples $\{x_1, x_2, ..., x_n\}$, $\hat{\sigma}_n^{(k)}$ is required, which is calculated as follows:

$$\hat{\sigma}_n^{(k)} = \hat{\sigma}_{n-1}^{(k)} + \frac{\omega_n^{(k)}}{\sum_{i=1}^n \omega_i^{(k)}}((x_n - \hat{\mu}_{n-1}^{(k)})(x_n - \hat{\mu}_{n-1}^{(k)})^T - \hat{\sigma}_{n-1}^{(k)}) \ . \tag{11}$$

4.2. Theoretical analysis

This section describes the ergodicity of the F-HMC sampler. The proof relies on Containment and Diminishing Adaptation conditions given by Roberts and Rosenthal (2001) to guarantee the ergodicity of adaptive MCMC. Formally, an adaptive MCMC technique for the target distribution π employs the transition kernel's parameters vector $H_n^{(j)} = [\mu_n^{(j)}, \sigma_n^{(j)}]$, which is permitted to vary throughout the simulation. It is supposed that the Markov chain kernel has a stationary distribution π for each parameter vector Meyn and Tweedie (1993). In other words, for a given initial point $x_0 \in S$ and a kernel ker_H, X_{n+1} is generated from $ker_H(X_n, .)$ at iteration $n+1$. We state that our suggested model is ergodic if:

Lemma 2: F-HMC satisfies Diminishing adaptation condition. This means that the distinction between the transition kernels employed throughout iterations n and $n+1$ diminish after long iterations.

Lemma 3: F-HMC satisfies Containment condition. This means that the process's convergence time is bounded in probability.

We refer the readers to Appendix C and Appendix D for proof of Lemma 2 and Lemma 3, respectively.

4.3. Integrating F-HMC with Bayesian GAN

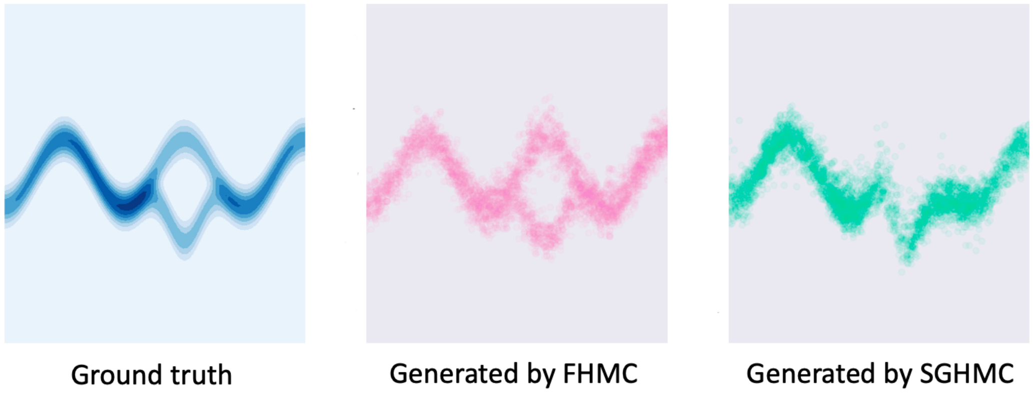
Since SGHMC is not efficient in exploring complex targets with multi-modality, we proposed F-HMC. The F-HMC design enables it to explore multi-modal targets efficiently. It remedies the challenge of mode collapse and covers target distribution modes by enabling cross mode jumps.

We substitute the SGHMC sampler with FHMC to make the proposed model operate within the Bayesian GAN framework. Recalling from Bayesian GAN, the posterior approximation is based on generating samples over π the posterior of generators $\hat{\alpha}_g$ and discriminators weights $\hat{\alpha}_d$, which is feasible by iteratively sampling from $p(\hat{\alpha}_g|\hat{\alpha}_d)$ and $p(\hat{\alpha}_d|\hat{\alpha}_g)$. We deployed F-HMC for estimating π to generate candidates for generators' and discriminators' weights. Finally, corresponding generators and discriminators are constructed to sample the multi-modal high dimensional target distribution.

Algorithm 1 One iteration of the generator in Bayesian GAN set up with our proposed F-HMC

```
1: **Input:** $(x_n, j)$ is current sample at mode $j$ out of total $J$ finite modes and $\hat{\alpha}_g^n$ is generator parameters from iteration $n$ and number of $\theta$ is HMC friction term, $\eta$ for the learning rate, and $I$ is the number of F-HMC iterations.
2: **for** $n = 1$ **to** $I$ **do**
3:     $z \sim p(z)$
4:     Generate $u \sim U[0, 1]$
5:     **if** $u > \epsilon$ **then**
6:         **Local mode move:**
7:         sample $y$ around current mode $j$ from $S_n^{(j)}$
8:         **if** $y$ is accepted **then**
9:             $(x_{n+1}, j) \leftarrow (y, j)$
10:        **else**
11:            $(x_{n+1}, j) \leftarrow (x_n, j)$
12:        **end if**
13:    **else**
14:        **Cross mode move:**
15:        propose new mode $k$
16:        sample $y$ around updated mode and $\hat{\mu}_n^{(k)}$ and $\hat{\sigma}_n^{(k)}$
17:        **if** $y$ is accepted **then**
18:            $(x_{n+1}, j) \leftarrow (y, k)$
19:        **else**
20:            $(x_{n+1}, j) \leftarrow (x_n, j)$
21:        **end if**
22:    **end if**
23:    update parameters $\mu_n^{(j)}$, $\sigma_n^{(j)}$, $S_n^{(j)}$, $\hat{\mu}_n^{(k)}$ and $\hat{\sigma}_n^{(k)}$
24:    **Bayesian GAN updates:**
25:    $n \sim \mathcal{N}(0, 2\theta\eta I)$
26:    $v \leftarrow (1 - \theta)v + \eta\nabla + n$
27:    $\hat{\alpha}_g^{n+1} \leftarrow \hat{\alpha}_g^n + v$
28: **end for**
29: **Output:** $\{\hat{\alpha}_g^n\}$ as generated samples for $\hat{\alpha}_g$
```

Considering total J finite modes, Algorithm 1 shows one iteration of the Bayesian learning for the generator parameters using our proposed model. We propose the sampling algorithm that relies on HMC steps of two types, performed with probabilities $(1-\epsilon)$ and ϵ, respectively. (Line 5-22). F-HMC assumes that the target is approximated using Gaussian mixture models and has J regions. On each iteration, the model selects the next move to either generate a sample from local mode with probability ϵ or takes the cross mode move with probability $(1 - \epsilon)$, enabling the model to explore other regions. In the case of the local move, the F-HMC proposes a sample within the current region j. If accepted, the model's next sample is updated (Line 9); Otherwise, it retains the sample on the former step (Line 11). In the case of the cross mode move, the model proposes a new region k and sample, and if they are accepted, the sample and mode in the next step are updated (Line 18). Otherwise, they remain unchanged (Line 20). It is important to note that J is a hyper-parameter, and in practice, it either relies on a satisfactory approximation of the target or can be determined by tuning the value.



Ground truth

Generated by FHMC

Generated by SGHMC

Figure 2: Both samplers converged to the target distribution, but the F-HMC (middle) covered the target (left) more accurately than SGHMC (right).

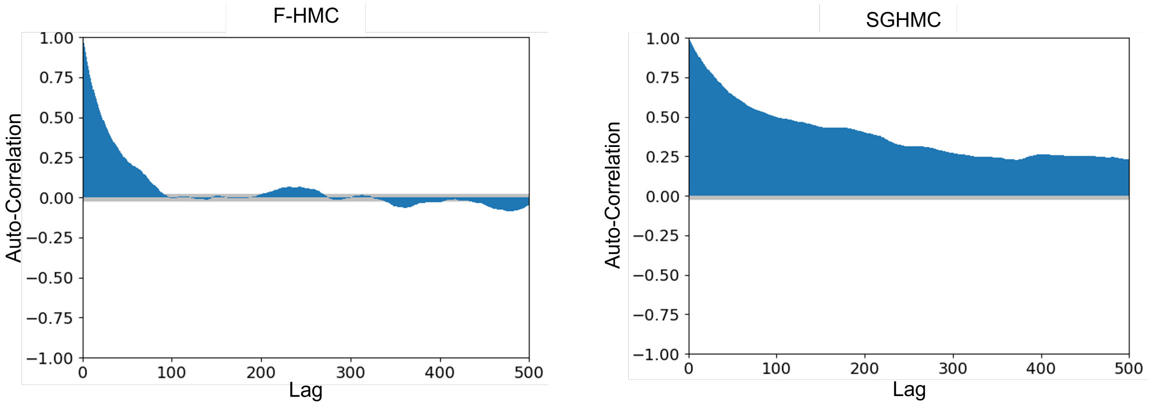


Figure 3: F-HMC converges faster than SGHMC in terms of lag number for exploring the target distribution in 100 dimensions.

5. Experiments

We implemented the proposed model using PyMC3 [1] and report its performance on generating samples from complex distributions described in Section 5.1. We examine the model's performance in marginalising the generators' parameters on synthetic and natural image datasets such as SVHN (Netzer et al., 2011), CIFAR 10 (Krizhevsky, 2009) and ImageNet (Deng et al., 2009) in Section 5.2 and 5.3, respectively. We compared results with WDCGAN (Arjovsky et al., 2017), DCGAN, 10DCGAN (which is a fully supervised convolutional neural network composed of ten DCGANs constructed by ten random subsets with 80% of the size of the training set, (Radford et al., 2016)), Bayesian GAN (Saatci and Wilson, 2017) under the Apache License, Version 2.0 and ProbGAN (He et al., 2019) on supervised and semi-supervised tasks with four different numbers of labelled examples. For a fair com-

[1] https://docs.pymc.io/

parison, each model has the same number of generators and discriminators with the same architecture. On both the generator and discriminator weights, a $\mathcal{N}(0, 10I)$ prior is used. We run the algorithm for 5000 iterations. So I, in algorithm 1 is 5000, we selected a learning rate (η) which decayed according to γ/d where γ is the per-batch learning rate set to 0.01 and d is the number of unique real data points as suggested by Saatci and Wilson (2017) and Chen et al. (2014). Guan and Krone (2007)'s theoretical derivation supports setting ϵ to 0.7. The memory requirement is 15 GB. All of the experiments were run on a TitanX GPU.

5.1. F-HMC performance

This section outlines the experiments we conducted to compare SGHMC with F-HMC's capacity to explore complex distributions. First, we use Normalising Flows (Kobyzev et al., 2020; Rezende and Mohamed, 2016) as a rich family of distributions to examine F-HMC and SGHMC's abilities to explore complex distribution. Figure 2 shows the potential energy of the rich target distribution and the generated candidates using SGHMC and F-HMC. F-HMC successfully covers the target distribution more accurately than SGHMC.

Secondly, we use auto-correlation between samples generated by each sampler as a metric to demonstrate the sampler's capability in exploring the target distribution. Less auto-correlation between samples provides more information about the target, with fewer samples, indicating a superior sampler. Figure 3 shows auto-correlation in F-HMC drops faster to zero than SGHMC in terms of lag number.

5.2. High-Dimensional Multi Modal Synthetic Dataset

To test the power of F-HMC in approximating a multi-modal posterior, we employ a multi-modal synthetic dataset. We generate D-dimensional synthetic dataset as follows:

$$z \sim \mathcal{N}(0, 10*I_d), \quad A \sim \mathcal{N}(0, I_{D\times d}), \quad x = Az+\theta, \quad \theta \sim \mathcal{N}(0, 0.001*I_D), \quad d << D\ . \tag{12}$$

The experiment demonstrates F-HMC's capability to explore a set of generator settings to encapsulate a rich data distribution. We fit a regular GAN, Bayesian GAN, and our proposed model to a dataset with a dimension of $D = 100$ and 500 and $d = 2$. The generator for all models is set to be a two-layer neural network with dimensions of the layers as 10, 1000, and 100, fully connected, with ReLU activation. The red samples in Figure 4 depict the target data, whereas the green samples depict the corresponding generated data. The experiments on $D = 100$ are shown in the first two rows, while the results on $D = 500$ are shown on two lower rows. The sampler's name appears on the left side of each row. The more similarities between the green and red samples as iterations proceed (from left to right in each row) show the power of the corresponding sampler in estimating the target. Figure 4 for $D = 100$ and $D = 500$ indicates that the F-HMC has a better match to the target density because the produced samples are more accurate in their resemblance to red samples.

Figure 5 depicts the comparison of the performance of GAN, F-HMC GAN, and Bayesian GAN in terms of Jensen-Shannon divergence. The experiment estimates the similarity of the probability distribution of generated data to the original data, and it confirms that the F-HMC exceeds other models.

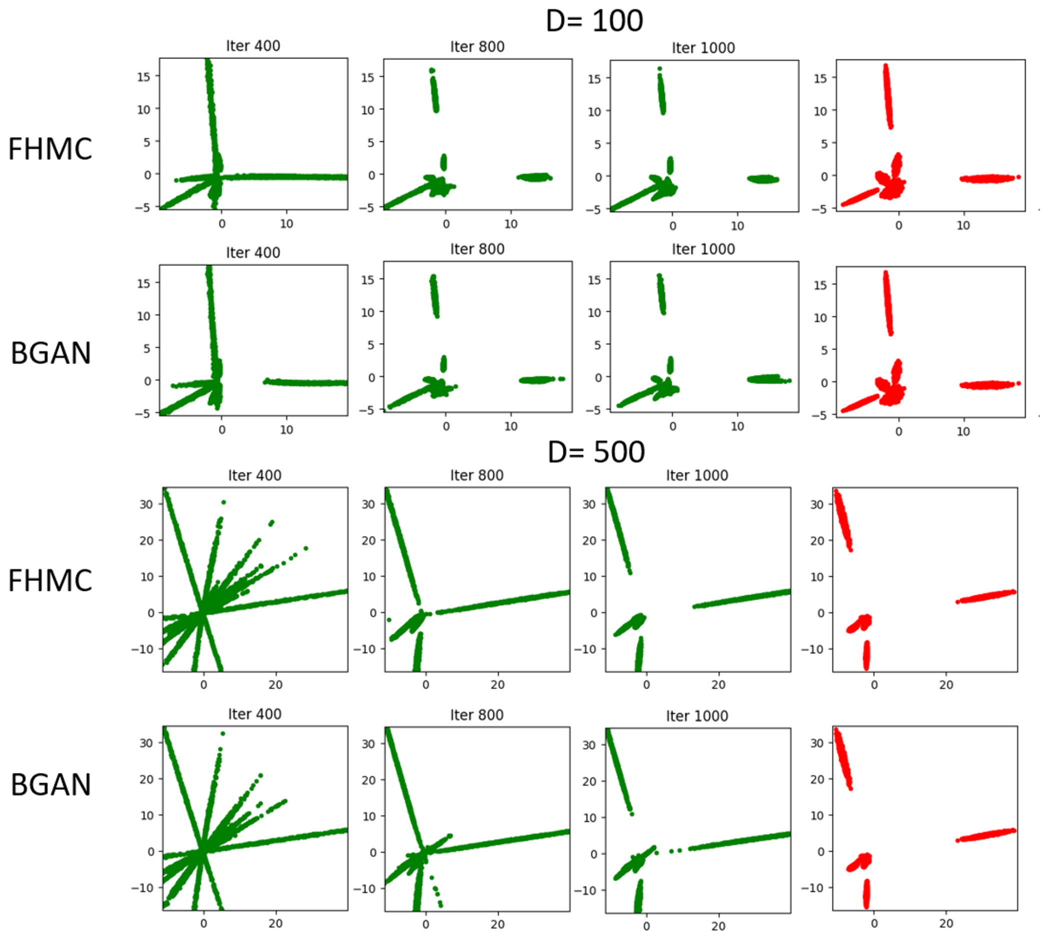


Figure 4: samples drawn from $p_{\text{data}}(x)$ and visualised in 2D after applying PCA. the red colour graph shows real data, and the green colour graph shows generated samples. The first two upper rows show experiment results with D = 100, and the two lower rows show those corresponding to D = 500. The corresponding sampler is shown on the right side of each row. The samples generated by F-HMC are more accurate in resembling the target.

5.3. Natural Image Dataset

To evaluate the performance of our proposed model on natural image datasets, namely CIFAR10, SVHN and ImageNet, we have employed experiments in three measurement levels: 1-performance metric in supervised and semi-supervised learning using test error rate. 2-run time per epoch in minutes by running all the models on a single GPU. 3-quality of generated images in terms of IS (Salimans et al., 2016) and FID (Heusel et al., 2017) scores. We should point out that the Bayesian GAN discriminator not only distinguishes real from generated data, it also get penalised for not classifying the correct label for real data. As a result, it demonstrated great out-of-sample performance with only a limited number of labelled

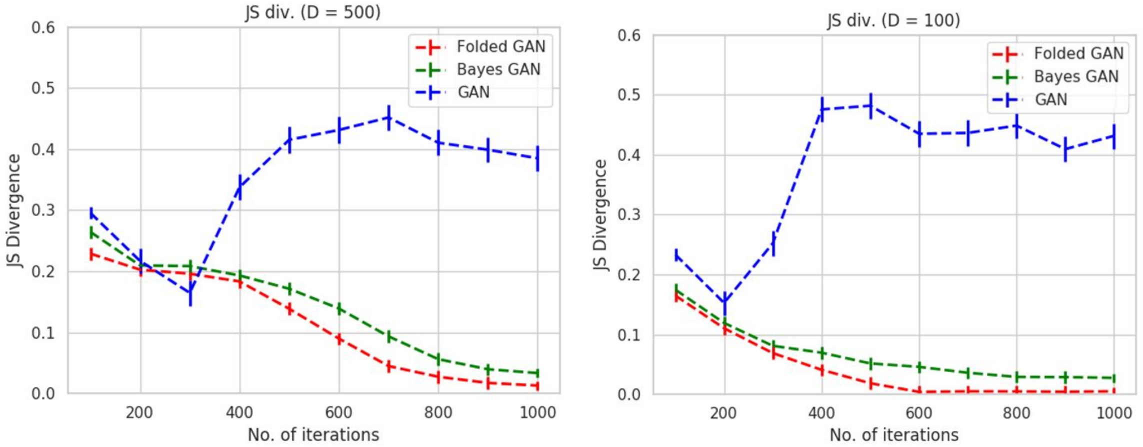


Figure 5: The figure shows the Jensen-Shannon divergence between $p_{\text{data}}(x)$ and the number of iterations of model training for $D = 100$ (left) and $D = 500$ (right). We can confirm that F-HMC exceeds other models in generating data more similar to the target

Table 1: The table shows the supervised and semi-supervised learning test error rates on classification for image benchmarks after splitting the dataset into train and test sets. The N_s shows the number of labelled examples. The experiments were repeated 10 times.

	N_s	Supervised	DCGAN10	W-DCGAN	BayesGAN	probGAN	F-HMC GAN
CIFAR-10	500	65.1 ± 2.3	30.9 ± 2.7	55.8 ± 2.9	30.5 ± 2.3	30.1 ± 2.8	$\mathbf{29.8 \pm 2.7}$
	1000	54.6 ± 2.1	29.1 ± 2.4	48.8 ± 3.2	$\mathbf{27.4 \pm 2.1}$	27.7 ± 3.1	27.6 ± 2.8
	2000	52.4 ± 2.4	26.8 ± 3.3	37.9 ± 2.5	24.2 ± 1.9	28.3 ± 2.5	$\mathbf{23.3 \pm 2.3}$
	4000	48.1 ± 1.0	24.7 ± 2.7	28.2 ± 2.9	22.3 ± 3.2	21.7 ± 2.8	$\mathbf{20.7 \pm 2.6}$
SVHN	1000	55.1 ± 3.3	30.8 ± 2.3	30.1 ± 1.9	28.7 ± 3.1	$\mathbf{26.4 \pm 2.1}$	26.6 ± 2.2
	2000	36.7 ± 2.63	17.9 ± 1.7	27.2 ± 2.6	14.2 ± 2.8	14.1 ± 2.6	$\mathbf{13.4 \pm 1.8}$
	4000	28.2 ± 3.13	15.8 ± 1.4	25.1 ± 2.8	12.7 ± 2.9	13.5 ± 1.7	$\mathbf{11.7 \pm 1.4}$
	8000	21.1 ± 2.2	15.1 ± 1.3	20.1 ± 1.9	9.2 ± 1.8	11.4 ± 1.8	$\mathbf{8.9 \pm 0.9}$
ImageNet	1000	57.6 ± 4.2	53.4 ± 3.1	55.7 ± 3.7	48.9 ± 4.3	47.8 ± 4.6	$\mathbf{43.8 \pm 4.4}$
	2000	42.3 ± 3.5	38.7 ± 2.5	40.6 ± 3.1	34.6 ± 4.6	34.5 ± 3.8	$\mathbf{33.6 \pm 3.7}$
	4000	40.1 ± 3.6	31.8 ± 2.1	35.5 ± 2.9	27.8 ± 3.8	26.8 ± 3.2	$\mathbf{25.9 \pm 3.5}$
	8000	36.8 ± 4.1	28.3 ± 1.8	34.3 ± 2.7	24.4 ± 3.1	24.1 ± 2.8	$\mathbf{22.7 \pm 2.7}$

inputs, highlighting the need for effective generative models for semi-supervised learning. Here, we follow their experimental setup. We evaluate the out-of-sampling performance for semi-supervised learning using a portion of labelled training examples.

We use a 5-layer network architecture for GAN's generator in all experiments on the natural images' datasets. The corresponding discriminator for supervised GAN is a 5-layer 2-class DCGAN, and we have used a 5-layer, K + 1 class DCGAN for a semi-supervised

Table 2: Inception scores (IS, higher is better), Frechet Inception Distance (FID, lower is better) both trained with WGAN objective and run time (epochs in minutes) results on natural images datasets.

	Score	10DCGAN	BayesGAN	probGAN	F-HMC GAN
CIFAR10	IS	7.78 ± 0.101	7.69 ± 0.96	7.72 ± 0.100	$\mathbf{8.86 \pm 0.095}$
	FID	23.81	24.75	24.63	**18.73**
	Run time	143	**91**	94	92
SVHN	IS	8.34 ± 0.107	8.27 ± 0.102	8.19 ± 0.087	$\mathbf{8.41 \pm 0.094}$
	FID	49.61	51.78	52.32	**11.21**
	Run time	151	98	94	**85**
ImageNet	IS	8.41 ± 0.113	8.51 ± 0.108	8.56 ± 0.075	$\mathbf{8.61 \pm 0.073}$
	FID	30.2	29.78	28.12	**12.83**
	Runt ime	671	358	349	**336**

GAN performing classification over K classes. We divided datasets into train/test sets and measured tests error on classification tasks.

Table 1 demonstrates supervised and semi-supervised learning results for all image benchmarks. Our proposed model mainly outperforms BayesGAN, probGAN, W-DCGAN, and 10-DCGAN in terms of test error rate. F-HMC shows its substantial impact when running on higher-dimensional data (ImageNet) due to its composition; it can efficiently explore higher dimension data. Table 2 shows the generated images' quality and the run time of the models. The quality of images improves by using F-HMC across all three datasets, while exhibiting similar or smaller run time compared to all the other compared state-of-the-art algorithms. This demonstrates the advantage of the composition of F-HMC, which leads to smaller run time per epoch than directly exploring the whole dimensions as in the Bayesian GAN and probGAN. It should be noted that method performance can be increased by using multiple GPUs to get a shorter runtime.

6. Conclusion

We present Folded Hamiltonian Monte Carlo (F-HMC) as a scalable strategy for sampling multi-modal, high-dimensional, highly correlated data to improve Bayesian GAN in producing synthetic images/generating data by estimating the weights of the generators and discriminators. We present its theoretical properties and demonstrate that it is capable of exploring the whole posterior rather than just one mode. Results show that it outperforms the state-of-the-art in terms of test error rates, run times per epoch, IS, and FID scores. Future directions include exploring the mixture of Student-t distributions to improve the method's robustness against misspecification of the number of modes. Our proposed model is a framework, and using a more efficient GAN structure can potentially enhance the results.

Acknowledgments

This study was supported by the UK Dementia Research Institute's Care Research and Technology Centre funded by the Medical Research Council (MRC), Alzheimer's Research UK, Alzheimer's Society (grant number: UKDRI–7002), and the UKRI Engineering and Physical Sciences Research Council (EPSRC) PROTECT Project (grant number: EP/W031892/1).

References

Ioan Andricioaei, John Straub, and Arthur Voter. Smart darting monte carlo. *The Journal of Chemical Physics*, 114:6994–7000, 04 2001. doi: 10.1063/1.1358861.

C Andrieu and E Moulines. On the ergodicity properties of some adaptive mcmc algorithms. *Annals of Applied Probability*, 16 (3):1462 – 1505, August 2006. ISSN 1050-5164. doi: 10.1214/105051606000000286. Publisher: Institute of Mathematical Statistics.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017.

Yan Bai, Radu V. Craiu, and Antonio F. Di Narzo. Divide and conquer: A mixture-based approach to regional adaptation for mcmc. *Journal of Computational and Graphical Statistics*, 20(1):63–79, 2011. doi: 10.1198/jcgs.2010.09035. URL https://doi.org/10.1198/jcgs.2010.09035.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=B1xsqj09Fm.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient hamiltonian monte carlo. *31st International Conference on Machine Learning, ICML 2014*, 5, 02 2014.

Radu V. Craiu, Lawrence Gray, Krzysztof Łatuszyński, Neal Madras, Gareth O. Roberts, and Jeffrey S. Rosenthal. Stability of adversarial markov chains, with an application to adaptive mcmc algorithms. *The Annals of Applied Probability*, 25(6), Dec 2015. ISSN 1050-5164. doi: 10.1214/14-aap1083. URL http://dx.doi.org/10.1214/14-AAP1083.

Jia Deng, R. Socher, Li Fei-Fei, Wei Dong, Kai Li, and Li-Jia Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, volume 00, pages 248–255, 06 2009. doi: 10.1109/CVPR.2009.5206848. URL https://ieeexplore.ieee.org/abstract/document/5206848/.

C. Geyer. Markov chain monte carlo maximum likelihood. *Comput. Sci. Statist.*, 23, 01 1991.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Y. Bengio. Generative adversarial networks. *Advances in Neural Information Processing Systems*, 3, 06 2014. doi: 10.1145/3422622.

Yongtao Guan and Stephen M. Krone. Small-world MCMC and convergence to multi-modal distributions: From slow mixing to fast mixing. *The Annals of Applied Probability*, 17(1):

284 – 304, 2007. doi: 10.1214/105051606000000772. URL https://doi.org/10.1214/105051606000000772.

Heikki Haario, Eero Saksman, and Johanna Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2):223 – 242, 2001. doi: bj/1080222083. URL https://doi.org/.

Hao He, Hao Wang, Guang-He Lee, and Yonglong Tian. Bayesian modelling and monte carlo inference for GAN. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1l7bnR5Ym.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, pages 6629–6640, 2017.

Ivan Kobyzev, Simon Prince, and Marcus Brubaker. Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 1–1, 2020. ISSN 1939-3539. doi: 10.1109/tpami.2020.2992934. URL http://dx.doi.org/10.1109/TPAMI.2020.2992934.

Supeng Kou, Q ZHOU, and WING WONG. Discussion paper equi-energy sampler with applications in statistical inference and statistical mechanics1,2,3. 01 2022.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

Shiwei Lan, Jeffrey Streets, and Babak Shahbaba. Wormhole hamiltonian monte carlo. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 2014, 05 2013.

Sean Meyn and R. Tweedie. *Markov Chains and Stochastic Stability*, volume 92. 01 1993. doi: 10.2307/2965732.

Blazej Miasojedow, Eric Moulines, and Matti Vihola. Adaptive parallel tempering algorithm. *Journal of Computational and Graphical Statistics*, 22, 05 2012. doi: 10.1080/10618600.2013.778779.

Yuval Netzer, T. Wang, A. Coates, A. Bissacco, Bo Wu, and A. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. F-gan: Training generative neural samplers using variational divergence minimization. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 271–279, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Emilia Pompe, Chris Holmes, and Krzysztof Latuszynski. A framework for adaptive mcmc targeting multimodal distributions. *Annals of Statistics*, 48:2930–2952, 10 2020. doi: 10.1214/19-AOS1916.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks, 2016.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2016.

Gareth O. Roberts and Jeffrey S. Rosenthal. Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science*, 16(4):351 – 367, 2001. doi: 10.1214/ss/1015346320. URL https://doi.org/10.1214/ss/1015346320.

Walter Rudin. *Principles of mathematical analysis / Walter Rudin.* McGraw-Hill New York, 3d ed. edition, 1976. ISBN 007054235. URL http://www.loc.gov/catdir/toc/mh031/75017903.html.

Yunus Saatci and A. G. Wilson. Bayesian gan. In *NIPS*, 2017.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 2234–2242, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

Cristian Sminchisescu and Max Welling. Generalized darting monte carlo. *Pattern Recognition*, 44:2738–2748, 10 2011. doi: 10.1016/j.patcog.2011.02.006.

Nanyang Ye and Zhanxing Zhu. Bayesian adversarial learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/586f9b4035e5997f77635b13cc04984c-Paper.pdf.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7354–7363. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/zhang19d.html.